CIA-TRIADE & SCHUTZZIELE

Informationssicherheit schützt Informationen und Systeme. Die CIA-Triade beschreibt drei zentrale Schutzziele. Weitere Ziele ergänzen sie, damit Identitäten, Berechtigungen und Handlungen nachvollziehbar bewertet werden können.

- Schutzziele helfen, Anforderungen und Risiken zu ordnen.
- Eine Maßnahme kann mehrere Ziele unterstützen; Zielkonflikte sind möglich.

1. VERTRAULICHKEIT (CONFIDENTIALITY)

Informationen sind nur für berechtigte Personen oder Systeme zugänglich.

- Beispiel: Zugriffsrechte begrenzen, wer eine Personalakte lesen darf.
- Mögliche Maßnahmen: Authentisierung, Autorisierung und Verschlüsselung.
- Risiko bei Verletzung: Offenlegung sensibler Daten.

2. INTEGRITÄT (INTEGRITY)

Informationen und Verarbeitung sollen korrekt und vor unbefugter oder unbeabsichtigter Veränderung geschützt sein.

- Beispiel: Eine Prüfsumme kann Veränderungen an einer Datei erkennbar machen.
- Zugriffsschutz und nachvollziehbare Änderungen unterstützen die Integrität.
- Integrität bedeutet nicht automatisch, dass eine Information ursprünglich wahr war.

3. VERFÜGBARKEIT (AVAILABILITY)

Berechtigte Nutzer sollen Informationen und Dienste bei Bedarf nutzen können.

- Beispiel: Backups und Wiederherstellungspläne helfen nach einem Ausfall.
- Redundanz und Kapazitätsplanung können Unterbrechungen abmildern.
- Angriffe, technische Fehler oder Stromausfälle können Verfügbarkeit beeinträchtigen.

4. AUTHENTIZITÄT

Authentizität betrifft die Echtheit einer Identität, Nachricht oder Herkunftsangabe.

- Authentisierung prüft eine behauptete behauptete Identität.
- Authentisierung prüft eine behauptete Identität, etwa beim Anmelden.
- Signaturen und Zertifikate können je nach Verfahren Herkunft oder Echtheit stützen.
- Authentizität allein gurantiert weder Vertraulichkeit noch fehlerfreie Inhalte.

5. VERBINDLICHKEIT (NON-REPUDIATION)

Verbindlichkeit soll unterstützen, dass eine Handlung oder Erklärung später nicht ohne Weiteres abgestritten werden kann.

- Digitale Signaturen können Belege für Zuordnung und Unveränderheit liefern.
- Aussagekraft hängt unter anderem von Schlüsselverwaltung und Prozess ab.
- Eine technische Signatur erestzt nicht automatisch rechtliche Bewertung.

6. AUTORISIERUNG

Autorisierung legt fest, welche Aktionen eine bereits identifizierte oder authentisierte Identität ausführen darf.

- Beispiel: Ein Konto darf Berichte lesen, aber keine Benutzerkonten anlegen.
- Das Prinzip der geringsten Rechte beschränkt Berechtigungen auf das Erforderliche.
- Authentisierung beantwortet „Wer bist du?", Autorisierung „Was darfst du?".

7. AUDITIERBARKEIT (ACCOUNTABILITY)

Auditierbarkeit ermöglicht, sicherheitsrelevante Aktivitäten anhand geeigneter Aufzeichnungen zu prüfen und Verantwortlichkeiten nachzuvollziehen.

- Protokolle sollten relevante Ereignisse, Zeit und Bezug zum Konto erfassen.
- Zugriff auf Protokolle schützen und Aufbewahrung zweckgebunden festlegen.
- Protokolle sind hilfreich, aber nicht automatisch vollständig oder manipulationssicher.

8. RISIKO- UND SCHADENSPOTENZIAL-BEWERTUNG

Risiken lassen sich anhand möglicher Ereignisse, Schwachstellen, Eintrittswahrschienlichkeit und Auswirkungen einschätzen.

- Frage: Welche Information oder welcher Dienst ist bertoffen?
- Betrachte mögliche Schäden für Vertraulichkeit, Integrität und Verfügbarkeit.
- Bewerte Auswirkungen und Unsicherheit; priorisiere angemessene Schutzmaßnahmen.
- Beispiel: Ein Ausfall des Terminservers kann den Betrieb stören, ohne Daten offenzulegen.

Merksatz

Schutzmaßnahmen richten sich nach dem konkreten Schutzbedarf und dem möglichen Schaden.

Einordnung

Die CIA-Triade ist ein Denkmodell, keine vollständige Liste aller Sicherheitsziele.

OSI-SCHICHTENMODELL & SCHICHTENSICHERHEIT

7 Ebenen ordnen Kommunikation; Angriffe und Schutz verteilen sich über den gesamten Stack.

1. BITÜBERTRAGUNGSSCHICHT (PHYSICAL)

Überträgt rohe Bits als elektrische, optische oder Funksignale über ein Medium. Sie beschreibt unter anderem Signal, Stecker und Übertragungsmedium.

- Beispiele: Ethernet-PHY, Glasfaser, WLAN-Funksignal.
- Angriffspunkte: manipulierte oder getrennte Kabel, Abhören ungeschützter Funkstrecken, Störsender.
- Schutz: physischer Zutrittsschutz, gesicherte Leitungswege, Funkabdeckung und Überwachung auf Störungen.



2. SICHERUNGSSCHICHT (DATA LINK)

Überträgt Daten in lokalen Netzen in Frames und nutzt dort lokale Adressen, etwa MAC-Adressen. Ethernet und WLAN umfassen Sicherungsfunktionen.

- Angriffspunkte: MAC-Spoofing, gefälschte DHCP- oder ARP-Nachrichten, unerlaubter Zugang zu WLAN oder Switchports.
- Schutz: WPA2 oder WPA3 passend konfigurieren, Switchports absichern, Netzwerke segmentieren und auffällige Änderungen überwachen.
- Ein MAC-Filter allein ist keine belastbare Authentisierung.



3. Vermittlungsschicht (NETWORK)

Vermittelt Pakete zwischen Netzen anhand logischer Adressen und Routing. IP ist das zentrale Beispiel; ICMP unterstützt unter anderem Diagnose und Fehlermeldungen.

- Angriffspunkte: gefälschte Quelladressen, missbrauchte oder fehlerhafte Routen, unerwünschter Paketverkehr.
- Schutz: Routing- und Firewall-Regeln nach Bedarf begrenzen, Netze segmentieren und Konfigurationen kontrollieren.
- IP selbst garantiert weder Vertraulichkeit noch Absenderauthentizität.



4. Transportschicht (TRANSPORT)

Stellt Ende-zu-Ende-Kommunikation zwischen Prozessen bereit. TCP bietet einen zuverlässigen, geordneten Bytestrom; UDP überträgt Datagramme ohne diese Zustellgarantie.

- Ports helfen, Verbindungen dem passenden Dienst zuzuordnen.
- Angriffspunkte: Port-Scans, überlastende Verbindungsversuche und unsichere oder unnötig erreichbare Dienste.
- Schutz: nur benötigte Ports freigeben, Dienste aktuell halten und Verbindungen angemessen begrenzen.
- TLS wird oft über TCP eingesetzt, ist im OSI-Modell aber nicht eindeutig einer einzelnen Schicht zuzuordnen.



5. Sitzungsschicht (SESSION)

Das klassische OSI-Modell beschreibt hier Aufbau, Steuerung und Beendigung von Kommunikationssitzungen. In heutigen Protokollstapeln werden diese Aufgaben häufig von Anwendungsprotokollen oder Bibliotheken übernommen.

- Angriffspunkte: gestohlene Sitzungstoken, unzureichende Sitzungsbeendigung oder Wiederverwendung kompromittierter Sitzungen.
- Schutz: zufällige, geschützte Sitzungstoken; Ablauf und Widerruf; sichere Cookies und erneute Anmeldung bei sensiblen Aktionen.
- Sitzungen sind im Internet nicht stets ein separates Protokoll oder eine klar abgegrenzte Schicht.



6. Darstellungsschicht (PRESENTATION)

Beschreibt im Modell die Darstellung, Kodierung und Umwandlung von Daten. Verschlüsselung und Kompression werden oft als Beispiele genannt; praktisch können sie in verschiedenen Schichten umgesetzt sein.

- Formate und Kodierungen müssen auf beiden Seiten sinnvoll interpretiert werden.
- Angriffspunkte: fehlerhafte Parser, unsichere Deserialisierung oder falsch eingesetzte Kryptografie.
- Schutz: Eingaben anhand eines erwarteten Formats prüfen, robuste Bibliotheken nutzen und Daten bei Bedarf mit aktuellen Verfahren schützen.
- TLS lässt sich nicht pauschal ausschließlich dieser OSI-Schicht zuordnen.



7. Anwendungsschicht (APPLICATION)

Stellt Netzwerkdienste für Anwendungen bereit. Hier werden Protokolle und Daten mit fachlicher Bedeutung ausgetauscht.

- Beispiele: HTTP, DNS, SMTP und SSH.
- Angriffspunkte: Phishing, fehlerhafte Zugriffssteuerung, Injection, unsichere Konfigurationen und gestohlene Zugangsdaten.
- Schutz: sichere Entwicklung, starke Authentisierung, minimale Berechtigungen, Updates und aussagekräftige Protokollierung.
- Ein Protokollname allein sagt nicht, ob seine konkrete Implementierung sicher ist.



8. Schichtübergreifende Bedrohungen

Reale Vorfälle lassen sich oft nicht auf eine einzelne OSI-Schicht reduzieren. Schwächen in Konfiguration, Identität, Software und Betrieb können zusammenwirken.

- Beispiel: Ein gestohlenes Kennwort ermöglicht einen Anwendungszugriff; Segmentierungsfehler erleichtern anschließend die Ausbreitung.
- Abwehr: mehrschichtige Schutzmaßnahmen, aktuelle Systeme, minimale Rechte und überwachte Netzwerkgrenzen.
- Untersuche den gesamten Kommunikationsweg: Gerät, lokales Netz, Routing, Transport und Anwendung.
- Das OSI-Modell hilft beim Ordnen und Eingrenzen; es ersetzt weder eine vollständige Analyse noch einen Sicherheitstest.



Merksatz

Frage bei jedem Vorfall: Welche Schicht transportiert die Daten, wo liegt der Angriffspunkt und welche Schutzmaßnahme wirkt genau dort?

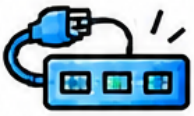
TCP/IP-PROTOKOLLSTAPEL UND HEADER-ANALYSE

Wie Daten durch Schichten gekapselt und über Netzwerke übertragen werden.

- Jede Schicht ergänzt Informationen für ihre Aufgabe.
- Beim Empfang werden die Header schrittweise ausgewertet.

1. NETWORK ACCESS LAYER

Verbindet Geräte über das jeweilige lokale Übertragungsmedium.



- Ethernet und WLAN übertragen Frames im lokalen Netz.
- MAC-Adressen adressieren Netzwerkschnittstellen auf der Verbindungsebene.
- Ein Frame enthält Steuerinformationen und Nutzdaten; die genaue Struktur hängt vom Link-Protokoll ab.

2. INTERNET LAYER

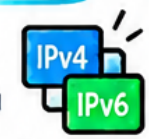
Vermittelt Pakete zwischen Netzwerken und ermöglicht Routing.



- IPv4-Adressen sind 32 Bit lang; IPv6-Adressen 128 Bit.
- IP ist verbindungslos und bietet selbst keine Zustellgarantie.
- Router nutzen IP-Zieladressen zur Weiterleitung; Link-Header gelten jeweils für den lokalen Übertragungsabschnitt.

3. IPv4- UND IPv6-HEADER

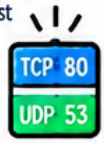
IP-Header tragen Adressierungs- und Steuerinformationen.



- IPv4: Quell- und Zieladresse, TTL, Protokollfeld sowie Gesamtlänge.
- IPv6: Quell- und Zieladresse, Hop Limit, Next Header sowie Payload Length.
- IPv4 kann Optionen enthalten; IPv6 nutzt für optionale Informationen Erweiterungsheader.
- Ein Paketmitschnitt kann je nach Erfassungspunkt bereits veränderte oder nur teilweise sichtbare Pakete zeigen.

4. TRANSPORT LAYER: TCP UND UDP

Transportprotokolle ordnen Daten einem Dienst auf dem Endgerät zu.



- TCP und UDP verwenden Quell- und Zielports.
- TCP bietet einen geordneten, zuverlässigen Bytestrom mit Wiederholungen und Flusskontrolle.
- UDP überträgt einzelne Datagramme ohne TCP-Verbindungsaufbau und ohne eigene Zustell- oder Reihenfolgegarantie.
- Ein Port allein identifiziert nicht weltweit einen Dienst; die Zuordnung hängt unter anderem von Adresse und Protokoll ab.

Merke
IP adressiert den Netzwerkpfad; Ports helfen dem Betriebssystem, Transportdaten einem Prozess zuzuordnen.

5. TCP-HANDSHAKE

Der übliche Verbindungsaufbau synchronisiert beide Endpunkte.



- SYN: Client fordert den Verbindungsaufbau an und sendet eine Anfangssequenznummer.
- SYN-ACK: Server bestätigt den SYN und sendet seine eigene Anfangssequenznummer.
- ACK: Client bestätigt den SYN-ACK; die Verbindung ist aufgebaut.
- Der Ablauf heißt Drei-Wege-Handshake. Abweichende Mitschnitte können durch Paketverlust, Wiederholungen oder den Erfassungszeitpunkt entstehen.

6. TCP-FLAGS UND HEADER-FELDER

Flags beschreiben Steuerfunktionen; weitere Felder unterstützen Übertragung und Auswertung.

- SYN baut eine Verbindung auf; ACK bestätigt Sequenzraum; FIN beendet eine Richtung geordnet; RST setzt eine Verbindung zurück.
- PSH fordert die zügige Übergabe gepufferter Daten an die Anwendung an; URG kennzeichnet relevante dringende Daten.
- Sequenz- und Bestätigungsnummern beziehen sich auf Bytes im TCP-Bytestrom.
- Fenstergröße unterstützt Flusskontrolle; Prüfsumme erkennt bestimmte Übertragungsfehler.
- Bedeutung und Kombination von Flags müssen im Zusammenhang mit Zustand und Richtung gelesen werden.



7. UDP-MECHANISMEN

UDP hält den Transportheader klein und überlässt zusätzliche Zuverlässigkeit höheren Protokollen.



- Der UDP-Header enthält Quellport, Zielport, Länge und Prüfsumme.
- UDP hat keinen Handshake, keine Bestätigungen und keine eingebaute Wiederholung oder Reihenfolgekontrolle.
- Anwendungen können bei Bedarf eigene Sequenzierung, Zeitüberschreitungen oder Wiederholungen implementieren.
- Die Prüfsumme dient der Fehlererkennung, nicht der Verschlüsselung oder Authentisierung.

8. HEADER-ANOMALIEN UND FRAGMENTIERUNG

Ungewöhnliche Header-Muster sind Hinweise zur Prüfung, aber für sich allein kein Beweis für einen Angriff.



- Achte auf widersprüchliche Längen, ungültige Prüfsummen, ungewöhnliche Flag-Kombinationen und unerwartete Fragmentierung.
- IPv4-Fragmentierung zerlegt ein Datagramm; Fragmente werden anhand von Identifikations- und Offset-Informationen zugeordnet.
- IPv6-Router fragmentieren nicht unterwegs; Fragmentierung kann durch den sendenden Host über einen Erweiterungsheader erfolgen.
- Filter, NAT, Offloading und [illegible] können Mitschnitte beeinflussen.
- Vergeeiche Paketdetails mit Verbindungskontext, Endpunkten und Sensorposition, bevor du einen Befund einordnest.

Einordnung
Eine Header-Anomalie ist ein Untersuchungsansatz. Prüfe Kontext und Erfassungsbedingungen, bevor du sie als Fehler oder Angriff bewertest.

ZENTRUM · DNS-ARCHITEKTUR & SICHERHEIT

DNS (Domain Name System) übersetzt Domainnamen in DNS-Antworten, etwa IP-Adressen. Die verteilte, hierarchische Namensauflösung ist leistungsfähig, aber ohne zusätzliche Schutzmechanismen nicht automatisch vertraulich oder gegen gefälschte Antworten abgesichert.

- Merksatz: DNS findet Namen; DNSSEC prüft die Herkunft und Integrität signierter DNS-Daten; DoH und DoT verschlüsseln den Transport zum Resolver.

1 · HIERARCHIE UND ZONEN



DNS verteilt Namensdaten auf eine hierarchische Struktur. Zuständigkeiten werden über Delegationen weitergereicht.

- Root-Zone: Ausgangspunkt der globalen DNS-Hierarchie.
- Top-Level-Domain (TLD): etwa .de oder .org.
- Autoritative Nameserver liefern Daten für ihre Zone.
- Eine Delegation verweist auf zuständige Nameserver einer untergeordneten Zone.

2 · REKURSION UND ITERATION



Eine rekursive Anfrage verlangt vom kontaktierten Resolver eine abschließende Antwort oder einen Fehler. Bei iterativer Auflösung verweist ein Nameserver auf den nächsten zuständigen Server; ein rekursiver Resolver kann diese Schritte für den Client ausführen.

- Client → rekursiver Resolver: typischerweise rekursive Anfrage.
- Resolver fragt bei Bedarf Root-, TLD- und autoritative Nameserver iterativ ab.
- Zwischenergebnisse können gemäß TTL im Cache liegen.
- Ein Verweis ist keine endgültige Antwort auf die gesuchte Adresse.

3 · WICHTIGE DNS-RECORDS



Resource Records (RRs) enthalten typisierte DNS-Daten. Der Record-Typ bestimmt, welche Information ein Eintrag bereitstellt.

- A: IPv4-Adresse; AAAA: IPv6-Adresse.
- NS: Nameserver einer Zone; CNAME: kanonischer Name als Alias.
- MX: Mailserver-Zuständigkeit; TXT: Textdaten, oft für Verifikations- oder E-Mail-Richtlinien.
- DS und DNSKEY: Material für die DNSSEC-Vertrauenskette.
- TTL gibt an, wie lange ein Cache Einträge höchstens verwenden soll.

4 · SPOOFING, HIJACKING UND CACHE POISONING



Angreifer können versuchen, DNS-Antworten umzulenken oder gefälschte Daten in einen Resolver-Cache einzubringen. Die Begriffe überschneiden sich, beschreiben aber unterschiedliche Perspektiven auf Manipulation und Umleitung.

- Spoofing: gefälschte DNS-Antworten vortäuschen.
- Cache Poisoning: falsche Daten in einem Resolver-Cache platzieren.
- Hijacking: Anfragen oder DNS-Konfiguration unbefugt umleiten, etwa durch kompromittierte Einstellungen.
- Folgen können falsche Zieladressen und Umleitung auf betrügerische Dienste sein.
- HTTPS-Zertifikatsprüfung bleibt wichtig; DNS allein beweist nicht die Identität einer Website.

WARNSIGNAL

Unerwartete DNS-Änderungen oder ungewöhnliche Zieladressen sind Hinweise zur Untersuchung, aber allein kein Beweis für einen Angriff.

5 · DNS-TUNNELING



DNS-Tunneling missbraucht DNS-Anfragen oder -Antworten als Träger für Daten oder Steuerkommunikation. Es kann zur Umgehung von Netzgrenzen oder zur Datenübertragung dienen.

- Auffällige Muster: ungewöhnlich viele Anfragen, lange oder zufällig wirkende Subdomain-Labels.
- Viele TXT-Abfragen oder ungewöhnliche Anfragevolumen können Anlass zur Prüfung sein.
- Legitime Anwendungen und Fehlkonfigurationen können ebenfalls auffälligen DNS-Verkehr erzeugen.
- Korrelation mit Endpunkt-, Proxy- und Firewall-Protokollen verbessert die Einordnung.

6 · DNSSEC-VALIDIERUNG



DNSSEC ergänzt DNS-Daten um digitale Signaturen und ermöglicht einem validierenden Resolver, die Authentizität und Integrität signierter Daten entlang einer Vertrauenskette zu prüfen.

- Zonensignaturen werden mit DNSKEY-Schlüsseln geprüft.
- DS-Einträge verbinden die Vertrauenskette zwischen übergeordneter und untergeordneter Zone.
- Validierung kann gefälschte oder veränderte signierte Daten erkennen.
- DNSSEC verschlüsselt DNS-Abfragen nicht und verbirgt den abgefragten Namen nicht.
- Fehlerhafte Signaturen oder eine unterbrochene Vertrauenskette können zu einer als ungültig bewerteten Antwort führen.

7 · DoH UND DoT



DNS over HTTPS (DoH) und DNS over TLS (DoT) schützen den Transport zwischen Client und entsprechendem Resolver durch Verschlüsselung.

- DoH überträgt DNS-Nachrichten innerhalb von HTTPS.
- DoT nutzt TLS für den DNS-Transport.
- Beide erschweren Beobachtung und Manipulation auf dem Transportweg zu diesem Resolver.
- Sie beweisen für sich genommen nicht, dass die erhaltenen DNS-Daten autoritativ oder korrekt sind.
- Der gewählte Resolver kann die Anfragen weiterhin
- Bei Verdacht: betroffene Konfiguration und Logs, bewerhackte keimnn Resolver-Vertrauen sind relevant.

8 · ABWEHR, RRL UND PRÜFUNG



Mehrschichtige Maßnahmen reduzieren Manipulations-, Missbrauchs- und Verfügbarkeitsrisiken. Response Rate Limiting (RRL) kann bei autoritativen DNS-Servern die Rate bestimmter wiederholter Antworten begrenzen und so deren Missbrauch für Verstärkungsangriffe erschweren.

- Resolver und autoritative Server aktuell halten; Zugriff und Konfiguration absichern.
- DNSSEC-Validierung auf geeigneten rekursiven Resolvern aktivieren und Fehler überwachen.
- RRL passend konfigurieren; legitime Anfragen und mögliche Nebenwirkungen berücksichtigen.
- DNS-Protokolle auf auffällige Volumen, Muster und unerwartete Änderungen untersuchen.
- Bei Verdacht: betroffene Konfiguration und Logs sichern, dann mit Endpunkt- und Netzwerkdaten korrelieren.

KURZ ZUSAMMENGEFASST

Rekursion beschreibt den Dienst für den Client, Iteration die schrittweise Auflösung. DNSSEC validiert signierte Daten; DoH/DoT verschlüsseln den Transport. RRL hilft gegen bestimmte Antwort-Fluten, ist aber kein vollständiger Schutz vor DDoS.

BEGRIFFE AUSEINANDERHALTEN

Verschlüsselung des Transportwegs ist nicht dasselbe wie kryptografische Validierung von DNS-Daten.

005 · GRUNDLAGEN SYMMETRISCHER KRYPTOGRAFIE

Ein gemeinsamer geheimer Schlüssel dient zum Ver- und Entschlüsseln. Die Sicherheit hängt nicht davon ab, den Algorithmus geheim zu halten, sondern davon, Schlüssel sicher zu erzeugen, zu speichern und auszutauschen.

- Vertraulichkeit: Unbefugte sollen den Klartext nicht lesen können.
- Verschlüsselung allein garantiert weder Integrität noch Absenderauthentizität.



Merksatz

Der Schlüssel ist geheim zu halten; ein gut geprüftes Verfahren darf öffentlich bekannt sein.

1. BLOCK- & STROMCHIFFREN



Beide sind symmetrische Verschlüsselungsverfahren, unterscheiden sich aber darin, wie sie Daten verabeiten.

- Blockchiffre: verarbeitet feste Blöcke, etwa AES mit 128-Bit-Blöcken.
- Stromchiffre: erzeugt einen Schlüsselstrom, der mit den Daten kombiniert wird; Schlüsselstrom niemals mit demselben Schlüssel/Nonce erneut verwenden.
- Blockchiffren brauchen für Nachrichten passende Betriebsarten; Stromchiffren benötigen ebenfalls korrekte Nonce- und Schlüsselverwaltung.

2. AES



Der Advanced Encryption Standard (AES) ist eine weit verbreitete Blockchiffre mit einer Blockgröße von 128 Bit.

- Zulässige Schlüssellängen: 128, 192 oder 256 Bit.
- Eine längere Schlüssellänge bedeutet nicht automatisch bessere Gesamtsicherheit: Betriebsart, Nonce und Schlüsselverwaltung zählen ebenfalls.
- AES verschlüsselt Blöcke; für längere Nachrichten wird eine geeignete Betriebsart verwendet.

3. DES & 3DES: HISTORIE



DES verwendet einen effektiven Schlüssel von 56 Bit und ist gegen Brute-Force-Angriffe nicht mehr sicher. 3DES erhöhte historisch die Widerstandsfähigkeit, gilt aber als veraltet und sollte für neue Systeme nicht eingesetzt werden.

- DES: heute unsicher, nicht für Schutzbedarf verwenden.
- 3DES: langsamer und ebenfalls aus der Nutzung auslaufend; moderre Systeme sollten etablierte aktuelle Verfahren nutzen.
- Historische Bekanntheit ist kein Sicherheitsnachweis.

Einordnung

Für neue Anwendungen ist AES mit einer geeigneten, korrekt eingesetzten Betriebsart die übliche Wahl.

4. BETRIEBSARTEN: ECB UND CBC



Eine Betriebsart bescheribt, wie eine Blockchiffre auf Nachrichten angewendet wird. Sie ist kein Ersatz für die Blockchiffre selbst.

- ECB verschlüsselt gleiche Klartextblöcke bei gleichem Schlüssel gleich: Muster bleiben sichtbar; für Nachrichten nicht verwenden.
- CBC verkettet Blöcke und benötigt einen unvorhersagbaren Initialisierungsvektor (IV).
- CBC bietet allein keine Integritätsprüfung; Änderungen können unbemerkt bleiben.
- IVs müssen gemäß Verfahren korrekt erzeugt und übertragen werden; sie sind normalerweise nicht geheim.

Wichtig

ECB ist für gewönnliche Nachrichtendaten ungeeignet. CBC ohne zusätzliche Authentifizierung ist ebenfalls keine vollständige sichere Nachrichtenlösung.

5. BETRIEBSARTEN: CTR UND GCM



CTR wandelt eine Blockchiffre in eine Stromchiffre um. GCM kombiniert Verschlüsselung mit Authentifizierung.

- CTR benötigt einen einmaligen Counter-/Nonce-Wert pro Schlüssel; Wiederverwendung kann Klartext-informationen offenlegen.
- CTR verschlüsselt, authentifiziert aber nicht selbst.
- GCM ist eine AEAD-Betriebsart: Sie schützt Vertraulichkeit und prüft Integrität sowie authentifizierte Zusatzdaten.
- Bei GCM darf ein Nonce mit demselben Schlüssel niemals wiederverwendet werden.

Praxisregel

Bevorzugt eine etablierte AEAD-Bibliothek verwenden und deren Nonce-Regeln exakt einhalten, keine Kryptografie selbst zusammensetzen.

6. SCHLÜSSELAUSTAUSCH



Symmetrische Verfahren setzen voraus, dass beide Seiten denselben geheimen Schlüssel besitzen. Der sichere Austasuch dieses Schlüssels ist eine zentrale Herausfoerdeung.

- Ein Schlüssel muss über einen geschützten Kanal oder mithilfe eines geeigneten Schlüsselaustausch-verfahrens etabliert werden.
- Asymmetrische Kryptografie kann beim Aufbau eines gemeinsamen Geheimnisses helfen; sie erstezt nicht automatisch die symmetrische Datenverschlüsselung.
- Schlüssel nicht in Quellcode, Logs oder ungeschützten Nachrichten ablegen.
- Zugriff begrenzen und Schlüssel bei Verdacht auf Offenlegung ersetzen.

7. AEAD UND KRYPTOANALYSE



Authenticated Encryption with Associated Data (AEAD) verschlüsselt den Inhalt und authentifiziert ihn. Zusatzdaten können authentifiziert, aber unverschlüsselt bleiben.

- Beispiele für Zusatzdaten: sichtbare Protokollfelder, die gegen unbemerkte Änderung geschützt werden sollen.
- Empfangene Daten erst verwenden, nachdem die Authentifizierungsprüfung erfolgreich war.
- Kryptoanalyse untersucht unter anderem Klartext-, Chiffrat- und Schlüsselbeziehungen sowie Implementierungsfehler.
- Sicherheit ergibt sich aus geprüftem Algorithmus, korrekter Betriebsart und sicherer Implementierung.

8. PADDING-ORACLE & MERKSÄTZE



Ein Padding-Oracle entsteht, wenn ein System Angreifern unterscheidbar verrät, ob die Padding-Prüfung eines entschlüsselten Chiffrats erfolgreich war. Solche Rückmeldungen können bei bertimmten CBC-Konstruktionen ausnutzbar sein.

- Fehlermeldungen, Antwortzeiten oder unterscheidliche Reakitionen können ein unbeabsichtigtes Orakel bilden.
- Fehler nicht als detaillierte, unterscheidbare Antworten nach außen geben; Integrität vor Verwendung prüfen.
- Wenn möglich AEAD statt selbst gebauter CBC-plus-Padding-Kenstruktionen einsetzen.
- Kurzüberblick: AES ist eine Blockchiffre; Betriebsart und Schlüssel-/Nonce-Verwaltung sind entscheidend; AEAD schützt zusätzlich Integrität.

Sicherheitsgrenze

Dies ist Lernmaterial zu Kryptografie-Grundlagen, keine Anleitung zur Prüfung fremder Systeme. Tests nur in ausdrücklich autorisierten Umgebungen durchführen.

006 GRUNDLAGEN ASYMMETRISCHER KRYPTOGRAFIE

Asymmetrische Kryptografie nutzt ein zusammengehöriges Schlüsselpaar: einen öffentlichen Schlüssel und einen geheimen privaten Schlüssel. Sie ermöglicht unter anderem vertraulichen Schlüsselaustausch und digitale Signaturen. Für große Datenmengen wird üblicherweise symmetrische Verschlüsselung eingesetzt.

- Öffentlich: darf verteilt werden
- Privat: geheim halten und schützen
- Sicherheit beruht auf geeigneten Verfahren, Parametern und Implementierungen

Wichtig
Ein öffentlicher Schlüssel allein beweist nicht, wem er gehört. Seine Zuordnung zu einer Identität muss vertrauenswürdig geprüft werden.

1. RSA-KRYPTOSYSTEM



RSA ist ein asymmetrisches Verfahren, dessen Sicherheit mit der Schwierigkeit der Faktorisierung großer geeigneter Zahlen zusammenhängt.

- Schlüsselerzeugung verwendet große Primzahlen
- Der öffentliche Schlüssel enthält typischerweise n und e
- Der private Schlüssel enthält geheime Informationen, darunter d
- Für Verschlüsselung nur mit sicherem, standardisiertem Padding verwenden; rohes RSA ist unsicher

2. Elliptic Curve Cryptography (ECC)



Elliptische-Kurven-Kryptografie umfasst asymmetrische Verfahren auf geeigneten elliptischen Kurven über endlichen Körpern.

- Sicherheit beruht auf dem Problem des diskreten Logarithmus auf Kurven
- Bei vergleichbarem Sicherheitsniveau sind oft kürzere Schlüssel als bei RSA möglich
- ECDH dient dem Schlüsselaustausch; ECDSA ist ein Signaturverfahren
- Kurven und Implementierungen müssen aus etablierten Standards stammen

3. Diffie-Hellman-Schlüsselaustausch



Diffie-Hellman ermöglicht zwei Parteien, über einen öffentlichen Kanal ein gemeinsames Geheimnis abzuleiten, ohne dieses Geheimnis direkt zu übertragen.

- DH arbeitet in geeigneten Gruppen; ECDH nutzt elliptische Kurven
- Ein einfacher unauthentifizierter Austausch verhindert keinen Man-in-the-Middle-Angriff
- Authentisierung, etwa durch Zertifikate oder Signaturen, schützt die Zuordnung der Gegenstellen
- Aus dem gemeinsamen Geheimnis werden mit einem KDF Sitzungsschlüssel abgeleitet

4. Digitale Signaturen



Eine digitale Signatur wird mit dem privaten Signaturschlüssel erzeugt und mit dem zugehörigen öffentlichen Schlüssel geprüft.

- Erkennt nachträgliche Änderungen am signierten Inhalt
- Unterstützt die Prüfung, ob eine Signatur zum öffentlichen Schlüssel passt
- Die Identität des Schlüsselinhabers muss separat vertrauenswürdig gebunden sein
- Signatur ist keine Verschlüsselung und verbirgt den Inhalt nicht

5. Falltürfunktionen und Primzahlen



Eine Einwegfunktion ist leicht zu berechnen, aber schwer umzukehren. Eine Falltür liefert Zusatzinformation, die die Umkehr erleichtert. Solche mathematischen Strukturen bilden Grundlagen bestimmter Public-Key-Verfahren.

- RSA nutzt Produkte großer Primzahlen und geheime Faktorisierungsinformation
- Das diskrete Logarithmusproblem liegt DH und ECC zugrunde
- Primzahlen allein machen ein Verfahren nicht sicher
- Zufallszahlen, Parameterwahl und korrekte Implementierung sind entscheidend

6. Schlüssellängen und Sicherheit



Schlüssellängen sind nur innenhalb eines bestimmten Verfahrens sinnvoll vergleichbar. Sicherheitsbedarf, Einsatzdauer und aktuelle Standards bestimmen die Wahl.

- RSA-Schlüssel werden typischerweise in Bit angegeben; 2048 Bit ist eine verbreitete Mindestgröße für viele heutige Anwendungen
- ECC-Schlüssel sind nicht direkt anhand derselben Bitzahl mit RSA vergleichbar
- Bevorzugt aktuelle Empfehlungen anerkannter Standards und gepflegte Bibliotheken
- Private Schlüssel sicher speichern, Zugriff begrenzen und bei Bedarf rotieren

Keine starre Garantie
Eine Bitzahl allein garantiert keine Sicherheit: Verfahren, Parameter, Implementierung und Bedrohungsmodell zählen ebenfalls.

7. Post-Quanten-Ausblick



Ein ausreichrend leistungsfähiger, fehlertoleranter Quantencomputer könnte verbretiete RSA- und ECC-Verfahren gefärnden. Post-Quanten-Kryptografie erforscht und standardisiert Verfahren, die gegen bekannte klassische und Quantenangriffe ausgelegt sind.

- Quantencomputer stellen nicht automatisch jede Kryptografie infrage
- Symmetrische Kryptografie und Hashfunktionen sind anders betroffen; Sicherheitsparameter können angepasst werden müssen
- Organisationen sollten kryptografische Abhängigkeiten erfassen und Austauschbarkeit einplanen
- Für neue Systeme aktuelle standardisierte Verfahren und Migrationshinweise beachten

8. Leistung und typischer Einsatz



Asymmetrische Verfahren benötigen meist mehr Rechenaufwand als symmetrische Verfahren. In vielen Protokollen authentisieren sie Gegenstellen oder vereinbaren Schlüssel; anschließend schützt ein symmetrischer Siztungsschlüssel den Datenverkehr.

- RSA kann bei großen Operationen rechenintensiv sein
- ECC bietet oft kompakte Schlüssel und Signaturen, abhängig von Verfahren und Plattform
- DH/ECDH vereinbart gemeinsame Geheimnisse, verschlüsselt aber nicht selbst den Datenstrom
- TLS kembiniert Authentisierung und Schlüsselaushandlung mit effizienter symmetrischer Verschlüsselung

007 HASHFUNKTIONEN & MACs

Hash = Fingerabdruck:
HMAC = schlüsselgebundener Integritätsschutz.

1. GRUNDPRINZIP



Eine kryptografische Hashfunktion H verarbeitet eine Nachricht m zu einem Digest H(m) mit festgelegter Bitlänge.

- Gleiche Eingabe und gleicher Algorithmus ergeben stets denselben Hashwert.
- Der Digest ist unabhängig davon gleich lang, ob die Eingabe kurz oder sehr groß ist.
- Hashfunktionen dienen unter anderem Integritätsprüfungen und als Baustein digitaler Signaturen.
- Ohne Schlüssel liefert ein Hash allein keine Absenderauthentisierung.

2. SICHERHEITSZIELE



Sicherheitsanforderungen beschreiben, wie schwierig bestimmte Aufgaben für Angreifende sein sollen.

- Urbildresistenz: Zu einem gegebenen Hashwert soll es praktisch nicht möglich sein, eine passende Eingabe zu finden.
- Zweiturbildresistenz: Zu einer vorgegebenen Nachricht soll keine andere mit gleichem Hash auffindbar sein.
- Kollisionsresistenz: Es soll praktisch schwierig sein, irgendein verschiedenes Nachrichtenpaar mit gleichem Hash zu finden.
- Eine Hashfunktion kann Kollisionen nicht ausschließen: Bei endlichem Ausgaberaum und unbegrenzten Eingaben existieren sie.

3. SHA-2 UND SHA-3



SHA-2 und SHA-3 sind unterschiedliche standardisierte Hashfamilien; ihre Konstruktionen unterscheiden sich.

- SHA-256 und SHA-512 gehören zur SHA-2-Familie und erzeugen Digestlängen von 256 beziehungsweise 512 Bit.
- SHA3-256 gehört zur SHA-3-Familie und erzeugt 256 Bit.
- SHA-3 ist keine bloße längere Version von SHA-2, sondern nutzt eine andere interne Konstruktion.
- Für neue Anwendungen geeignete Varianten und Parameter entsprechend dem aktuellen Sicherheitsbedarf wählen.

4. VERALTETE HASHALGORITHMEN



MD5 und SHA-1 gelten für kollisionssicherheitsabhängige Anwendungen als gebrochen und sollen nicht neu eingesetzt werden.

- Für MD5 und SHA-1 wurden praktisch nutzbare Kollisionsangriffe demonstriert.
- Ein Angreifer kann geeignete unterschiedliche Dateien mit gleichem Hash konstruieren.
- Ein Hashvergleich mit MD5 oder SHA-1 ist daher kein verlässlicher Kollisionsschutz.
- Bestehende Systeme auf geeignete moderne Verfahren umstellen, ein bloßer Austausch des Dateinamens genügt nicht.

WICHTIG

Für Passwortspeicherung sind schnelle Hashfunktionen wie SHA-256 ebenfalls ungeeignet; dafür sind speziell dafür entwickelte Passwort-Hashverfahren gedacht.

5. HMAC



HMAC erzeugt einen Message Authentication Code aus einer Nachricht, einem geheimen Schlüssel und einer kryptografischen Hashfunktion.

- HMAC verwendet eine standardisierte Konstruktion mit innerem und äußerem Hashschritt, nicht einfach Schlüssel und Nachricht aneinanderhängen.
- Sender und Empfänger benötigen denselben geheim gehaltenen Schlüssel.
- Der Empfänger berechnet den Tag erneut und vergleicht ihn sicher mit dem empfangenen Tag.
- Ein gültiger HMAC-Tag unterstützt Integrität und Authentizität zwischen Parteien, die den Schlüssel teilen.
- HMAC verschlüsselt die Nachricht nicht und verhindert allein keine Wiederholungsangriffe.

6. SALTED HASHES UND PASSWORTSPEICHERUNG



Passwort-Hashing muss Angriffe auf gestohlene Passwortdatenbanken erschweren.

- Ein Salt ist ein zufälliger, pro Passwort individueller Wert; er wird üblicherweise zusammen mit dem Ergebnis gespeichert.
- Salts verhindern gemeinsame vorberechnete Tabellen und gleiche Hashwerte bei gleichen Passwörtern, machen schwache Passwörter aber nicht stark.
- Argon2id ist eine verbreitete speicherharte Passwort-Hashoption; bcrypt und PBKDF2 sind weitere etablierte Verfahren.
- Parameter wie Arbeitsfaktor und Speicherbedarf passend zur Einsatzumgebung wählen und bei Bedarf aktualisieren.
- Passwörter nicht mit schnellem SHA-2, MD5 oder SHA-1 allein speichern; Bibliotheken mit sicherer Salt- und Parameterverwaltung nutzen.

7. ANGRIFFE UND GRENZEN



Die Auswahl der Konstruktion ist wichtig: Manche Hashkonstruktionen mit Merkle-Damgård-Aufbau können bei naiver Verwendung für Schlüsselpräfix-MACs anfällig für Length-Extension sein.

- Length-Extension kann unter bestimmten Bedingungen aus Hashwert und bekannter Nachrichtenlänge einen Hash für eine verlängerte Nachricht ermöglichen.
- Das ist kein allgemeiner Bruch von SHA-256 und nicht gleichbedeutend mit einer Kollision.
- HMAC ist so konstruiert, dass diese naive Schlüsselpräfix-Verwendung vermieden wird.
- Ein MAC verhindert nicht automatisch Lösshen, Umordnen oder Wiederholen gültiger Nachrichten; Protokolle brauchen dafür etwa Sequenznummern oder Nonces.

8. PRÜFSUMMEN SINNVOLL VALIDIEREN



Ein Digest aus einer unabhängigen, vertrauenswürdigen Quelle kann helfen, unbeabsichtigte Änderungen oder Übertragungsfehler zu erkennen.

- Datei hashen und den vollständigen Digest mit dem erwarteten Wert vergleichen.
- Stammt der erwartete Hash aus derselben manipulierbaren Quelle wie die Datei, beweist die Übereinstimmung keine Authentizität.
- Für einen authentischen Ursprung signierte Prüfsummen oder einen MAC mit vertrauenswürdigem Schlüssel verwenden.
- Algorithmen, Eingabedatei und verwendete Digestlänge dokumentieren; nicht auf visuelle Kurzvergleiche verlassen.

KURZFASSUNG

Hash: Fingerabdruck. HMAC: schlüsselgebundener Integritätsschutz. Passwort-Hash: absichtlich aufwendige, gesalzene Speicherung. Prüfsumme: nur so vertrauenswürdig wie Quelle und Prüfverfahren.

008 · PKI UND ZERTIFIKATSMANAGEMENT

Eine Public Key Infrastructure verbindet öffentliche Schlüssel mit geprüften Identitäten und organisiert Zertifikate über ihren gesamten Lebenszyklus.

- Vertrauensmodell für digitale Zertifikate

1 · X.509-ZERTIFIKAT



Ein X.509-Zertifikat bindet einen öffentlichen Schlüssel an Identitätsangaben und wird von einer ausstellenden Stelle digital signiert.

- Enthält unter anderem Aussteller, Subject, Gültigkeitszeitraum und öffentlichen Schlüssel.
- SAN (Subject Alternative Name) führt die DNS-Namen oder andere Identitäten auf, für die das Zertifikat gilt.
- Signaturprüfung bestätigt die Unveränderheit und die Signatur des Ausstellers; sie beweist nicht automatisch die Identität des Nutzers.
- Der private Schlüssel gehört nicht in das Zertifikat und muss geheim beliben.

2 · CA-VERTRAUENSKETTE



Eine Certificate Authority (CA) signiert Zertifikate. Clients prüfen Signaturen entlang einer Kette bis zu einem lokal vertrauten Vertrauensanker.

- Root-CA: Vertrauensanker; ihr selbstsigniertes Zertifikat wird typischerweise über einen Trust Store vertraut.
- Intermediate-CA: stellt häufig Endzertifikate aus und begrenzt die direkte Nutzung des Root-Schlüssels.
- Die Kette muss kryptografisch valide sein und zu einem vom Client akzeptierten Root-Zertifikat führen.
- Eine gültige Kette erstezt weder Namensprüfung noch Gültigkeits- und Sperrstatusprüfung.

3 · RA UND AUSSTELLUNG



Eine Registration Authority (RA) kann Identitätsangaben prüfen und die Zertifikatsausstellung unterstützen. Die CA signiert und verwaltet das Zertifikat.

- Antrag: gewünschte Namen und öffentlicher Schlüssel werden übermittelt.
- Validierung: die CA oder RA prüft die vorgeschriebene Kontrolle über Namen beziehungsweise Organisation.
- Ausstellung: die CA erstellt und signiert das Zertifikat mit passenden Nutzungsangaben.
- Die genaue Rollenverteilung zwischen RA und CA hängt vom jeweiligen Betriebsmodell ab.

4 · SPERRSTATUS: CRL UND OCSP



Sperrmechanismen informieren Clients, wenn ein Zertifikat vor seinem regulären Ablauf nicht mehr als vertrauenswürdig gelten soll.

- CRL: signierte Liste gesperrter Zertifikatsseriennummern, die von einer CA veröffentlicht wird.
- OCSP: Statusabfrage für ein bestimmtes Zertifikat bei einem Responder.
- OCSP-Stapling: der Server liefert eine zeitlich begrenzte, signierte OCSP-Antwort mit; der Client kann sise prüfen.
- Erreichbarkeit, Aktualität und Client-Verhalten beeinflussen, wie Sperrinformationen ausgewertet werden.
- Ablaufdatum und Sperrstatus sind unterschiedliche Prüfungen.

5 · CERTIFICATE PINNING



Pinning bindet eine Anwendung an erwartete Zertifikate oder öffentliche Schlüssel und kann dadurch bestimmte Fehl- oder Fremdzertifikate zurückweisen.

- Ein Pin kann beipisielweise einen öffentlichen Schlüssel oder dessen Hash bezeichnen.
- Pinning ist zusätziliche Prüfung, kein Ersatz für sichere TLS-Validierung.
- Falsch gepflegte Pins können legitime Zertifikatswechsel blockieren und Anwendungen unerreichbar machen.
- Betrieb braucht einen sicheren Rotations- und Wiederherstellungsplan; unbedachtes Pinning birgt Verfügbarkeitsrisiken.

6 · SCHLÜSSEL-LEBENSZYKLUS



Schlüsselmanagement umfasst Erzeugung, Speicherung, Nutzung, Rotation, Sicherung und sichere Außerbetriebnahme von Schlüsseln.

- Erzeugen: geeignete kryptografische Verfahren und ausreichend sichere Zufallsquellen verwenden.
- Schützen: private Schlüssel mit Zugriffskontrollen und geeigneter Hardware- oder Software-Sicherung schützen.
- Rotieren: Wechsel und Erneuerung planbar durchführen; Zertifikatslaufzeit ist nicht gleich Schlüsselwechsel.
- Kompromittierung: betroffene Zertifikate sperren, Schlüssel ersetzen und Auswirkungen untersuchen.
- Außerbetriebnahme: nicht mehr benötigte Schlüssel sicher entfernen; erforderliche Nachweise und Wiederherstellung berücksichtigen.

7 · CROSS-SIGNING



Beim Cross-Signing signiert eine CA ein Zertifikat einer anderen CA, oodass zusätzliche Vertrauenspfade entstehen können.

- Kann Übergänge zwischen Vertrauensökosystemen unterstützen.
- Clients können je nach Trust Store und verfügbarer Kette unterschiedliche Pfade auswählen.
- Cross-Signing macht einen Vertrauensanker nicht universell vertrauenswürdig.
- Kettenaufbau, Ablaufdaten und Client-Kompatibilität müssen im jeweiligen Einsatz geprüft werden.

8 · SAN UND WILDCARD



SAN-Zertifikate decken die ausdrücklich aufgeführten Namen ab. Wildcard-Namen können mehrere Hosts unter einer Domain abdecken, jedoch nicht beliebig viele Ebenen.

- Prüfe den angefragten Hostnamen gegen die SAN-Einträge des Zertifikats.
- Ein Wildcard-Eintrag wie *.example.org deckt typischerweise einen Hostnamen direkt unter example.org ab, nicht beliebige Subdomain-Tiefen.
- Der genaue Abgleich bestätigt den Regeln des eingesetzten TLS-Clients und Namensformats.
- Breite Namensabdeckung erleichtert Betrieb, vergrößert aber bei Schlüsselkompromittierung den möglichen Wirkungsbereich.

MERKSATZ

Ein gültiges Zertifikat allein beweist nicht, dass eine Verbindung sicher ist: Identität, Gültigkeit, Vertrauenskette und Nutzungskontext müssen ebenfalls geprüft werden.

TLS 1.2 vs. TLS 1.3

TLS im Überblick



TLS schützt Daten während der Übertragung zwischen Kommunikationspartnern. Beim Verbindungsaufbau werden Parameter ausgehandelt, Schlüsselmaterial abgeleitet und die Identität des Servers typischerweise anhand eines Zertifikats geprüft.

- TLS liegt oberhalb eines Transportprotokolls; HTTPS verwendet TLS für HTTP-Verbindungen.
- Schutzziele: Vertraulichkeit, Integrität und – bei erfolgreicher Zertifikatsprüfung – authentisierte Serveridentität.
- TLS erstetzt weder sichere Anwendungen noch schützt es Daten vor kompromittierten Endgeräten.

Einordnung
Die Darstellung ist ein Lernmodell. Der genaue Nachrichtenablauf hängt von Version, Erweiterungen und Konfiguration ab.

1. TLS 1.2 und TLS 1.3



Beide Versionen unterstützten verschlüsselte Verbindungen, unterscheiden sich aber unter anderem in Handshake und Cipher-Suite-Definition.

- TLS 1.3 reduziert den regulären vollständigen Handshake typischerweise auf einen Netzwerk-Round-Trip nach Aufbau des Transports.
- TLS 1.2 hat je nach Schlüsselaustausch und Konfiguration einen längeren Ablauf; die genaue Zahl der Nachrichten kann variieren.
- TLS 1.3 entfernte viele veraltete und unsichere Optionen, darunter statischen RSA-Schlüsselaustausch und veraltete Cipher.
- Eine höhere Versionsnummer allein beweist keine sichere Konfiguration.

2. Vollständiger Handshake



Beim vollständigen Handshake einigen sich Client und Server auf gemeinsame Parameter und erzeugen frisches Sitzungsschlüsselmaterial.

- ClientHello: unterstützte Versionen, Zufallswert, Cipher-Angebote und Erweiterungen; oft mit KeyShare.
- ServerHello: Auswakl der Parameter und gegebenenfalls Server-KeyShare.
- TLS 1.3 schützt nach ServerHello weitere Handshake-Nachrichten mit daraus abgeleiteten Schlüsseln.
- Der Server sendet sein Zertifikat und einen CertificateVerify-Nachweis; der Client prüft die Authentisierung.
- Finished-Nachrichten bestätigen den Handshake-Transcript und den Besitz des abgeleiteten Schlüssels.
- Bei TLS 1.2 sind die Nachrichten anders angeordnet; nicht jedes Detail des TLS-1.3-Flows gilt dort.

3. Ephemerer Diffie-Hellman und PFS



Ephemerer Diffie-Hellman-Schlüsselaustausch (DHE oder ECDHE) erzeugt pro Verbindung frisches gemeinsames Geheimnis. Die Authentisierung bindet den Austausch an die Gegenstelle.

- PFS (Forward Secrecy) begrenzt den Schaden, falls ein langfristiger privater Authentisierungsschlüssel später kompromittiert wird.
- Dazu müssen flüchtige Sitzungsschlüssel und ephemeres Schlüsselmaterial nach Gebrauch verworfen werden.
- TLS 1.3 verwendet für den normalen Handshake (EC)DHE-KeyShares; PSK kann je nach Modus hinzukommen.
- PFS schützt nicht vor kompromittierten Endpunkten, aufgezeichteten Klartexten oder Fehlern bei der Zertifikatsprüfung.

4. Cipher Suites



Eine Cipher Suite bezeichnet eine vereinbarte Kombination kryptografischer Verfahren. Ihre Bedeutung ist versionsabhängig.

- TLS 1.2-Suites kodieren typischerweise Schlüsselaustausch, Authentisierung, Verschlüsselung und Hash/MAC.
- TLS 1.3-Suites legen AEAD-Verschlüsselung und Hashfunktion fest; Key Exchange und Signaturalgorithmus werden separat ausgehandelt.
- AEAD-Verfahren wie AES-GCM oder ChaCha20-Poly1305 verbinden Verschlüsselung und Integritätsschutz.

  Beispiel TLS 1.3: TLS_AES_128_GCM_SHA256. Der Name bezeichnet nicht den Schlüsselaustausch.

5. Record Layer

Der TLS-Record-Layer transportiert TLS-Nachrichten und Anwendungsdaten in geschützten Datensätzen.

- Anwendungsdaten werden mit Sitzungsschlüsseln verschlüsselt und authentisiert.
- TLS 1.3 verschlüsselt den Inhalt der geschützten Records; äußere Felder bleiben für den Transport sichtbar.
- Records werden in Transportdaten übertragen; ein Record entspricht nicht zwingend einer Anwendungsmeldung oder einem Netzwerkpaket.
- Integritätsschutz erkennt Veränderungen, verhindert aber nicht jede Verkehrsbeobachtung wie Timing oder Datenvolumen.



6. Wiederaufnahme und 0-RTT



TLS 1.3 kann frühere Sitzungen über PSK-basierte Wiederaufnahme fortsetzen. Im 0-RTT-Modus darf der Client bestimmte frühe Daten vor Abschluss des neuen Handshakes senden.

- 0-RTT spart Wartezeit, bietet für frühe Daten aber keinen Schutz vor Replay durch Angreifer.
- Server und Anwendung sollten 0-RTT nur für geeignete, idempotente Vorgänge zulassen.
- Zustandsändernde Aktionen wie Kauf oder Überweisung sind ohne zusätzliche Replay-Abwehr ungeeignet.
- 0-RTT-Daten haben nicht dieselben Forward-Secrecy-Eigenschaften wie Daten nach abgeschlossenem frischem Schlüsselaustausch.

Merke
Wiederaufnahme ist kein Freibrief: Replay-Risiko und Vertraulichkeitsanforderungen der Anwendung müssen berücksichtigt werden.

7. ALPN-Aushandlung



Application-Layer Protocol Negotiation (ALPN) teilt während des TLS-Handshakes mit, welches Anwendungsprotokoll über der Verbindung verwendet wird.

- Der Client nennt unterstützte Protokollkennungen, zum Beispiel h2 oder http/1.1.
- Der Server wählt eine unterstützte Kennung aus und teilt sie im Handshake mit.
- ALPN legt nicht selbst fest, wie das ausgehandelte Protokoll sicher implementiert wird.
- Bei Nichtübereinstimmung muss die Verbindung entsprechend der Anwendungskonfiguration behandelt werden.

8. Downgrade-Schutz und Zertifikatsprüfung



Downgrade-Angriffe versuchen, Kommunikationspartner auf eine ältere oder schwächere Protokolloption zu lenken. Zertifikatsprüfung schützt vor falscher Serveridentität.

- TLS 1.3 enthält Downgrade-Schutzsignale für bestimmte Rückfälle auf TLS 1.2 oder älter; sie erstezen keine sichere Konfiguration.
- Veraltete Protokolle und unsichere Cipher deaktivieren; Versionen und Parameter bewusst konfigurieren.
- Zertifikatskette bis zu einem vertrauenswürdigen Anker prüfen, Gültigkeit und Servernamen abgleichen.
- Auch Sperrstatus und Nutzungszweck sind gemäß Client- und Richtlinienverhalten relevant.
- Bei Fehlern darf die Anwendung nicht stillschweigend auf unverschlüsseltes HTTP oder ungeprüfte Zertifikate ausweichen.

010 · IAM

Identitäten verwalten, Zugriffe gezielt steuern.

IAM IM ÜBERBLICK



Identity and Access Management (IAM) ordnet digitale Identitäten und steuert, wer auf welche Ressourcen zugreifen darf.

Die acht Bausteine zeigen zentrale Begriffe und Abläufe.

- Identität beschreiben, Anmeldung prüfen und Rechte gezielt vergeben.
- Zugriffe während des gesamten Lebenszyklus verwalten und regelmäßig überprüfen.

1. IDENTIFIKATION



Identifikation benennt, welche digitale Identität einen Zugriff anfragt.

- Beispiele: Benutzername, Mitarbeitenden-ID oder Dienstkonto.
- Ein Kennzeichen allein beweist nicht, wer es verwendet.
- Konten eindeutig zuordnen und gemeinsam genutzte Konten vermeiden, wo möglich.

2. AUTHENTIFIZIERUNG



Authentifizierung prüft, ob eine Person oder ein Dienst die angegebene Identität glaubhaft belegen kann.

- Faktoren: Wissen (Passwort), Besitz (Sicherheitsschlüssel) oder Inhärenz (biometrisches Merkmal).
- Mehr-Faktor-Authentifizierung kombiniert Faktoren unterschiedlicher Kategorien.
- Authentifizierung bestätigt einen Nachweis, nicht automatisch die Berechtigung für jede Ressource.

3. AUTORISIERUNG



Autorisierung entscheidet, welche Aktionen eine authentifizierte Identität auf einer Ressource ausführen darf.

- Berechtigungen können Lesen, Ändern oder Administrieren umfassen.
- Entscheidungen richten sich nach einer festgelegten Richtlinie und dem Zugriffskontext.
- Authentifizierung beantwortet „Wer?", Autorisierung „Was darf diese Identität?"

4. LEAST PRIVILEGE



Das Prinzip der geringsten Rechte beschränkt Berechtigungen auf das notwendige Maß für eine Aufgabe.

- Nur benötigte Ressourcen und Aktionen freigeben.
- Zeitlich begrenzte Rechte nutzen, wenn der Bedarf vorübergehend ist.
- Berechtigungen nach Aufgabenwechsel oder Projektende anpassen.

5. SEPARATION OF DUTIES (SoD)



Aufgabentrennung verteilt kritische Tätigkeiten auf mehrere Rollen oder Personen, um Fehler und Missbrauch schwerer zu machen.

- Beispiel: Eine Person beantragt eine Zahlung, eine andere genehmigt sie.
- Unvereinbare Berechtigungen in Richtlinien festlegen und prüfen.
- Bei kleinen Teams mögliche Ausnahmen mit geeigneten Kontrollen auffangen.

6. RBAC UND ABAC



Zugriffsmodelle helfen, Berechtigungen nachvollziehbar zu vergeben.

- RBAC: Rechte werden Rollen zugeordnet; Identitäten erhalten passende Rollen.
- ABAC: Eine Richtlinie bewertet Attribute, etwa Abteilung, Ressource, Aktion oder Kontext.
- Modelle können kombiniert werden; Richtlinien sollten verständlich und überprüfbar bleiben.

7. IDENTITY LIFECYCLE (JML)



JML beschreibt Änderungen im Kontolebenszyklus: Joiner (Eintritt), Mover (Rollen- oder Aufgabenwechsel) und Leaver (Austritt).

- Joiner: Identität anlegen und passende Startrechte vergeben.
- Mover: Nicht mehr benötigte Rechte entfernen und neue Aufgaben berücksichtigen.
- Leaver: Zugang zeitnah sperren oder entziehen und zugehörige Zugänge prüfen.

8. PAM UND ZUGRIFFSREZERTIFIZIERUNG



Privileged Access Management (PAM) schützt besonders mächtige Konten. Bei der Zugriffsrezertifizierung bestätigen oder korrigieren Verantwortliche bestehende Berechtigungen.

- Adminrechte auf notwendige Aufgaben und Zeiträume begrenzen.
- Privilegierte Zugriffe nachvollziehbar protokollieren und schützen.
- Regelmäßig prüfen: Ist der Zugriff noch erforderlich und angemessen?

011 · MFA

Mehrere voneinander verschiedene Nachweise erschweren unbefugte Anmeldungen. Entscheidend ist nicht die Anzahl der Schritte, sondern die Kombination unterschiedlicher Faktorarten.

1. WISSEN

Etwas, das die Person kennt.



- Passwort oder Passphrase
- PIN
- Geheimantwort – oft leicht zu erraten oder zu recherchieren
- Ein gestohlenes Passwort kann weiterverwendet werden.

2. BESITZ

Etwas, das die Person hat oder kontrolliert.



- Authenticator-App oder Hardware-Sicherheitsschlüssel
- Registriertes Telefon für einen Einmalcode
- Ein Gerät muss sicher dem Konto zugeordnet werden.
- Verlust und Ersatz des Geräts brauchen einen **sicheren Wiederherstellungsweg.**

3. INHÄRENZ

Ein körperliches oder verhaltensbezogenes Merkmal, das zur Person gehört.



- Fingerabdruck oder Gesichtserkennung
- Biometrie ist nicht gleich Besitz eines eigenständigen Faktors: Die genaue Umsetzung zählt.
- Biometrische Merkmale lassen sich nach Offenlegung nicht einfach wie ein Passwort wechseln.
- Lokale Gerätebiometrie kann einen kryptografischen Schlüssel freigeben, ohne dass das Merkmal selbst an den Dienst gesendet wird.

Einordnung

Ob eine konkrete Anmeldung wirklich mehrere Faktoren nutzt, hängt vom gesamten Verfahren ab – nicht nur davon, dass mehrere Eingaben erscheinen.

4. SMS- UND E-MAIL-CODES

Einmalcodes können eine zusätzliche Hürde sein, sind aber nicht alle gleich sicher.



- SMS kann durch SIM-Swap, Rufnummernübernahme oder Weiterleitung abgefangen werden.
- E-Mail-Codes hängen von der Sicherheit des E-Mail-Kontos und dessen Wiederherstellung ab.
- Codes können durch Phishing erbeutet werden; gib sie nicht auf einer unerwarteten Anmeldeseite ein.
- Wenn möglich, stärkere phishing-resistente Verfahren bevorzugen.

5. HOTP UND TOTP

Zeit- oder zählerbasierte Einmalpasswörter werden häufig von Authenticator-Apps erzeugt.



- HOTP basiert auf einem gemeinsamen Geheimnis und einem Zähler; erfolgreiche Nutzung schreitet den Zähler fort.
- TOTP leitet den Code aus einem gemeinsamen Geheimnis und einem Zeitintervall ab.
- Beide Verfahren verwenden zeitlich begrenzte oder einmalige Codes, die manuell eingegeben werden.
- Ein Angreifer kann einen abgefangenen Code in Echtzeit weiterleiten; der Code allein verhindert Phishing nicht.

Geheimnis schützen

Das bei der Einrichtung angezeigte gemeinsame Geheimnis ist sensibel. Sichere es nicht in einem öffentlich zugänglichen Screenshot oder ungeschützten Dokument.

6. FIDO2 UND WEBAUTHN

Sicherheitsschlüssel und unterstützte Plattform-Authentifikatoren ermöglichen eine Anmeldung mit kryptografischen Schlüsseln.



- WebAuthn bindet Anmeldedaten an die jeweilige Website beziehungsweise ihren Ursprung.
- Der private Schlüssel bleibt beim Authentifikator; der Dienst prüft eine kryptografische Antwort.
- Passkeys können eine phishing-resistente Anmeldung ermöglichen.
- Kontowiederherstellung und zusätzliche Anmeldewege müssen ebenfalls sicher gestaltet sein.

7. MFA FATIGUE UND PUSH-SPAMMING

Angreifer können wiederholt Anmeldebestätigungen auslösen, um eine Person zu ermüden oder zu einer **Freigabe zu verleiten.**



- Unerwartete Push-Anfragen nicht bestätigen.
- Anmeldeversuche ablehnen und dem zuständigen Sicherheitsteam melden.
- Nummernabgleich oder kontextbezogene Anzeige kann Fehlfreigaben erschwerer.
- Eine Freigabe nur erteilen, wenn man die eigene Anmeldung gerade begonnen hat.

Bei wiederholten Anfragen

Nicht aus Gewohnheit zustimmen. Zugangsdaten bei Verdacht auf Offenlegung über den offiziellen Weg ändern und den Vorfall melden.

8. RISIKOBASIERTE ADAPTIVE AUTHENTIFIZIERUNG

Ein System kann zusätzliche Prüfungen verlangen, wenn eine Anmeldung ungewöhnlich oder riskant wirkt.



- Mögliche Signale: neues Gerät, ungewohnter Ort oder verdächtiges Anmeldeverhalten.
- Ein Signal ist kein Beweis für einen Angriff; Kontext und Fehlalarme beachten.
- Zusätzliche Prüfung kann eine erneute Anmeldung oder einen stärkeren Faktor verlangen.
- MFA bleibt wichtig: Risikobewertung ersetzt keinen verlässlichen Anmeldenschutz.

Begriffe

Authentifizierung prüft einen behaupteten Identitätsnachweis. Autorisierung legt fest, worauf ein angemeldetes Konto zugreifen darf.

0. FÖDERIERTE IDENTITÄTEN & SSO



Ein Identitätsanbieter stellt Identitätsinformationen bereit; ein Dienstanbieter nutzt sie, um Zugriff auf einen Dienst zu ermöglichen. Föderation verbindet getrennte Systeme über vereinbarte Vertrauensbeziehungen. SSO ermöglicht mehrere Zugriffe nach einer Anmeldung, ersetzt aber nicht die Prüfung jedes einzelnen Zugriffs.

- Föderation: Identitäten oder Authentifizierungen werden über Systemgrenzen hinweg anerkannt.
- SSO: Anmeldung bei einem Identitätsanbieter kann den Zugang zu mehreren verbundenen Diensten ermöglichen.
- Protokolle, Zielgurppe und Vertrauensregeln bestimmen, welche Aussagen ein Dienst übernehmen darf.

Wichtig

Eine erfolgreiche Anmeldung beweist nicht automatisch, dass jede angeforderte Aktion erlaubt ist. Authentifizierung und Autorisierung sind unterschiedliche Prüfungen.

2. SAML 2.0: ANMELDEABLAUF



SAML 2.0 ist ein XML-basiertes Framework für föderierte Identitätsaussagen. Beim verbreiteten browserbasierten Web-SSO sendet der SP eine Authentifizierungsanfrage an den IdP. Der IdP authentifiziert den Nutzer und übermittelt eine SAML-Aussage, häufig über den Browser, an den Assertion Consumer Service des SP.

- SP erstellt oder veranlasst eine AuthnRepuest; Weiterleitung und Bindung hängen von der Konfiguration ab.
- IdP authentifiziert den Nutzer und erstellt eine SAML Response mit einer Assertion.
- SP prüft Signatur, Aussteller, Empfänger, Ziel, Zeitbedingungen und Anfragebezug gemäß Profil und Konfiguration.
- Erst nach erfolgreicher Prüfung kann der SP eine lokale Sitzung anlegen.
- SAML-Assertionen sind nicht einfach mit OAuth-Zugriffstoken gleichzusetzen.

Prüfpunkt

Eine gültige Signatur allein genügt nicht: Auch Kontext, Zielgruppe, Zeitfenster und weitere Profilbedingungen müssen passen.

3. OAUTH 2.0: AUTORISIERUNG



OAuth 2.0 ist ein Autorisierungsframework. Es ermöglicht einem Client, mit Zustimmung und gemäß Richtlinien begrenzten Zugriff auf geschützte Ressourcen zu erhalten. OAuth selbst ist kein allgemeines Protokoll zur Nutzeranmeldung.

- Rollen: Resource Owner, Client, Authorization Server und Resource Server.
- Authorization Code Flow: Der Client erhält einen Code und tauscht ihn am Token-Endpunkt gegen Tokens ein.
- Für öffenliche Clients wird PKCE eingestazt, um den Code an die ursprüngliche Anfrage zu binden.
- Redirect-URIs müssen exakt gegen registrierte Werte geprüft werden.
- Implicit Grant und Resource Owner Password Credentials Grant gelten in akteellen Best Practices als veraltet; keine neuen Anwendungen darauf aufbauen.

Abgrenzung

Ein Access Token ist für den Zugriff auf Ressourcen bestimmt. Es ist nicht uutomatisch ein Beleg für die Identität des Nutzers beim Client.

4. OPENID CONNECT: IDENTITÄTSLAYER



OpenID Connect (OIDC) ergänzt OAuth 2.0 um eine standardisierte Identiältsschicht. Bei einer erfolgreichen Authentifizierungsanfrage erhält der Client typischerweise ein ID Token mit Aussagen zur Authentifizierung und zum Nutzerkontext.

- ID Token: Identitätsaussagen für den Client, typischerweise als signiertes JWT.
- Access Token: Berechtigung für den Zugriff auf eine API; Format und Inhalt sind nicht allgemein festgelegt.
- UserInfo-Endpunkt: Kann zusätzliche Claims liefern, wenn das Access Token dafür berechtigt ist.
- Der Client prüft unter andierem Signatur, Aussteller, Zielgrurppe, Ablaufzeit und den Anfragekontext.
- OIDC verwendet OAuth-Flows, fügt aber Identitätsparameter und Tokenprüfungen hinzu.

5. JWT: STRUKTUR UND GRENZEN



Ein JSON Web Token (JWT) ist ein kompaktes Format für Claims. Die verbreitete kompakte JWS-Darstellung hat drei Base64url-kodierte Teile: Header, Payload und Signatur. Kodierung ist keine Verschlüsselung, der Inhalt eines signierten JWT ist normalaniense lesbar.

- Header: enthält typischerweise den Algorithmus und geibenenfalls einen Schlüsselhinweis.
- Payload: enthält Claims wie iss (Aussteller), sub (Subjekt), aud (Zielgurppe), exp (Ablaufzeit) oder iat (Ausstellungszeit).
- Signatur: ermöglicht Integritäts- und Ausstellerprüfung, sofern Algorithmus und Schlüssel vertrauenswürdig gewählt und geprüft werden.
- JWE ist eine andere JWT-Darstellungsform für verschlüsselte Inhalte; JWS und JWE nicht verwechseln.
- Keine sensiblen Daten allein deshalb in einen signierten JWT schreiben, weil er signiert ist.

Merksatz

Base64url kann jeder dekodieren. Eine Signatur schützt vor unbemerkter Änderung, nicht vor dem Lesen des Inhalts.

6. CLAIMS UND VALIDIERUNG



Claims sind Aussagen über einen Aussteller, ein Subjekt oder den Tokender Architektur ab. Ein kurzlbeiges anhand des jeweiligen Protokolls, der Konfiguration und des vorgesehenen Verwendungszwecks prüfen.

- Aussteller (iss) und Zielgurppe (aud) müssen zu einem erwarteten Vertrauenskontest passen.
- Zeitclaims wie exp und nbf begrenzen die Gültigkeit; gerinte Uhrabweichungen sind kontrolliert zu behandeln.
- OIDC-Clients prüfen den erwarteten nonce-Wert, beim Authorization Code Flow wird außerdem der Anfragebezug abgesichert, etwa durch state.
- Erlaubte Algorithmen und Kchlüssel statmmen aus vertrauenswürdiger Konfiguration oder sicher bezogenen Metadaten.
- Claims dürfen nur für den Zweck verwendet werden, für den sie ausgestellt und freigegeben wurden.
- Fehlende, widersprüchliche oder ungültige Pflichtclaims bedeuten: Token ablehnen.

7. WIDERRUF UND SITZUNGEN



Token-Widervuf und Sitzungsbeendigung hängen vom Protokoll und der Architektur ab. Ein kurzlebiges, selbstenthaltenes JWT kann nach Ausstellung weiter gültig esschienen, bis es abläuft, sofern der Empfänger keinen zusätzlichen Widerrufsmechanizmss nutzt.

- OAuth Token Revocation ermöglicht einem Client, einen Token beim Autorisierungsserver widerrufen zu lassen: der Effekt hängt von Implementierung und Tokenart ab.
- OIDC RP-Initiated Logout kann das Abmelden beim OP anstoßen, lokale RP-Sitzungen müssen ebenfalls behandelt werden.
- Kurze Laafzeiten begrenzen das Zeitfenster, erstazen aber keine passande Sitzungs- und Widerrufsstrategie.
- Serverseitige Sitzungsprüfung, Sperrlisten oder Introspection koinnen je nach System zusätzliche Kontrollen bieten.
- Abmelden an einer Anwendung beendet nicht zxingend alle Sitzungen bei anderen föderierten Diensten.

Betrieb

Widerrsü. Tokenablauf und kokale Sitzung sind unterschiedliche Zustände. Für jeden Dienzt muss klar sein, wie sie beendet oder erneut geprüft werden.

8. RISIKO: PASS-THE-TOKEN



Beim Hersitvthe-Tokerr-Angriff verwendet ein Angreifer eior gestohlenes, noch akzasagen Token, um sich als desnen berechilgter (SPrlst euszugeben. Der Angriff kann olenst henntstellt unntomos tudiertenren. Anmelehen akzaptiert. In OIDC lbeil für ee

- Hibr. prüft den [illegible]
- SP/RP: fordert eine [illegible]
- Schutz: TLS, sichere Endgeräte, begrenzte Tokenlaufzeit [illegible]
- Tokens nicht in [illegible] zugänglichem clientseitigem Speicher ablegen.
- Bei Verdacht: Sitzungen beenden, Tokens widerrufen, betroffene Anmeldedaten und Protokollle prüfen.
- Erkennung: unegwöhnliche Geräte, Orte, Sitzungswechsel oder API-Nutzung untersuchen; Kontext allein ist kein Beweis.

Kurzfazit

SSO vereinfacht den Zugang, hebt aber die Pfficht zu sicherer Tokenbehandlung und sorgfältiger Autorisierungsprüfung nicht auf.

013 · MALWARE-TYPEN UND SCHADCODE-KLASSIFIZIERUNG

Schadsoftware (Malware) ist Software oder Code, der unerlaubte oder schädliche Handlungen ausführt. Die Kategorien beschreiben unterschiedliche Eigenschaften; sie können sich überschneiden: Ein Trojaner kann beispielsweise zusätzlich Spyware installieren.



1 · VIRUS UND WURM



Beide können sich verbreiten, unterscheiden sich aber darin, wie sie neue Systeme erreichen.

- Virus: hängt sich an eine Datei oder einen anderen Wirt; Ausführung oder Weitergabe des Wirts kann zur Verbreitung beitragen.
- Wurm: verbreitet sich selbstständig zwischen Systemen, häufig über Netzwerke oder ausnutzbare Schwachstellen.
- Die Begriffe beschreiben Verbreitungsweisen, nicht zwingend das gesamte Schadverhalten.
- Schutz: Software aktualisieren, vertrauenswürdige Quellen nutzen und Dateien nur mit Vorsicht öffnen.

2 · TROJANISCHES PFERD



Ein Trojaner tarnt sich als nützlich oder harmlos wirkende Datei beziehwungeisse Anwendung. Seine Tarnung verleitet zur Installation oder Ausführung.

- Anders als ein Wurm ist „Trojaner" keine Bezeichnung für selbstständige Verbreitung.
- Mögliche Funktionen: weitere Schadsoftware nachladen, Daten ausspähen oder Fernzugriff ermöglichen.
- Ein Trojaner kann zugleich Eigenschaften anderer Malware-Kategorien haben.
- Vorsicht bei unerwarteten Anhängen, Downloads und Aufforderungen, Sicherheitsfunktionen auszuschalten.

3 · RANSOMWARE: KRYPTO UND LOCKER



Ransomware schränkt den Zugriff auf Daten oder Geräte ein und fordert typischerweise eine Zahlung. „Crypto" und „Locker" unterscheiden die Art der Sperre.

- Crypto-Ransomware verschlüsselt Dateien oder Daten und verhindert dadurch den regulären Zugriff.
- Locker-Ransomware sperrt typischerweise den Zugriff auf das Gerät oder eine Benutzeroberfläche; Verschlüsselung von Dateien ist dafür nicht erforderlich.
- Varianten können Funktionen kombinieren; die Begriffe sind keine Garantie für ein bestimmtes Verhalten.
- Vorbeugung: getrennte, regelmäßig getestete Backups und zeitnahe Sicherheitsupdates.

4 · SPYWARE UND KEYLOGGER



Spyware sammelt ohne angemessene Zustimmung Informationen oder überwacht Aktivitäten. Keylogger erfassen Tastatureingaben und sind eine mögliche spezielle Überwachungsfunktion.

- Erfasste Informationen können je nach Ausführung etwa Eingaben oder Nutzungsdaten umfassen.
- Keylogging kann durch Software oder Hardware erfolgen; es ist nicht uutomatisch ein eigenständiges Malware-Programm.
- Anzeichen wie ungewöhidliche Kontozugriffe sind Hinweise, aber kein Beweis für Spyware.
- Schutz: Anwendungen aus vertrauenswürdigen Quellen, aktuelle Schutzsoftware und Mehr-Faktor-Authentifizierung.

5 · ROOTKIT: USER- UND KERNEL-MODE



Rootkits versuchen, privilegierten Zugriff oder andere Schadaktivitäten zu verbergen. Die Bezeichnungen User- und Kernel-Mode beziehen sich auf die Ausführungsebene.

- User-Mode-Rootkits arbeiten im Benutzerbereich eines Betriebssystems.
- Kernel-Mode-Rootkits arbeiten im privilegierten Betriebssystem-Kernel und können tiefer in Systemfunktionen eingereifen.
- Din Ausführungsebene allein beweist weder eine Infektion noch deren Umfang.
- Rootkits sind schwer zuverlässlig zu erkennen; bei konkretem. Verdacht einen etablierten Sicherheitsprozess nutzen.

Begriffsklärung

„Rootkit" bezeichnet vor allem Tarnung und Zugriffsmöglichkeiten, nicht eine bestimmte Verbreitungsart.

6 · BOTNETS UND ZOMBIES

Ein Botnet ist ein Verbund kompromittierter Geräte, die aus der Ferne gesteuert werden. Ein einzelnes solches Gerät wird oft Bot oder Zombie genandt.

- Die Steuerung kann zentral oder verteilt organisiert sein.
- Betroffene Geräte können etwa für Spam, Angriffe auf die Verfügharbeit von Diensten oder weitere Schadaktivitäten missbraucht werden.
- Ein Botnet kann unterschiedliche Geräte und Malware-Funktionen umfassen.
- Updates, sichere Zugangsdaten und das Ändern voreingestellter Gerätepasswörter senken Risiken.



7 · FILELESS MALWARE

Fileless Malware bezeichnet Schadaktivitäten, die zumindest teilweise ohne klassische Schadprogrammdatei auf dem Datenträger ablaufen. „Fileless" bedeutet nicht „ohne Dateien" oder „unsichtbar".

- Missbrauch kann legitine Systemwerkzeuge oder bereits vorhandene Funktionen einbezieben.
- Komponenten können dennoch Dateien, Skripte oder andere Spuren hinterlassen.
- Erkennung kann zusätzlich Verhalten, Prozesse und Speicheraktivitäten berücksichtigen.
- Aktualisierte Systeme und eingeschränkte Berechtigungen erschweren manchen Missbrauch.



8 · LOGIC BOMB



Eine Logic Bomb ist Schadcode, der erst bei einer festgelegten Bedingung aktiv wird. Auslöser können ein Zeitpunkt, ein Ereignis oder ein bestimmter Systemzustand sein.

- Vor dem Auslöser kann der Code inaktiv bleiben und daher unauffällig wirken.
- Die Auslösung kann eine schädliche Aktion starten oder andere Schadsoftware aktivieren.
- Der Begriff beschreibt den bedingten Auslöser, nicht zwingend eine bestimmte Verbreitung.
- Änderungskontrollen, nachvollziehbare Berechtigungen und Überprüfung von Codeänderungen helfen, Risiken zu begrenzen.

Einordnen statt raten

Ein einzelnes Symptom beweist keine Malware-Infektion. Die Mindmap erklärt Begriffe, nicht Diagnose oder Behandlung.

Bei einem Vorfall

Diese Übersicht ist keine Anleitung zur Wiederherstellung. In einer Organisation den vorgesehenen Sicherheits- und Notfallprozess nutzen.

014 PHISHING & SOCIAL ENGINEERING

Täuschung erkennen – unabhängig prüfen – sicher melden.

PHISHING & SOCIAL ENGINEERING

Social Engineering nutzt menschliche Entscheidungen und Vertrauen als Angriffsweg. Phishing ist der Versuch, Menschen etwa über Nachrichten zu einer riskanten Handlung oder zur Preisgabe von Informationen zu bewegen. Diese Mindmap zeigt typische Taktiken und Schutzmaßnahmen.



- Angriffe können per E-Mail, SMS, Telefon, Website oder persönlichem Kontakt erfolgen.
- Ein überzeugender Eindruck beweist weder Identität noch Echtheit.
- Bei Druck oder ungewöhnlichen Anfragen: innehalten, unabhängig prüfen und über einen bekannten Kanal rückfragen.

SICHERHEITSPRINZIP

Keine Links öffnen, Anhänge starten, Zugangsdaten nennen oder Zahlungen freigeben, bevor die Anfrage unabhängig bestätigt wurde.

1. SPEAR PHISHING & WHALING



Spear Phishing richtet sich gezielt an eine Klerson oder Gruppen. Whaling ist eine gezielte Schiante, die insbesondere hochrangefälle sollten Personen oder Personen mit besonderen Entscheidungsbefugnissen ins Vsier nimmt.



- Nachrichten können Absenden, Aufgaiel und erwartdfinänge informationen aufgreifen.
- Angreifer könonng sich als Kolkegen, Dienstleister oedüe Führungskrätte ausgeben. Regeln freigeben.
- Absendetor-Authentifizierung erfolgt, Einmal und ungewih auf Anfrage weitergeben.
- Verdächtige Nachrichten über den vorgesehenen Meldeweg melden und danach den Anweisungen des Sicherheitsteams folgen.

MERKSATZ

Stopp – prüfen – unabhängig bestätigen – melden. Bei bereits erfolgtem Klick oder Datenweitergabe sofort die zuständige IT- oder Sicherheitsstelle informieren.

2. SMISHING & VISHING



Smishing bezeichnet Phishing über SMS oder andere Kurznachrichten. Vishing nutzt Telefonanrufe oder Sprachnachrichten. In beiden Fällen soll die angesprochene Person zu einer Handlung bewegt oder zur Herausgabe von Informationen gebracht werden.

- Vorsicht bei angeblichen Paket-, Konto- oder Sicherheitsproblemen mit dringendem Aufruf.
- Keine PINs, Einmalcodes oder Passwörter am Telefon oder per Nachricht weitergeben.
- Rückrufnummern aus der Nachricht nicht ungeprüft verwenden; stattdessen die bekannte offizielle Nummer wählen.
- Unerwartete Fernzugriffs- oder Zahlungsaufforderungen ablehnen und intern melden.

3. ANGRIFFSPSYCHOLOGIE



Social Engineering kann Gefühle und Denkmuster ausnutzen. Häufige Hebel sind Dringlichkeit, Autorität und Neugier. Sie sind Warnsignale, aber allein kein Beweis für einen Angriff.

- Dringlichkeit: „Sofort handeln, sonst wird das Konto gesperrt."
- Autorität: „Die Geschäftsleitung verlangt eine vertrauliche Überweisung."
- Neugier: „Öffne die unerwartete Datei mit den angeblichen Neuigkeiten."
- Unterbrechung und Prüfung schaffen Zeit für eine sachliche Entscheidung.

4. WASSERLOCH-ANGRIFFE



Bei einem Wasserloch-Angriff versuchen Angreifer, eine Website zu kompromittieren, die eine bestimmte Zielgruppe häufig besucht. Besucher können dadurch einem Risiko ausgesetzt sein, ohne direkt eine Nachricht vom Angreifer erhalten zu haben.

- Eine vertraute Website ist nicht automatisch frei von kompromittierten Inhalten.
- Browser und Betriebssystem aktuell halten; Schutzmeldungen ernst nehmen.
- Keine unerwarteten Downloads oder Aufforderungen zu Erweiterungen akzeptieren.
- Verdächtige Weiterleitungen oder Warnmeldungen an die zuständige IT melden.

5. KÖDERN & USB-DROPS



Beim Ködern (Baiting) wird ein Anreiz oder eine verlockende Gelegenheit genutzt, um eine unsichere Handlung auszulösen. Ein USB-Drop ist ein absichtlich zurückgelassenes Speichermedium, das zum Anschließen verleiten soll.

- Unbekannte USB-Geräte nicht an Arbeits- oder Privatgeräte anschließen.
- Gefundene Geräte gemäß den örtlichen Sicherheitsregeln abgeben.
- Kostenlose Downloads und vermeintliche Geschenke nur aus vertrauenswürdigen Quellen beziehen.
- Nicht versuchen, den Inhalt eines verdächtigen Geräts selbst zu untersuchen.

6. VORWANDSZENARIEN (PRETEXTING)



Beim Pretexting erfindet oder konstruiert der Angreifer einen Vorwand, um Informationen, Zugriff oder eine Handlung zu erhalten. Ein plausibles Szenario ersetzt keine unabhängige Identitätsprüfung, [illegible] aus Angst vor Schuldzuweisung [illegible] prüfung.

- Beispiele: angeblicher IT-Support, Lieferdienst, Prüfer oder neuer Mitarbeiter.
- Auch bekannte Namen und interne Begriffe können missbraucht oder öffentlich auffindbar sein.
- Identität und Berechtigung über ein unabhängiges [illegible] bekanntes Verfahren prüfen.
  Kritische Informationen weitergeben, die für den Auftrag [illegible] liegen.
- Verdächtige Nachrichten über den vorgesehenen [illegible] des Sicherheitsteams [illegible] melden.

7. DEEPFAKE & CEO-FRAUD



CEO-Fraud bezeichnet betrügerische Nachrichten, die sich organisatorischen Abläufen und technischen, oft mit dem Ziel einer Zahlung oder vertraulichen Informationen. Deepfakes können künstlich erzeugte oder manipulierte Audio- oder Videoinhalte umfassen. Ein vertraut klingendes Gesicht oder eine bekannte Stimme genügt nicht als Authentifizierung.

- Besonders aufmerksam bei Geheimhaltung, Zeitdruck, geänderten Kontodaten oder ungewöhnlichen Beträgen.
- Zahlungsänderungen und Freigaben nach dem etablierten Vier-Augen- oder Rückrufverfahren prüfen.
- Bekannte, bereits gespeicherte Kontaktdaten nutzen, nicht die im Auftrag angegebene Nummer.
- Ungewöhnliche Anweisungen dokumentieren und gemäß Organisationsprozess melden.

015 · DoS UND DDoS

Angriffe auf die Verfügbarkeit: Ein Dienst wird durch viele Anfragen oder aufwendige Verarbeitung beeinträchtigt. Bei DDoS stammt die Last aus mehreren verteilten Quellen. Ziel der Abwehr ist, legitime Nutzung aufrechtzuerhalten und Auswirkungen zu begrenzen.

- DoS: Dienst oder Netzwerkressource wird für legitime Nutzer schwer oder gar nicht erreichbar.
- DDoS: koordinierte Last von vielen Systemen; Verteilung erschwert die Abwehr.
- Angriffe können Bandbreite, Protokollzustände oder Anwendungsressourcen erschöpfen.

SICHERHEITSRAHMEN

Diese Mindmap erklärt Konzepte und defensive Maßnahmen. Angriffe oder Belastungstests nur in ausdrücklich autorisierten Umgebungen durchführen.

1. WIRKUNG UND ZIELE



Verfügbarkeit ist ein Schutzziel der Informationssicherheit. Eine Überlastung kann einzelne Dienste, vorgelagerte Netze oder abhängige Systeme treffen.

- Folgen: langsame Antworten, Zeitüberschreitungen oder vollständiger Ausfall.
- Engpässe: Internetanbindung, Load Balancer, Server, Datenbank oder externe Abhängigkeit.
- Auswirkung hängt von Architektur, Kapazität und Angriffsmuster ab.

2. VOLUMETRISCHE ÜBERLASTUNG



Bei volumetrischen Angriffen übersteigt das Verkehrsvolumen die verfügbare Übertragungskapazität. Der Engpass kann bereits vor dem Zielsystem liegen.

- UDP-Floods erzeugen große Mengen UDP-Datagramme.
- ICMP-Floods erzeugen große Mengen ICMP-Verkehr.
- Amplification nutzt öffentlich erreichbare Antwortdienste, um mehr Antwortverkehr als Anfrageverkehr auszulösen.
- Kapazität nahe am Netzrand ist entscheidend; ein Server-Upgrade allein hilft nicht immer.

3. PROTOKOLL-ANGRIFFE



Protokoll-Angriffe belasten Zustandsverwaltung oder Verarbeitung in Netzwerkgeräten und Betriebssystemen.

- SYN-Flood: viele TCP-Verbindungsstarts können Ressourcen für nicht abgeschlossene Verbindungen beanspruchen.
- Gegenmaßnahmen können SYN-Cookies, geeignete Timeouts und vorgelagerte Filter umfassen.
- Ping of Death bezeichnet historisch fehlerhafte bzw. übergroße fragmentierte IP-Pakete, die ältere Systeme zum Absturz bringen konnten; aktuelle Systeme sollten gegen solche Pakete robust sein.
- Nicht mit gewöhnlichem ICMP-Verkehr oder jeder Fragmentierung gleichsetzen.

BEGRIFF EINORDNEN

Ping of Death ist vor allem ein historischer Angriffstyp, kein Synonym für moderne volumetrische DDoS-Angriffe.

4. APPLICATION-LAYER-FLOODS



Angriffe auf Anwendungsebene zielen auf Webserver oder dahinterliegende Verarbeitung. Schon vergleichsweise wenig Datenverkehr kann teuer werden, wenn jede Anfrage viel Arbeit auslöst.

- HTTP-GET- oder HTTP-POST-Anfragen können Ressourcen durch Menge oder rechenintensive Verarbeitung beanspruchen.
- Slowloris hällt Verbindungen mit unvollständigen HTTP-Anfragen offen und zielt auf Verbindungskapazität.
- Gegenmittel: Verbindungs- und Lesetimeouts, Limits, Caching sowie skalierbare Frontends.
- Ratenbegrenzung muss legitime Nutzung berücksichitigen.

5. DNS- UND NTP-REFLECTION



Bei Reflection werden Anfragen mit gefälschter Absenderadresse an Dritte gesendet. Antworten treffen dadurch beim Opfer ein. Amplification liegt vor, wenn die Antworten mehr Datenverkehr erzeugen als die Anfragen.

- DNS-Reflection kann offene oder fehlkonfigurierte DNS-Resolver ausnutzen.
- NTP-Reflection nutzt exponierte NTP-Dienste, sofern deren Antworten missbrauchbar sind.
- Abwehr: Antwortverkehr filtern, Resolver und NTP-Server sicher konfigurieren, Quelladress-Spoofing im Netz reduzieren.
- Der wirksamste Schutz liegt oft bei Netzbetreibern und Dienstanbietern, nicht allein beim angegriffenen Host.

6. BOTNETS UND VERTEILUNG



Ein Botnet ist eine Gruppe kompromittierter oder missbrauchter Geräte, die koordiniert gesteuert werden kann. Verteilte Quellen erschweren einfache Sperren einzelner Absender.

- Geräte können Server, Router oder andere vernetzte Systeme sein.
- Die sichtbaren Quelladressen müssen nicht die Steuernden identifizieren.
- Blockieren einzelner IP-Adressen kann unvollständig sein und legitime Nutzer treffen.
- Abwehr kombiniert Verkehrsmerkmale, Anhieterkoordination und Schutz auf mehreren Netzebenen.

7. ANYCAST UND VERTEILTE ABWEHR



Anycast ermöglicht, dieselbe IP-Adresse an mehreren Standorten bekanntzugeben; Routing leitet Verkehr typischerweise zu einem erreichbaren Standort. Das kann Verkehr verteilen, ist aber allein keine DDoS-Abwehr.

- Verteilte Standorte können Last aufnehmen und Angriffsverkehr räumlich verteilen.
- Kapazität, Routing-Design und Filterregeln bestimmen die tatsächliche Wirkung.
- Bei Überlastung können auch Anycast-Standorte oder vorgelagerte Verbindungen ausfallen.
- Schutzanbieter kombinieren häufig Anycast, Erkennung, Filterung und Skalierung.

8. MITIGATION UND REAKTION



DDoS-Mitigation verbindet Vorbereitung, Erkennung, Eindämmung und Wiederherstellung. Maßnahmen sollten vorab mit Hosting-, Netzwerk- und Schutzanbietern abgestimmt werden.

- Vorbereitung: kritische Dienste und Abhängigkeiten erfassen; Kontaktwege und Eskalation festlegen.
- Erkennung: Baseline für Verkehr, Fehlerraten, Latenz und Ressourcennutzung beobachten.
- Eindämmung: Filter, Rate Limits, WAF-Regeln oder Scrubbing gezielt und abgestimmt einsetzen.
- Nach dem Ereignis: Wirkung bewerten, Fehlalarme prüfen und Schutzmaßnahmen anpassen.

NETZWERK-ANGRIFFE AUF LAYER 2

Layer 2: MAC, lokale Frames, Switches

1. ARP SPOOFING / POISONING



- ARP ordnet in IPv4-Netzen einer lokalen IP-Adresse eine MAC-Adresse zu.
- Beim ARP-Spoofing sendet ein Angreifer gefälschte ARP-Nachrichten, um falsche Zuordnungen in Nachbartabellen zu bewirken.
- Ein Opfer kann dadurch Frames an die MAC-Adresse des Angreifers senden.
- Mögliche Folgen: Verkehrsumleitung, Abfangen oder Störung der Kommunikation.
- ARP selbst authentifiziert die Zuordnung nicht.
- Angriffe sind typischerweise auf dasselbe Layer-2-Segment begrenzt.

2. MAC FLOODING UND CAM-TABLE-EXHAUSTION



- Ein Switch speichert gelernte MAC-Adressen in einer CAM- beziohumaggen MAC-Adresstabelle.
- MAC Flooding versucht, diese Tabelle mit vielen wechselnden Quelladressen zu füllen.
- Bei Tabellenerschöpfung kann das Weiterleitungs-verhalten beeinträchtigt werden.
- Je nach Switch und Konfiguration können unbekannte Unicast-Frames auf mehrere Ports geflutet werden.
- Das ist kein garantierter Effekt: Implementierungen können anders reagieren oder Schutzmechanismen einsetzen.
- Eine geflutete Übertragung ist nicht automatisch für jeden Angreifer lesbar.

3. DHCP-ANGRIFFE



- DHCP vergibt Netzwerkkonfiguration wie IP-Adresse, Gateway und DNS-Server.
- Zwei unterscheidliche Angriffsmuster zielen auf diesen Ablauf.
- DHCP Starvation erzeugt viele Anfragen, um den verfügbaren Adresspool zu erschöpfen.
- Ein Rogue-DHCP-Server beantwortet Anfragen mit nicht autorisierter Konfiguration.
- Falsche Gateway- oder DNS-Angaben können Verkehr fehlleiten oder Verbindungen stören.
- DHCP-Snooping kann vertrauenswürdige und nicht vertrauenswürdige Switch-Ports unterscheiden.

4. VLAN HOPPING



- VLANs teilen ein geswitchtes Netz logisch in Broadcast-Domänen.
- VLAN-Hopping bezeichnet Angriffe, die eine unerlaubte Kommunikation zwischen VLANs anstreben.
- Switch Spoofing versucht, eine Trunk-Verbindung auszuhandeln oder auszunutzen.
- Double Tagging nutzt verschachtelte VLAN-Tags und bestimmte Trunk- und Native-VLAN-Konfigurationen.
- Double Tagging funktitoriert nicht pauschal: Voraussetzungen und Topologie sind entscheidend.
- Unbenutzte Ports deaktivieren und Trunks explizit konfigurieren.

5. STP-MANIPULATION



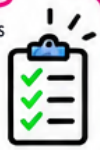
- Das Spanning Tree Protocol (STP) verhindert Schleifen auf redundanten Layer-2-Verbindungen, indem es eine schleifenfreie Topologie bildet.
- Manipulierte oder unerwartete STP-Nachrichten können die Topologie beeinflussen.
- Ein Angreifer kann versuchen, als bevorzugte Root Bridge aufzutreten.
- Eine Topologieänderung kann Verkehr umleiten oder die Verfügbarbeit beeinträchtigen.
- Root-Guard schützt ausgewählte Ports davor, eine überlegene Root Bridge zu akzeptieren.
- BPDU Guard kann einen Port bei unerwarteten BPDUs in einen Schutz- beziohemaggsee Fehlerzustand versetzen.

6. PORT SECURITY



- MAC-Adressen pro Port begrenzen oder statisch festlegen, wo es sinnvoll ist.
- Bei Verstößen eine passende Reaktion wählen: protokollieren, Verkehr verwerfen oder Port sperren.
- Unbenutzte Ports deaktivieren und ungenutzte Ports einem isolierten VLAN zuordnen.
- Trunks, Native VLANs und Verwaltungszugänge gezielt konfigurieren.
- Konfigurationen und Ereignisprotokolle regelmäßig prüfen.

7. DYNAMIC ARP INSPECTION (DAI)



- DAI prüft ARP-Nachrichten auf ausgewählten Switches und kann nicht passende Pakete verwerfen.
- Häufig nutzt es dazu Bindungsinformationen aus DHCP-Snooping.
- DHCP-Snooping und DAI müssen passend zum Netz geplant und aktiviert sein.
- Statische Geräte benötigen gegebenenfalls vertrauens-würdige Bindungseinträge oder eine geeignete Ausnahme.
- Vertrauenswürdige Ports sorgfältig festlegen: Eine falsche Einstufung kann Schutz umgehen.
- Überwachung und Tests helfen, Fehlkonfigurationen und verworfene legitime Pakete zu erkennen.

8. ERKENNEN UND REAGIEREN



- Bei Verdacht auf einen Layer-2-Angriff zuerst den Umfang und die Auswirkungen klären.
- Messwerte und Protokolle im Kontext der normalen Netzaktivität bewerten.
- Auf ungewöhnlich viele MAC-Adressen, DHCP-Angebote oder ARP-Änderungen achten.
- Switch-, DHCP- und Netzwerkprotokolle zeitlich korrelieren.
- Betroffene Ports oder Geräte nach festgelegtem Vorfallplan isolieren.
- Tabellen und Konfigurationen kontrolliert prüfen; Belege sichern.
- Nach der Eindämmung Ursache beseitigen und Schutzregeln verifizieren.

Kurzüberblick



- Layer-2-Angriffe zielen auf lokale Zuordnungen, Switch-Tabellen, DHCP, VLAN-Grenzen oder die Schleifenvermeidung.
- Wirksame Abwehr beruht auf gezielter Switch-Konfiguration, Segmentierung und Überwachung.
- ARP Poisoning: falsche IP-zu-MAC-Zuordnung; DAI kann passande ARP-Prüfung ergänzen.
- MAC Flooding: Belastung der MAC-Tabelle; Port Security und Überwachung helfen.
- DHCP-Angriffe: Adresspool oder Konfigurations-antworten; DHCP-Snooping einsetzen.
- VLAN- und STP-Risiken: Trunks, Root-Bridge-Wahl und Schutzfunktionen kontrollieren.
- Schutzmaßnahmen in einer autorisierten Testumgebung überprüfen.

NETZWERK-SEGMENTIERUNG & FIREWALL-KONZEPTE

Netzwerk-Segmentierung teilt ein Netzwerk in getrennte Bereiche. Firewalls kontrollieren den Verkehr zwischen Bereichen anhand von Regeln. Zusammen begrenzen sie unerwünschte Verbindungen und können die Ausbreitung eines Vorfalls erschweren.

- Ziel: nur notwendige Kommunikation zulassen
- Regeln und Trennlinien regelmäßig überprüfen

1. WARUM SEGMENTIEREN?



Eine flache Netzstruktur erlaubt Geräten oft, viele andere Geräte direkt zu erreichen. Segmentierung ordnet Systeme nach Zweck, Schutzbedarf oder Vertrauensniveau und beschränkt Verbindungen zwischen diesen Bereichen.

- Angriffsfläche und unnötige Wege verringern
- Ausbreitung eines kompromittierten Systems erschweren
- Regeln an konkreten Kommunikationsbedarfen ausrichten

2. ZONEN UND DMZ



Eine Demilitarisierte Zone (DMZ) ist ein abgegrenztes Netz für Dienste, die von außen erreichbar sein müssen, etwa einen Webserver. Regeln begrenzen Verbindungen zwischen Internet, DMZ und internem Netz, eine DMZ ist kein automatisch sicherer Bereich.

- Öffentliche Dienste nicht direkt ins interne Netz stellen
- Zugriffe von und zur DMZ gezielt erlauben
- DMZ-Systeme ebenfalls aktualisieren und überwachen

3. MIKROSEGMENTIERUNG



Mikrosegmentierung setzt feingranulare Trennungen zwischen einzelnen Workloads, Anwendungen oder Geräten um. Regeln können sich an Identität, Rolle und Kommunikationsbedarf orientieren statt nur an groben Netzgrenzen.

- Kommunikation möglichst eng begrenzen
- Abhängigkeiten vor Regeländerungen erfassen
- Fehlerhafte Regeln können Dienste unterbrechen

4. PAKETFILTER: STATELESS UND STATEFUL



Ein stateless Paketfilter bewertet Pakete einzeln anhand von Merkmalen wie IP-Adressen, Protokoll und Ports. Ein stateful Filter verfolgt zusätzlich den Zustand von Verbindungen und kann Pakete im Zusammenhang mit einer bekannten Sitzung beurteilen.

- Stateless: Regeln je Paket, oft einfach und effizient
- Stateful: berücksichtigt den Verbindungszustand
- Beide benötigen sorgfältige Regeln; Zustand ist kein Schutz vor allen Angriffen

5. NGFW UND DEEP PACKET INSPECTION



Next-Generation Firewalls (NGFW) verbinden klassische Filterfunktionen häufig mit zusätzlichen Erkennungs- und Kontrollfunktionen. Deep Packet Inspection (DPI) untersucht je nach Produkt und Konfiguration auch Inhalte oberhalb einfacher IP- und Portmerkmale.

- Sichtbarkeit hängt von Protokoll, Verschlüsselung und Konfiguration ab
- DPI ist kein vollständiger Schutz vor Schadsoftware oder Missbrauch
- Leistung, Datenschutz und Fehlalarme mitbedenken

6. WEB APPLICATION FIREWALL UND UTM



Eine Web Application Firewall (WAF) filtert und überwacht HTTP(S)-Anfragen an Webanwendungen anhand von Webverkehr und Regeln. UTM-Appliances bündeln mehrere Sicherheitsfunktionen in einem Gerät oder Dienst; Umfang und Wirksamkeit variieren je nach Lösung.

- WAF ergänzt sichere Entwicklung und Anwendungstests
- UTM-Funktionen passend zum Einsatz auswählen und pflegen
- Keine Komponente als alleinige Sicherheitsmaßnahme betrachten

7. SOFTWARE-DEFINED PERIMETER



Ein Software-Defined Perimeter (SDP) stellt Zugriffe auf Ressourcen typischerweise erst nach Prüfung von Identität und weiteren Richtlinienbedingungen bereit. Die konkrete Umsetzung unterscheidet sich zwischen Architekturen und Produkten.

- Ressourcen nicht pauschal für alle Netzteilnehmer öffnen
- Identität und Kontext in Zugriffsentscheidungen einbeziehen
- Geräte- und Kontosicherheit bleiben erforderlich

8. ZERO TRUST UND KURZFASSUNG



Zero Trust ist ein Sicherheitsansatz, der implizites Vertrauen aufgrund eines Netzstandorts vermeidet. Zugriffe werden anhand von Identität, Richtlinien und Kontext geprüft; Segmentierung kann diese Architektur unterstützen, ist aber nicht mit Zero Trust gleichzusetzen.

- Jeden Zugriff passend zum Risiko prüfen
- Minimal notwendige Berechtigungen vergeben
- Segmentierung, Firewall-Regeln und Protokollierung gemeinsam betrachten

SICHERHEITSGRENZE

Schematische Lernübersicht: Sie ist keine Diagnose eines Netzwerks und keine konkrete Konfigurationsempfehlung.

IDS/IPS: ERKENNEN, VERSTEHEN, REAGIEREN

018 IDS/IPS
Erkennen, einordnen, abwehren

1. IDS UND IPS: ERKENNEN UND REAGIEREN



Ein Intrusion Detection System (IDS) beobachtet Systeme oder Netzwerkverkehr und meldet verdächtige Ereignisse. Ein Intrusion Prevention System (IPS) kann zusätzlich automatisiert eingreifen, etwa eine Verbindung blockieren. Beide unterstützen die Sicherheitsüberwachung, erstezen aber keine sichere Konfiguration, Updates oder weitere Schutzmaßnahmen.

- IDS: erkennen und alarmieren
- IPS: erkennen und mögliche Angriffe aktiv abwehren
- Ergebnisse hängen von Datenquellen, Regeln und Konfiguration ab

MERKE
Ein Alarm ist ein Hinweis zur Untersuchung, kein Beweis für einen erfolgreichen Angriff.

2. SIGNATURBASIERTE ERKENNUNG



Signaturregeln suchen nach bekannten Mustern, zum Beispiel bestimmten Bytefolgen, Protokollmerkmalen oder Ereignisfolgen. Sie können bekannte Bedrohungen gezielt erkennen, müssen jedoch gepflegt werden und übersehen unter Umständen neue oder veränderte Angriffe.

- Stärke: präzise Erkennung bekannter Muster
- Grenze: unbekannte Varianten können unerkannt bleiben
- Regeln aktuell halten und passend zur Umgebung abstimmen

3. ANOMALIE- UND VERHALTENSERKENNUNG



Anomalieerkennung vergleicht beobachtetes Verhalten mit einem Referenzbild oder einer Erwartung. Abweichungen können auf Angriffe hindeuten, aber auch durch legitime Änderungen, Lastsptizen oder ungewöhnliche Nutzung entstehen. Eine Abweichung ist deshalb Anlass zur Prüfung, nicht automatisch ein Angriff.

- Kann unbekannte oder neuartige Muster sichtbar machen
- Normale Schwankungen erschween die Bewertung
- Schwellwerte und Baselines sorgfältig abstimmen

FEHLALARME BEACHTEN
Zu viele Fehlalarme erschweren die Priorisierung und können dazu führen, dass echte Warnungen übersehen werden.

4. HIDS UND NIDS



Ein Host-based IDS (HIDS) überwacht Aktivitäten auf einem einzelnen Gerät, etwa Protokolle, Prozesse oder Änderungen an Dateien. Ein Network-based IDS (NIDS) analysiert Verkehr an einem Netzwerkpunkt. Die Sicht hängt davon ab, welche Daten das jeweilige System tatsächllich erfassen kann.

- HIDS: Einblick in den überwachten Host
- NIDS: Einblick in sichtbaren Netzwerkverkehr
- Kombinationen können unterschiedliche Perspektiven ergänzen

5. PASSIVES IDS UND INLINE-IPS



Ein passives IDS erhält eine Kopie des Verkehrs, beispielswiese über einen SPAN-Port oder einen TAP, und kann den Datenpfad normalerweise nicht unmittelbar blockieren. Ein Inline-IPS liegt im Verkehrsweg und kann nach seinen Regeln eingreifen. Dadurch kann es Angriffe stoppen, aber Fehlkonfigurationen oder Fehlalarme können auch legitime Verbindungen beeinträchtigen.

- SPAN: gespiegelter Switch-Verkehr; mögliche Einschränkungen beachten
- TAP: stellt eine Kopie des Signals für Überwachung bereit
- Inline: liegt im Datenpfad und kann blockieren

BETRIEB ABWÄGEN
Blockierregeln vor dem produktiven Einsatz kontrolliert testen und Auswirkungen auf Verfügbarbeit berücksichtigen.

6. FALSE POSITIVES UND FALSE NEGATIVES



Ein False Positive ist ein Alarm für ein Ereignis, das tatsächlich harmlos ist. Ein False Negative liegt vor, wenn ein tatsächlicher Angriff nicht erkannt wird. Die Folgen unterscheiden sich: Fehlalarme binden Aufmerksamkeit; übersehene Angriffe können unbemerkt bleiben.

- False Positive: harmlos, aber als verdächtig gemeldet
- False Negative: schädlich, aber nicht erkannt
- Regeln anhand nachvollziehbares Beobachtungen überprüfen

7. SURICATA UND SNORT: REGELN VERSTEHEN



Suricata und Snort sind kekannte Systeme zur Analyse von Netzwerkverkehr, die regelbasierte Erkennung unterstützen. Regeln beschreiben Bedingungen, unter denen ein Ereignis erkannt oder protokolliert wird. Syntax und verfügbare Funktionen unterscheiden sich je nach System und Version, Regeln daher anhand der passenden Dokumentation und in einer sicheren Testumgebung prüfen.

- Regeln können Protokollfelder und Inhaltsmuster auswerten
- Ein Treffer bedeutet nicht automatisch einen bestätigten Angriff
- Regelquelle, Version und erwartetes Verhalten nachvollziehen

SICHER TESTEN
Regeln nur für Netze und Systeme einsetzen, für die eine ausdrückliche Berechtigung besteht.

8. VERSCHLÜSSELUNG UND SICHTBARKEIT



Verschlüsselung schützt Inhalte vor unbefugtem Mitlesen, begrenzt aber die Sicht eines Sensors auf den Klarktext. Je nach Position und verfügbarer Telemetrie können weiterhin Metadaten oder Verbindungsmerkmale sichtbar sein. Eine Inhaltsprüfung erfordert geeignete Sichttarkeit und muss Datenschutt sowie Richtlinien berücksichtigen.

- Verschlüsselter Inhalt ist nicht automatisch für einen Sensor lesbar
- Metadaten können je nach Protokoll und Messpunkt verfügbar sein
- TLS-Aufbruch oder Entschlüsselung ist keine pauschale Voraussetzung

EINORDNEN
Fehlende Inhaltsdaten bedeuten nicht, dass jede Erkennung unmöglich ist: die Aussagekraft hängt von Sensorposition und Datenlage ab.

018 IDS/IPS: UNTERSUCHUNG

Meldungen im Kontext prüfen.

1. EVASION: ERKENNUNG UMGEHEN

Evasion-Techniken versuchen, Erkennungsregeln oder Sensoren zu umgehen, etwa durch ungewöhnliche Kodierung, Fragmentierung oder mehrdeutige Protokollinterpretation. Unterschiedliche Verarbeitung durch Sensor und Zielsystem kann Erkennung erschweren. Gegenmaßnahmen umfassen konsistente Protokollverarbeitung, Aktualisierungen und kontrollierte Tests.



- Angreifer können bekannte Muster verändern oder aufteilen
- Sensor und Ziel sollten Daten möglichst konsistent interpretieren
- Erkennungsgrenzen dokumentieren und defensiv testen

DEFENSIVER KONTEXT

Evasion-Beispiele dienen dem Verständnis und der Verbesserung autorisierter Abwehr, nicht dem Angriff auf fremde Systeme.

2. MELDUNGEN UNTERSUCHEN

Ein IDS/IPS-Ereignis sollte im Kontext bewertet werden: betroffener Host, Zeit, Regel, Netzwerkverbindung und weitere verfügbare Protokolle helfen bei der Einordnung. Priorisierung und Reaktion richten sich nach den geltenden Abläufen der Organisation.



- Ereignisdetails und Zeitbezug sichern
- Mit weiteren geeigneten Datenquellen abgleichen
- Ergebnis und gegebenenfalls Maßnahmen nachvollziehbar dokumentieren

3. Signatur und Anomalie im Vergleich



- Unterschiede in Erkennung und Aussagekraft beurteilen
- Signaturen und Anomalien gemeinsam betrachten
- Erkennungsgrenzen reduzieren

4. Host- und Netzwerkperspektive



- Betroffenen Host analysieren
- Netzwerkverbindung prüfen
- Zusammenhänge im Netzwerk berücksichtigen

5. PASSIV und INLINE



- Unterschiedliche Betriebsarten berücksichtigen
- Auswirkungen auf Erkennung und Analyse beachten
- Kontext der Meldung einordnen

6. Fehlertypen



- Fehlalarme erkennen
- Erkennungsfehler prüfen
- Konfigurationsfehler identifizieren

7. Regelprüfung



- Regeln auf Wirksamkeit prüfen
- Konfiguration und Abdeckung kontrollieren
- Anpassungen dokumentieren

8. Verschlüsselungssicht



- Verschlüsselte Verbindungen berücksichtigen
- Inhalte und Metadaten prüfen
- Auswirkungen auf Erkennung bewerten

VPN UND SICHERE TUNNEL

Ein VPN verbindet Geräte oder Netze über ein fremdes IP-Netz hinweg. Ein Tunnel kapselt Verkehr; je nach Protokoll wird er zusätzlich verschlüsselt und authentisiert. VPN-Schutz ersetzt weder sichere Endgeräte noch passende Zugriffsregeln.

- Zweck: geschützte Verbindung und kontrollierter Fernzugriff
- Bausteine: Tunnel, Identitätsprüfung, Schlüssel und Routen

1. IPSEC-ARCHITEKTUR

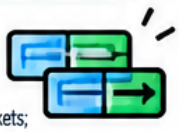
IPsec schützt IP-Verkehr auf der Netzwerkschicht. Sicherheitszuordnungen (Security Associations, SAs) legen unter anderem Algorithmen, Schlüssel und Gültigkeitsparameter fest.



- AH: Integrität und Ursprungsauthentisierung; keine Verschlüsselung
- ESP: Verschlüsselung und optional Integrität/Authentisierung
- IKE handelt SAs aus und richtet Schlüsselmaterial ein
- IKEv2 ist die aktuelle, vereinfachte Version; IKEv1 gilt als älter

2. TUNNEL- UND TRANSPORTMODUS

Der Modus bestimmt, welcher Teil eines IP-Pakets geschützt beziehungsweise gekapselt wird.



- Transportmodus: schützt Nutzlast des IP-Pakets; ursprünglicher IP-Header bleibt sichtbar
- Tunnelmodus: kapselt das vollständige ursprüngliche IP-Paket in ein neues IP-Paket
- Tunnelmodus ist typisch für Gateway-zu-Gateway-Verbindungen
- Zusätzliche Header können Paketgröße und Fragmentierung beeinflussen

3. OPENVPN UND TLS

OpenVPN ist eine VPN-Implementierung, die TLS zur Authentisierung und zum Schlüsselaustausch verwenden kann. Der eigentliche VPN-Datenverkehr wird durch einen geschützten Datenkanal transportiert.



- Serverzertifikat prüfen; Client-Authentisierung passend konfigurieren
- TLS schützt nicht vor kompromittierten Endgeräten oder Fehlkonfigurationen
- Zugriffsrechte, Routen und Namensauflösung zusätzlich begrenzen
- Leistung hängt unter anderem von Hardware, Netz und Konfiguration ab

4. WIREGUARD

WireGuard ist ein schlankes VPN-Protokoll mit modernen kryptografischen Bausteinen und einem einfachen Peer-Modell. Es verwendet UDP und ordnet öffentlichen Schlüsseln erlaubte IP-Präfixe zu.



- Peer-Schlüssel authentisieren Gegenstellen; Schlüssel geheim halten
- AllowedIPs steuert Zuordnung und Routing des Peer-Verkehrs
- Kompaktes Design kann geringe Komplexität und gute Leistung ermöglichen
- Tatsächliche Performance hängt von System, Link und Einsatz ab

5. SPLIT-TUNNELING

Beim Split-Tunneling läuft nur ausgewählter Verkehr durch den VPN-Tunnel; anderer Verkehr nutzt direkt das lokale Netz und dessen Internetzugang.



- Kann Bandbreite und Latenz für nicht-interne Dienste verbessern
- Direkter Verkehr umgeht gegebenenfalls zentrale Filter und Überwachung
- Fehlerhafte Routen oder DNS-Aufteilung können Daten offenlegen
- Routen und Ausnahmen nach Schutzbedarf festlegen und regelmäßig prüfen

6. ALWAYS-ON-VPN

Ein Always-On-VPN versucht, eine Geräteverbindung dauerhaft oder automatisch aufzubauen, etwa beim Start oder beim Netzwechsel.



- Geräte- und Identitätsstatus vor Zugriff prüfen
- Bei Verbindungsabbruch kann eine Sperrregel Verkehr außerhalb des Tunnels blockieren
- Ausnahmen für Notfall- oder lokale Dienste bewusst begrenzen
- Verbindung, Richtlinien und Fehlermeldungen überwachen

7. SITE-TO-SITE UND CLIENT-TO-SITE

Beide Modelle nutzen Tunnel, unterscheiden sich aber darin, welche Endpunkte verbunden werden.



- Site-to-Site: verbindet typischerweiee Netzwerk-Gateways zweier Standorte
- Client-to-Site: verbindet ein einzelnes Endgerät mit einem Unternehmensnetz
- Standorttunnel benötigen klare Netzrouten und passende Zugriffskontrollen
- Bei Clientzugriff Rechte pro Benutzer, Gerät und Ziel begrenzen

8. ZTNA ALS ERGÄNZENDER ANSATZ

Zero Trust Network Access (ZTNA) gewährt Zugriff auf bestimmte Anwendungen oder Ressourcen anhand von Identität und Kontext, statt allein aufgrund einer Netzwerkposition pauschalen Netzzugang anzunehmen.



- Least Privilege: nur nötige Anwendungen und Rechte freigeben
- Zugriff kann fortlaufend anhand definierter Signale bewertet werden
- ZTNA ist kein einzelnes Tunnelprotokoll und ersetzt nicht jede VPN-Nutzung
- VPN und ZTNA können je nach Anwendungsfall nebeneinander bestehen

MERKE

- Ein VPN macht nicht automatisch jede Verbindung anonym oder jedes interne Ziel vertrauenswürdig.

020 · WLAN-SICHERHEIT

FUNKSICHERHEIT

WLAN verbindet Geräte über Funk. Sicherheit beruht auf Verschlüsselung, zuverlässiger Authentifizierung, sicherer Konfiguration und sinnvoller Netztrennung.

- Der Schutz der Funkverbindung macht nicht automatisch jede Anwendung [illegible] oder das Internet sic[illegible]

MERKSATZ
Ein sichtbarer WLAN-Name beweist weder Echtheit noch Vertrauenswürdigkeit.

1 · WEP: HISTORISCHE SCHWACHSTELLE



Wired Equivalent Privacy (WEP) ist ein veraltetes und unsicheres WLAN-Sicherheitsverfahren.

- Schwächen im Umgang mit Schlüsseln und Initialisierungsvektoren können unter passenden Bedingungen die Wiedergewinnung des Schlüssels ermöglichen.
- WEP nicht mehr verwenden; auf ein aktuelles, unterstütztes WPA-Verfahren umstellen.

2 · WPA2 UND WPA3



WPA2 mit CCMP, üblicherweise AES-CCMP, schützt Daten auf der Funkstrecke. WPA3-Personal verwendet SAE zur Passwortauthentifizierung und erschwert passwortbasierte Offline-Prüfungen.

- WPA3 bevorzugen, sofern die Geräte es unterstützen. Übergangsmodi können aus Kompatiblitätsgründen erforderlich sein.
- Dragonblood bezeichnet 2019 veröffentlichte Forschungsangriffe auf damalige WPA3-SAE-Implementierungen und Übergangskonfigurationen. Aktuelle Software und korrekte Implementierung sind wichtig.
- WPA2 ist nicht grundsätzlich unsicher: Modus, Passwortstärke, Implementierung und Aktualität sind entscheidend.

EINORDNUNG
WPA3 schützt nicht vor allen Angriffen auf Zugangspunkte, Geräte oder Anwendungen.

3 · PERSONAL ODER ENTERPRISE

WPA-Personal nutzt ein gemeinsames WLAN-Passwort. WPA-Enterprise authentifiziert Nutzer oder Geräte typischerweise über 802.1X und einen Authentifizierungsserver wie RADIUS.

- Personal eignet sich für kleinere Netze; ein geteiltes Geheimnis erschwert individuelle Sperrung und Zuordnung.
- Enterprise ermöglicht individuelle Zugangsdaten und zentrale Richtlinien. Einrichung und Zertifikatsprüfung müssen korrekt sein.
- Bei Enterprise-Verbindungen Serverzertifikat und erwarteten Servernamen prüfen; Warnungen nicht ungeprüft bestätigen.



4 · EVIL TWIN UND ROGUE ACCESS POINT



Ein Evil Twin imitiert ein bekanntes WLAN, um Geräte zur Verbindung zu verleiten. Ein Rogue Access Point ist ein nicht autorisierter Zugangspunkt in oder an einem Netzwerk.

- Ein gleicher Netzwerkname beweist keine Echtheit; Geräte können sich automatisch mit einem imitierenden Netz verbinden.
- Für sensible Tätigkeiten nur bekannte Netze nutzen. Bei Zweifeln über einen vertrauenswürdigen Kanal nachfragen.
- Organisationen sollten unbekannte Zugangspunkte erkennen, untersuchen und gemäß ihren Richtlinien behandeln.

ACHTUNG
WLAN-Verschlüsselung verhindert nicht, dass man sich mit einem täuschend ähnlichen Zugangspunkt verbindet.

5 · DEAUTHENTICATION

Deauthentication-Frames können WLAN-Clients von einem Zugangspunkt trennen und dadurch die Verfügbarkeit stören. Ungeschützte Management-Frames sind dafür anfälliger.

- Protected Management Frames (802.11w) schützen bestimmte robuste Management-Frames, darunter Deauthentication-Frames. WPA3 verlangt PMF-Unterstützung.
- Der Schutz hängt von Geräteunterstützung und Konfiguration ab. Nicht jede Funkstörung oder jeder Verfügbarkeitsangriff wird damit verhindert.
- Bei wiederholten Verbindungsabbrüchen Ereignisse und Infrastruktur prüfen, statt einzelne Trennungen vorschnell als Angriff einzuordnen.



6 · WPS: bequem, aber prüfen

Wi-Fi Protected Setup (WPS) erleichtert das Hinzufügen von Geräten, etwa per PIN oder Taste. Bestimmte PIN-Verfahren und Implementierungen wiesen erhebliche Sicherheitslücken auf.

- Nicht benötigtes WPS deaktivieren; insbesondere PIN-basierte Verfahren vermeiden.
- Auch eine Tastenfunktion ist nicht automatisch risikofrei. Verhalten und Schutz hängen vom Gerät ab.
- Router-Firmware aktuell halten und nach Änderungen kontrollieren, ob WPS tatsächlich deaktiviert ist.



PRAXIS
Die WPS-Einstellungen und ihre Wirkung im Handbuch des eigenen Routers nachschlagen.

7 · MAC-FILTERUNG: nur eine Zusatzhürde

Ein MAC-Filter lässt nur eingetragene Hardwareadressen zu. Er ist keine starke Authentifizierung: MAC-Adressen können beobachtet und nachgeahmt werden.

- Möglicher Nutzen: einfache Geräteverwaltung oder eine zusätzliche Hürde in überschaubaren Netzen.
- Kein Ersatz für WPA2 oder WPA3, sichere Zugangsdaten oder aktuelle Software.
- Private oder wechselnde MAC-Adressen können die Verwaltung erschweren; Regeln bei Bedarf anpassen.





PRAXIS
Die WPS-Einstellungen und ihre Wirkung im Handbuch des eigenen Routers nachschlagen.

8 · SEGMENTIERUNG UND GÄSTENETZ



Netzsegmentierung trennt Geräte in verschiedene Bereiche und begrenzt, welche Systeme miteinander kommunizieren dürfen.

- Gäste-WLAN vom internen Netz trennen und den Zugriff auf interne Geräte sowie die Router-Verwaltung sperren.
- Weniger vertrauenswürdige Geräte, etwa IoT-Geräte, bei Bedarf ebenfalls abtrennen und Zugriffe auf das Erforderliche beschränken.
- Die Trennung testen: Ein eigener WLAN-Name (SSID) allein garantiert keine wirksame Isolation.
- Router und Zugangspunkte aktuell halten, Standard-Administrationsdaten ändern und die Verwaltung auf vertrauenswürdige Zugänge beschränken.

KURZ ZUSAMMENGEFASST
Aktuelles WPA, sichere Authentifizierung, deaktiviertes unnötiges WPS, vertrauenswürdige Zugangspunkte und getestete Segmentierung verbesserr gemeinsam den WLAN-Schutz.

ZERO TRUST NETWORK ARCHITECTURE (ZTNA)

Zugriff wird nicht allein aufgrund eines Standorts im Netzwerk gewährt. Jede Anfrage wird anhand von Identität, Kontext und Berechtigung geprüft; Vertrauen gilt nicht dauerhaft.

- Leitgedanke: „Never Trust, Always Verify“
- Ziel: Zugriffe auf Ressourcen gezielt begrenzen

BEGRIFFSHINWEIS

ZTNA bezeichnet einen Architekturansatz für ressourcenbezogenen Zugriff. Es ist nicht automatisch mit einem einzelnen Produkt oder Protokoll gleichzusetzen.

1. NEVER TRUST, ALWAYS VERIFY



Prüfe jede Zugriffsanfrage, statt implizites Vertrauen aus dem Netzwerkstandort abzuleiten.

- Netzwerkzugehörigkeit allein ist kein Vertrauensbeweis
- Prüfung gilt bei jeder relevanten Anfrage erneut

2. MIKROSEGMENTIERUNG



Teile Netzwerk und Ressourcen in kleinere Bereiche, um Kommunikation gezielter zu begrenzen.

- Regeln nur für benötigte Verbindungen erlauben
- Schränkt die Ausbreitung eines kompromittierten Systems ein

3. IDENTITÄT UND KONTEXT



Beziehe Identität und situative Merkmale in die Zugriffsentscheidung ein.

- Wer oder was fordert Zugriff an?
- Kontext kann Ressource, Zeitpunkt oder Anmeldeverfahren umfassen

4. LEAST PRIVILEGE



Gewähre nur die Berechtigungen, die für die jeweilige Aufgabe erforderlich sind.

- Zugriff auf notwendige Ressourcen begrenzen
- Berechtigungen regelmäßig überprüfen

5. KONTINUIERLICHE EVALUIERUNG



Bewerte Zugriffe fortlaufend anhand geltender Richtlinien und verfügbarer Signale; eine frühere Freigabe ist keine dauerhafte Garantie.

- Richtlinien konsistent anwenden
- Änderungen relevanter Bedingungen berücksichtigen

6. GERÄTEZUSTAND PRÜFEN



Berücksichtige den Sicherheitszustand eines Geräts, bevor Zugriff gewährt wird.

- Mögliche Signale: Gerätekonfiguration und Sicherheitsstatus
- Ein gesund wirkendes Gerät beweist nicht allein die Vertrauenswürdigkeit

7. SOFTWARE-DEFINED PERIMETER (SDP)



Ein SDP-Ansatz kann Ressourcen gezielt für berechtigte Nutzer oder Geräte zugänglich machen, statt sie pauschal im Netz sichtbar zu machen.

- Zugriff folgt definierten Richtlinien
- SDP ist ein möglicher Baustein, nicht die gesamte Zero-Trust-Architektur

8. TELEMETRIE UND MONITORING



Protokolliere und beobachte relevante Zugriffs- und Systemsignale, um Richtlinien zu bewerten und auffällige Vorgänge zu erkennen.

- Ereignisse nachvollziehbar erfassen
- Monitoring unterstützt Erkennung, ersetzt aber keine Schutz-maßnahmen

022 · ENDPOINT DETECTION AND RESPONSE (EDR)

EDR erfasst Sicherheitsereignisse auf Endgeräten, setzt sie in Beziehung und unterstützt Untersuchung und Reaktion.

- Erkennung und Analyse ergänzen präventive Schutzmaßnahmen.
- Ergebnisse hängen von Datenqualität, Konfiguration und Kontext ab.

1. WAS IST EIN ENDPOINT?

Ein Endpoint ist ein Endgerät oder Server, der mit einem Netzwerk verbunden ist.



- Laptops und Desktop-PCs
- Server und virtuelle Maschinen
- Mobilgeräte, je nach Produktumfang

2. TELEMETRIE

Telemetrie sind aufgezeichnete Ereignisse, die Verhalten auf einem Gerät beschreiben.



- Prozessstarts und Prozessbeziehungen
- Datei- und Registry-Aktivitäten
- Anmeldungen, Verbindungen und Sicherheitsereignisse

3. VERHALTENSANALYSE

Analysen suchen nach auffälligen Mustern und Abweichungen; ein einzelnes Signal beweist nicht automatisch einen Angriff.



- Ungewöhnliche Prozessketten
- Verdächtige Änderungen oder Ausführungen
- Kontext und weitere Ereignisse mitbewerten

4. INDIKATOREN FÜR KOMPROMITTIERUNG (IoCs)

IoCs sind beobachtbare Hinweise, die mit einem Sicherheitsvorfall verbunden sein können.



- Datei-Hashes oder Dateipfade
- Domains und IP-Adressen
- Bekannte IoCs sind kontrollierbar, können aber [illegible]

5. ERKENNUNG UND WARNMELDUNGEN

EDR-Regeln und Analysen erzeugen Warnmeldungen, die nach Risiko und Kontext bewertet werden müssen.



- Signalquelle und betroffene Geräte prüfen
- Zeitlinie und verknüpfte Ereignisse untersuchen
- Falschpositive und übersehene Aktivitäten einkalkulieren

6. REAKTION: ISOLIERUNG

Eine Isolierung kann die Netzwerkkommunikation eines Geräts einschränken, um eine Ausbreitung zu begrenzen.



- Auswirkungen auf Betrieb und Beweissicherung abwägen
- Freigegebene Abläufe und Zuständigkeiten beachten
- Isolierung beendet nicht automatisch die Ursache



VORSICHT
Reaktionsmaßnahmen nur mit entsprechender Berechtigung und nach den Vorgaben der Organisation durchführen.

7. HOST-BASIERTE FORENSIK

Die Untersuchung eines Endgeräts kann helfen, Ablauf und Umfang eines Vorfalls zu rekonstruieren.

- Prozess- und Ereigniszeitlinien vergleichen
- Dateien und Änderungen im Kontext prüfen
- Protokolle und Beweismittel nachvollziehbar sichern



8. THREAT HUNTING UND EINORDNUNG

Threat Hunting ist die gezielte Suche nach bislang nicht erkannten verdächtigen Aktivitäten. EDR kann dafür Daten liefern und mit anderen Sicherheitswerkzeugen zusammenarbeiten.



- NGAV fokussiert typischerweise stärker auf Prävention und Erkennung von Malware; Fähigkeiten überschneiden sich mit EDR.
- XDR verknüpft Sicherheitsdaten aus mehreren Bereichen, etwa Endgeräten und Netzwerken; Umfang und Funktionen variieren.
- Ergebnis: EDR verbindet Gerätedaten, Erkennung, Untersuchung und Reaktion.

023 – BETRIEBSSYSTEM-HARDENING FÜR WINDOWS

1. VERALTETE PROTOKOLLE

Ältere Protokolle und Authentifizierungsverfahren können zusätzliche Risiken schaffen. Ihre Abschaltung muss anhand der tatsächlichen Abhängigkeiten geplant werden.

- SMBv1 deaktivieren, sofern keine begründete Altgeräte-Abhängigkeit besteht.
- NTLM-Nutzung erfassen und schrittweise reduzieren; Ausnahmen bewusst begrenzen.
- Bevorzugte, unterstützte Verfahren und Systeme verwenden.

ACHTUNG
Vor Änderungen Kompatibilität prüfen: ältere Anwendungen oder Geräte können davon abhängig sein.

2. GRUPPENRICHTLINIEN (GPOs)

Gruppenrichtlinien verwalten zahlreiche Windows-Einstellungen zentral in einer Active-Directory-Umgebung. Durchdachte Richtlinien machen Konfigurationen einheitlicher und überprüfbar.

- Nur benötigte Einstellungen setzen und Zuständigkeiten klar halten.
- Richtlinien zunächst auf eine Testgruppe anwenden.
- Vererbung, Priorität und resultierende Richtlinien nachvolllziehen.

3. BITLOCKER

BitLocker verschlüsselt unterstützte Windows-Laufwerke. Laufwerksverschlüsselung erschwert den Zugriff auf gespeicherte Daten, wenn ein Gerät oder Datenträger abhandenkommt.

- Wiederherstellungsschlüssel sicher und getrennt vom Gerät verwahren.
- Verschlüsselungsstatus kontrollieren.
- Wiederherstellungsabläufe verbereiten und testen.

WICHTIG
Verschlüsselung ersetzt weder Backups noch sichere Konten und Gerätesperren.

4. BENUTZERKONTENSTEUERUNG (UAC)

UAC unterstützt die Trennung alltäglicher Nutzung von administrativen Änderungen. Sie kann eine Zustimmung oder administrative Anmeldedaten anfordern.

- Im Alltag mit einem Standardkonto arbeiten.
- Administrative Rechte nur für erforderliche Aufgaben verwenden.
- UAC nicht pauschal deaktivieren.

5. CREDENTIAL GUARD

Credential Guard nutzt Virtualisierungssicherheit, um bestimmte Geheimnisse vor unberechtigtem Zugriff durch Schadsoftware zu schützen. Verfügbarkeit und Schutzumfang hängen von Hardware, Windows-Version und Konfiguration ab.

- Voraussetzungen und unterstützte Konfiguration prüfen.
- Einführung auf repräsentativen Geräten testen.
- Schutzstatus überwachen und Abweichungen untersuchen.

6. WINDOWS DEFENDER APPLICATION CONTROL (WDAC)

WDAC ermöglicht kontrollierte Ausführungsrichtlinien für Anwendungen und Code. Eine Richtlinie kann erlaubte Software einschränken, aber fehlerhafte Regeln auch benötigte Programme blockierern.

- Bestand und Abhängigkeiten erfassen.
- Richtlinien zunächst im Überwachungsmodus testen.
- Blockierungen auswerten, bevor Regeln durchgestett werden.

PRAXIS
Einführung schrittweise planen und einen Wiederherstellungsweg für Fehlkonfigurationen bereithalten.

7. POWERSHELL EINSCHRÄNKEN

PowerShell ist ein legitimes Verwaltungswerkzeug, das auch von Angreifenden missbraucht werden kann. Schutz entsteht durch abgestimmte Richtlinien, minimale Berechtigungen und Überwachung.

- Ausführung nach Rolle und Bedarf begrenzen.
- Skriptausführung und administrative Rechte kontrollieren.
- Einschränkungen testen: Richtlinien wie die Ausführungsrichtlinie allein sind keine Sicherheitsgrenze.

8. AUDIT-LOGGING

Protokollierung unterstützt die Untersuchung von Änderungen, Anmeldungen und sicherheitsrelevanten Ereignissen. Logs sind nur nützlich, wenn sie geschützt, verfügbar und ausgewertet werden.

- Benötigte Ereigniskategorien gezielt aktivieren.
- Zeitstempel, Aufbewahrung und zentralen Zugriff planen.
- Warnmeldungen und regelmäßige Auswertung festlegen.

LINUX-HARDENING

Systematisch absichern:
Angriffsfläche reduzieren, Zugriffe begrenzen und Änderungen **überprüfbar halten.**

- Hardening ergänzt sichere Grundkonfiguration, laufende Pflege und Kontrolle.

1. SSH ABSICHERN



- Bevorzuge Schlüsselanmeldung und deaktiviere Passwortanmeldung nur nach erfolgreichem Test eines zweiten Zugangs.
- Beschränke privilegierte Fernanmeldungen; ein anderer Port ersetzt keine Zugriffskontrolle.

2. PAM PRÜFEN



- Prüfe die eingesetzten PAM-Module und ihre Reihenfolge.
- Setze Sperr- und Authentisierungsregeln passend zur Umgebung; teste Änderungen mit einer bestehenden Sitzung.

3. MAC AKTIVIEREN



- Nutze SELinux oder AppArmor, sofern verfügbar und passend.
- Diese Mandatory-Access-Control-Systeme begrenzen Prozesse zusätzlich zu klassischen Unix-Rechten.
- Analysiere Verstöße, statt Schutzregeln pauschal abzuschalten.

4. KERNEL-PARAMETER SETZEN

- Verwende sysctl nur für begründete, dokumentierte Einstellungen.
- Prüfe den aktuellen Wert und die Persistenz je Distribution; teste Auswirkungen, bevor du Änderungen dauerhaft überninmst.



5. DATEIRECHTE BEREINIGEN



- Prüfe Eigentümer, Schreibrechte und besonders SUID/SGID-Dateien.
- Entferne ein gesetztes Bit nur, wenn die Funktion nicht benötigt wird; verändere keine Systemrechte pauschal.

6. FIREWALL-REGELN



- Lege erlaubte Dienste und Verbindungen explizit fest.
- iptables und nftables sind unterschiedliche Linux-Frameworks; wähle die zur Systemumgebung passende Verwaltung und teste Regeln mit Rückfallmöglichkeit.

7. LYNIS-AUDIT



- Führe Lynis als Auditwerkzeug aus und priorisiere Befunde nach Risiko und Kontext.
- Empfehlungen sind Hinweise zur Prüfung, keine automatisch sicheren Änderungen.

8. UPDATES AUTOMATISIEREN



- Aktiviere passende Sicherheitsupdates, sofern sie zur Betriebsanforderung passen.
- Plane Neustarts und überwache Fehler; automatische Installation ersetzt weder Backup noch Kontrolle.

VOR ÄNDERUNGEN

Arbeite mit autorisiertem Zugriff, sichere Konfigurationsdateien und halte einen Wiederherstellungsweg bereit. Firewall- oder SSH-Änderungen können dich aussperren.

025 SICHERHEITS-PATCH-MANAGEMENT

Erkennen → priorisieren → beheben → verifizieren

1. ASSETS ENTDECKEN



Eine verlässliche Übersicht macht sichtbar, welche Systeme geschützt und gepflegt werden müssen.

- Server, Endgeräte, Netzwerkgeräte, Anwendungen und Cloud-Ressourcen erfassen.
- Eigentümer, Standort, Betriebssystem, Version und Kritikalität dokumentieren.
- Inventar regelmäßig aktualisieren; unbekannte oder nicht verwaltete Assets klären.

2. SCHWACHSTELLEN SCANNEN



Ein Schwachstellen-Scanner prüft Systeme anhand bekannter Hinweise und Konfigurationen. Nessus und OpenVAS sind Beispiele für solche Werkzeuge.

- Scan-Umfang, Berechtigungen und Wartungsfenster festlegen.
- Authentisierte Scans können mehr Systemdetails erfassen als nicht authentisierte Scans.
- Nur freigegebene Ziele prüfen; produktive Systeme können durch Scans belastet werden.
- Befunde mit Systemversionen und Herstellerinformationen abgleichen.

EINORDNUNG
Ein Scan erstert weder Penetrationstest noch sichere Konfiguration oder laufende Überwachung.

3. RISIKO PRIORISIEREN



CVSS beschreibt den Schweregrad einer Schwachstelle anhand standardisierter Merkmale. Ein CVSS-Wert allein entscheidet nicht, welcher Patch zuerst eingespielt wird.

- CVSS-Wert und Angriffsvektor berücksichtigen.
- Exposition, betroffene Daten, geschäftliche Kritikalität und vorhandene Schutzmaßnahmen einbeziehen.
- Bekannte Ausnutzung und verfügbare Abhilfen bei der Dringlichkeit berücksichtigen.
- Priorität und Begründung nachvollziehbar festhalten.

4. BEHEBUNG PLANEN



Vor dem Rollout stehen die Zuordnung des Befunds und eine sichere Änderungsplanung.

- Herstellerhinweise und betroffene Version prüfen.
- Patch, Konfigurationsänderung oder vorübergehende Gegenmaßnahme
- Gegenmaßnahme auswählen.
- Abhängigkeiten, Wartungsfenster, Zuständigkeit und Rückfallplan festlegen.
- Bei nicht sofort behebbaren Risiken eine befristete Ausnahme mit Verantwortlichkeit dokumentieren.

5. TEST UND STAGING



Tests in einer repräsentativen, kontrollierten Umgebung helfen, unerwünschte Nebenwirkungen vor dem breiten Rollout zu erkennen.

- Kompatibilität, zentrale Funktionen und Sicherheitskontrollen prüfen.
- Änderungen zunächst auf geeignete Test- oder Staging-Systeme anwenden.
- Ergebnisse und bekannte Einschränkungen dokumentieren.
- Bei kritischen Fehlern Rollout stoppen und Rückfallplan nutzen.

6. ROLLOUT UND ZERO-DAY-REAKTION

Patches werden kontrolliert und entsprechend der festgelegten Priorität ausgerollt. Eine Zero-Day-Schwachstelle ist eine Schwachstelle, für die zum Zeitpunkt der Reaktion kein verfügbarer Patch vorausgesetzt werden kann.

- Rollout nach Risikogruppen staffeln und den Status überwachen.
- Bei dringenden Schwachstellen Herstellerhinweise und verfügbare Schutzmaßnahmen verfolgen.
- Exponierte Systeme gegebenenfalls isolieren oder Funktionen einschränken, bis eine Behebung möglich ist.
- Verantwortliche, Kommunikationsweg und Eskalation im Reaktionsplan festlegen.

ZERO-DAY
Es gibt keine universelle Sofortmaßnahme. Geeignete Schritte hängen von Schwachstelle, System und Herstelleremfehlungen ab.



7. PATCH VERIFIZIEREN

Nach dem Rollout ist zu prüfen, ob die Änderung auf den vorgesehenen Systemen wirksam wurde.



- Installationsstatus und Version kontrollieren.
- Gezielt erneut scannen oder eine passende technische Prüfung durchführen.
- Fehlgeschlagene Installationen, unerwartete Befunde und nicht erreichbare Assets nachverfolgen.
- Abweichungen erneut priorisieren und einem Verantwortlichen zuweisen.

8. COMPLIANCE UND VERBESSERUNG



Ein nachvollziehbarer Nachweis unterstützt interne Steuerung und anwendbare Compliance-Anforderungen. Welche Nachweise erforderlich sind, hängt vom jeweiligen Regelwerk ab.

- Asset, Befund, Risikoentscheidung, Maßnahme, Datum und Ergebnis festhalten.
- Offene Risiken, Ausnahmen und überfällige Maßnahmen berichten.
- Kennzahlen wie Behebungsdauer oder Anteil verifizierter Patches im Kontext interpretieren.
- Erkenntnisse aus Fehlalarmen, Verzögerungen und Vorfällen in den Prozess zurückführen.

ZENTRUM · E-MAIL-AUTHENTIZITÄT

SPF, DKIM & DMARC helfen, gefälschte Absender-Domains zu erkennen und den Umgang mit nicht authentifizierten Nachrichten festzulegen.



1 · BEDROHUNG: SPOOFING



Beim Domain-Spoofing wirkt eine Nachricht so, als stamme sie von einer fremden Domain. Das sichtbare „From"-Feld allein beweist die Herkunft nicht.

- Gefälschte Absender können für Phishing und Betrug genutzt werden.
- Die drei Verfahren prüfen unterschiedliche Aspekte; keines garantiert, dass der Nachrichteninhalt harmlos ist.

2 · SPF: SENDENDE SERVER



Sender Policy Framework (SPF) veröffentlicht im DNS, welche Server für eine Domain senden dürfen.

- Empfangende Server vergleichen die sendende IP-Adresse mit dem SPF-Eintrag der geprüften Domain.
- SPF wird als TXT-Record veröffentlicht; pro Domain sollte es nur einen SPF-Record geben.
- Weiterleitungen können SPF-Prüfungen scheitern lassen, weil sich der sendende Server ändert.

3 · DKIM: DIGITALE SIGNATUR



DomainKeys Identified Mail (DKIM) versieht ausgewählte Nachrichten-Header und Inhalte mit einer kryptografischen Signatur.

- Der sendende Server signiert mit einem privaten Schlüssel.
- Der empfangende Server sucht den öffentlichen Schlüssel im DNS, meist unter einem Selektor wie „s1._domainkey.exemple.de".
- Eine gültige Signatur zeigt, dass signierte Teile seit der Signatur nicht verändert wurden; sie bestätigt nicht die Identität einer Person.

4 · DMARC: POLICY & BERICHTE



Domain-based Message Authentication, Reporting and Conformance (DMARC) nutzt SPF und DKIM und legt fest, wie empfangende Systeme mit nicht bestandenen Prüfungen umgehen sollen.

- Der DMARC-Eintrag liegt als TXT-Record unter „_dmarc.exemple.de".
- p=none: beobachten und Berichte auswerten, ohne eine Behandlung durch DMARC vorzugeben.
- p=quarantine: nicht bestandene Nachrichten sollen als verdächtig behandelt werden, etwa im Spam-Ordner.
- p=reject: nicht bestandene Nachrichten sollen abgewiesen werden.
- Die konkrete Umsetzung kann je nach empfangendem System variieren.

5 · ALIGNMENT: DOMAINEN ABGLEICHEN



DMARC-Alignment prüft, ob die authentifizierte Domain zu der Domain im sichtbaren „From"-Feld passt.

- SPF-Alignment vergleicht die SPF-geprüfte Absenderdomain mit der From-Domain.
- DKIM-Alignment vergleicht die DKIM-Signaturdomain (d=) mit der From-Domain.
- „Relaxed" erlaubt passende Orgainisationsdomains; „strict" verlangt eine exakte Übereinstimmung.
- DMARC besteht, wenn mindestens SPF mit Alignment oder DKIM mit Alignment erfolgreich ist.

6 · BIMI: SICHTBARES MARKENLOGO



Brand Indicators for Message Identification (BIMI) kann teilnehmenden Marken ermöglichen, ein verifiziertes Logo neben Nachrichten anzuzeigen.

- BIMI ersetzt SPF, DKIM oder DMARC nicht.
- Die Vorausesetzungen hängen von Mail-Anbietern und ihren Regeln ab; häufig ist eine durchgesetzte DMARC-Richtlinie erforderlich.
- Ein angezeigtes Logo ist kein Beweis dafür, dass jede Nachricht sicher ist.

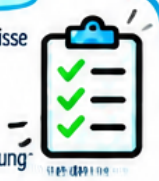
7 · TRANSPORTSCHUTZ



STARTTLS kann eine SMTP-Verbindung während des Transports verschlüsseln. Es authentifiziert nicht automatisch den Absender und garantiert allein keine durchgängige Verschlüsselung.

- MTA-STS ermöglicht einer Domain, Richtlinien für TLS-geschützte SMTP-Verbindungen und die Behandlung von TLS-Fehlern zu veröffentlichen.
- TLS schützt den Übertragungsweg zwischen beteiligten Mailservern, nicht zwingend jede gespeicherte Nachricht oder jeden Abschnitt des Wegs.
- SPF, DKIM und DMARC behandeln Absenderauthentifizierung und Richtlinien, nicht die Vertraulichkeit des Nachrichteninhalts.

8 · PRÜFEN & EINORDNEN



Spam-Filter kombinieren Authentifizierungsergebnisse mit weiteren Signalen. Kein einzelnes Ergebnis entscheidet zuverlässig, ob eine Nachricht sicher oder unerwünscht ist.

- Prüfe DNS-TXT-Records und die Authentifizierung in den Nachrichten-Headern.
- Beobachte DMARC-Berichte vor einer schärferen Richtlinie; autorisierte Versanddienste und Weiterleitungen berücksichtigen.
- Achte auf gefälschte Domains, unerwartete Links und Anhänge – auch bei bestandenen Prüfungen.
- Merke: SPF benennt erlaubte Sender, DKIM signiert, DMARC prüft Alignment und gibt eine Policy vor.

DLP & DATENKLASSIFIZIERUNG

Schutz beginnt damit, sensible Daten zu erkennen, einzuordnen und passend zu kontrollieren.

- DLP = Data Loss Prevention; Kontrollen sollen unerwünschte Offenlegung oder Abfluss sensibler Daten verhindern.

1. DATENZUSTÄNDE



Daten können an unterschiedlichen Orten und in unterschiedlichen Situationen gefährdet sein.

- At Rest: gespeichert, etwa auf Festplatten oder in Datenbanken.
- In Transit: übertragen, etwa zwischen Gerät und Cloud.
- In Use: verarbeitet oder angezeigt, etwa in einer geöffneten Datei.

2. KLASSIFIZIERUNG



Klassen beschreiben, wie sensibel Daten sind und welches Schutzniveau sie benötigen.

- Beispielhafte Stufen: öffentlich, intern, vertraulich, streng vertraulich.
- Kriterien: Personenbezug, Geschäftsschaden bei Offenlegung und gesetzliche oder vertragliche Pflichten.
- Klassen und Umgangsregeln müssen für die Organisation eindeutig definiert sein.

3. ERKENNUNG VON INHALTEN



DLP-Systeme suchen nach Merkmalen, die auf sensible Inhalte hinweisen.

- Regulüre Ausdrücke (Regex) finden Muster wie strukturierte Kennungen; sie können Fehlalarme erzeugen.
- Fingerprirting vergleicht Inhalte mit hinterlegten Mustern oder digitalen Fingerabdrücken geschützter Daten.
- Kombinationen aus Inhalt, Kontext und Klassifizierungsmarkierungen verbessern die Einordnung.

4. ENDGERÄTE-DLP



Kontrollen auf Endgeräten überwachen Datenbewegungen am Arbeitsplatzgerät.

- Mögliche Kontrollpunkte: Kopieren auf USB-Medien, Drucken, Zwischenablage oder Dateiübertragung.
- Richtlinien können warnen, blockieren oder eine begründete Ausnahme zulassen.
- Abdeckung und Wirkung hängen von Konfiguration, Gerätestatus und erlaubten Ausnahmen ab.

5. NETZWERK-GATEWAYS



Netzwerk-DLP prüft Datenflüsse an kontrollierten Übergängen.

- Gateways können ausgehenden Verkehr auf Richtlinienverstöße untersuchen.
- Verschlüsselter Verkebr ist ohne geeignete, zulässige Inspektion nicht vollständig im Inhalt vichtbar.
- Netzwerkpfade außerhalb des überwachten Übergangs können Kontrollen umgehen.

6. CLOUD-DLP



Cloud-DLP ergänzt Kontrollen für Daten in Cloud-Diensten und Cloud-Speichern.

- Prüft je nach Integration Dateien, Freigaben, Aktivitäten oder gespeicherte Inhalte.
- Zu breite Freigaben und nicht verwaltete Anwendungen können Daten exponieren.
- Sichtbarkeit und Durchestzung richten sich nach Dienst, Berechtigungen und Integration.

7. RICHTLINIEN-DURCHSETZUNG



Eine Richtlinie verknüpft Datenklasse, Kontext und gewünschte Reaktion.

- Mögliche Reaktionen: protokollieren, warnen, blockieren oder Ausnahme zur Prüfung weiterleiten.
- Schweregrad, Benutzerrolle, Ziel und Datenmenge können die Entscheidung beeinflussen.
- Zu strenge Regeln stören legitime Arbeit; zu schwache Regeln lassen Risiken bestehen.

8. VORFALL UND ESKALATION



Ein Treffer ist ein Hinweis zur Prüfung, nicht automatisch ein bestätigter Datenabfluss.

- Ereignis, betroffene Datenklasse, Aktion und Kontext nachvollziehbar dokumentieren.
- Nach festgelegtem Verfahren an zuständige Sicherheits- oder Datenschutzstellen eskalieren.
- Fehlalarme, Ausnahmen und tatsächliche Vorfälle nutzen, um Regeln gezielt anzupassen.

Einordnung



Diese Mindmap vermittelt ein Lernmodell. Sie ist keine Diagnose eines konkreten Vorfalls und keine Handlungsempfehlung für einen bestimmten Betrieb.

ZENTRUM · BACKUP-STRATEGIEN & RANSOMWARE-RESILIENZ

Backups helfen, Daten nach Verlust oder Verschlüsselung wiederherzustellen. Resilienz entsteht, wenn Sicherungen getrennt, geschützt und tatsächlich wiederherstellbar sind.

- Ziel: wichtige Daten und Dienste verlässlich zurückbringen.



1 · 3-2-1-1-0-REGEL



Mehrere Kopien auf unterschiedlichen Medien und mindestens eine getrennt oder unveränderbar halten.

- 3 Kopien insgesamt, einschließlich der Produktivdaten.
- 2 unterschiedliche Speichertypen.
- 1 Kopie außerhalb des Hauptstandorts.
- 1 Kopie offline oder unveränderbar.
- 0 ungeprüfte Fehler: Sicherungen regelmäßig auf Lesbarkeit und Wiederherstellung testen.

2 · IMMUTABLE STORAGE / WORM



Unveränderbarer Speicher verhindert für eine festgelegte Zeit Änderungen oder Löschungen an geschützten Daten. WORM bedeutet „Write Once, Read Many".

- Schützt Sicherungen vor bestimmten Manipulationen, auch durch kompromittierte Konten.
- Aufbewahrungsdauer und Löschregeln bewusst festlegen.
- Unveränderbarkeit erstetzt keine getrennte Kopie oder Wiederherstellungstests.

3 · AIR-GAPPING



Ein Air Gap trennt eine Sicherung vom erreichbaren Produktionsnetz; die Trennung kann physisch oder durch streng kontrollierte Verfahren erfolgen.

- Verringert direkte Erreichbarkeit durch Angreifer aus dem Produktionsnetz.
- Zugriff und Verbindung nur für notwendige Sicherungs- oder Wiederherstellungsabläufe erlauben.
- Auch getrennte Medien sicher aufbewahren und regelmäßig testen.

4 · RTO- UND RPO-ZIELE

RTO (Recovery Time Objective) bezeichnet die angestrebte maximale Wiederherstellungszeit. RPO (Recovery Point Objective) beschreibt den maximal tolerierten Datenverlust in Zeit.

- Beispiel: RTO 4 Stunden, RPO 1 Stunde.
- Ziele je Dienst und Datenart festlegen; sie beeinflussen Sicherungsfrequenz und Wiederanlaufplanung.
- Ein Backup allein garantiert nicht, dass ein RTO erreicht wird.



5 · OFFLINE-SNAPSHOTS



Snapshots halten einen Zustand von Daten oder Systemen zu einem bestimmten Zeitpunkt fest. Offline-Snapshots sind während ihrer Trennung nicht über das Produktionsnetz erreichbar.

- Zeitpunkte und Aufbewahrung passend zu Wiederherstellungszielen planen.
- Snapshot-Funktionen können von der Speicherplattform abhängen und sind nicht automatisch ein unabhängiges Backup.
- Wiederherstellung aus einem geeigneten Snapshot praktisch erproben.

6 · BACKUP-SERVER ZUGRIFFSBESCHRÄNKEN



Backup-Infrastruktur ist ein besonders wichtiges Schutzgut: Wer sie kontrolliert, könnte Sicherungen verändern oder löschen.

- Zugriff auf notwendige Personen und Systeme begrenzen.
- Getrennte Konten, starke Authentisierung und minimale Berechtigungen einsetzen.
- Administrationszugriffe protokollieren und regelmäßig überprüfen.

7 · RECOVERY-DRILLS

Wiederherstellungsübungen prüfen, ob Daten, Abläufe und Zuständigkeiten im Ernstfall funktionieren.

- Dateien, Systeme und kritische Dienste anhand realistischer Szenarien wiederherstellen.
- Ergebnis, Dauer, Datenstand und Probleme dokumentieren.
- Nach Fehlern Maßnahmen umsetzen und erneut testen.



8 · NOTFALLBETRIEBSPLAN



Ein Notfallbetriebsplan beschreibt, wie wichtige Arbeit weitergeht, wenn Systeme ausfallen oder wiederhergestellt werden.

- Prioritäten, Verantwortlichkeiten und Kontaktwege festhalten.
- Alternatie Abläufe und Abhängigkeiten kritischer Dienste benennen.
- Plan aktuell halten und mit Wiederherstellungs-übungen abstimmen.

WAF & REVERSE PROXY

Ein Reverse Proxy nimmt Anfragen vor einem Webserver entgegen. Ein WAF prüft HTTP-Anfragen und -Antworten auf verdächtige Muster. Beide können zusammenarbeiten, sind aber nicht dasselbe.

- WAF = Filter für Webverkehr auf Anwendungsebene
- Reverse Proxy = Vermittler zwischen Clients und Ursprungsserver
- Ziel: Risiken mindern, nicht jede Attacke sicher verhindern



1. ANFRAGEN IM BLICK

Der Proxy empfängt HTTP- oder HTTPS-Verkehr und leitet passende Anfragen an den Ursprung weiter.



- Kann mehrere Webserver hinter einer Adresse bündeln
- Kann Anfragen protokollieren und weiterleiten
- Fehlkonfigurationen können neue Risiken schaffen

2. SIGNATURBASIERTE ERKENNUNG

Signaturen beschreiben bekannte Angriffsmuster, etwa verdächtige Zeichenfolgen oder typische Anfrageformen.



- Erkennt bekannte Muster schnell
- Regeln und Signaturen müssen pepflegt werden
- Veränderte oder neue Angriffe können unentdeckt bleiben

3. ANOMALIEN & HEURISTIKEN

Heuristiken bewerten Merkmale einer Anfrage; Anomalieerkennung sucht Abweichungen von einem erwarteten Muster.



- Kann unbekannte Auffälligkeiten sichtbar machen
- Ungewöhnlich bedeutet nicht automatisch bösartig
- Schwellwerte beeinflussen Fehlalarme und Erkennungsquote

4. LAYER-7-DDoS & RATE LIMITING

Ein WAF kann HTTP-Anfragen begrenzen oder auffällige Muster blockieren. Das hilft gegen bestimmte Anwendungslast, ersetzt aber keine umfassende DDoS-Abwehr.



- Rate Limit: höchstens eine festgelegte Anfragerate
- Limits passend zu Endpunkt und legitimer Nutzung setzen
- Große volumetrische Angriffe brauchen oft vorgelagerte Netz- oder CDN-Abwehr

WICHTIG

Überlastung kann auch entstehen, bevor der WAF die Anfrage prüfen kann.

5. TLS-TERMINIERUNG

Bei TLS-Terminierung endet die verschlüsselte Verbindung am Proxy: dort kann Verkehr geprüft werden. Zum Ursprungsserver kann eine neue TLS-Verbindung aufgebaut werden.



- Zertifikat und Schlüssel am Terminierungspunkt schützen
- Proxy–Ursprung-Verbindung ebenfalls absichern
- Prüfen, wo Verschlüsselung endet und erneut beginnt

6. HEADER-VALIDIERUNG

HTTP-Header transportieren Zusatzinformationen zur Anfrage. Validierung kann fehlende, unerwartete oder widersprüchliche Werte erkennen.



- Weitergeleitete IP-Header nur von vertrauenswürdigen Proxies übernehmen
- Mehrdeutige oder fehlerhafte Header nach klarer Regel behandeln
- Sicherheitsheader der Anwendung nicht unbedacht überschreiben

7. COOKIE-SICHERHEIT

Cookies können Sitzungsinformationen enthalten. Proxy und Anwendung sollten sichere Cookie-Eigenschaften gemeinsam berücksichtigen.



- Secure: nur über HTTPS senden
- HttpOnly: kein Zugriff durch JavaScript
- SameSite begrenzt bestimmte websiteübergreifende Sendungen

ERGÄNZUNG

Cookie-Attribute ersetzen keine sichere Sitzungsverwaltung und keinen Schutz vor allen Angriffen.

8. CDN & URSPRUNGSSCHUTZ

Ein CDN kann Inhalte über verteilte Standorte ausliefern und zusätzliche Filter- oder Drosselfunktionen anbieten.



- WAF kann am CDN oder am Ursprungsnetz eingebunden sein
- Ursprung so konfigurieren, dass Umgehung des CDN erschwert wird
- Regeln, Protokolle und Ausnahmen regelmäßig überprüfen

EINORDNUNG

Ein WAF ist eine Schutzschicht. Sichere Anwendung, Updates und Überwachung bleiben ebenfalls wichtig.

INJECTION-SCHWACHSTELLEN

Injection entsteht, wenn nicht vertrauenswürdige Eingaben als Teil einer Anweisung interpretiert werden. Im Fokus: SQLi, CMDi, LDAPi und XPath-Injection.

- Ursache: Daten und ausführbare Anweisungen werden nicht sicher getrennt.
- Folgen reichen von Datenabfluss bis zu unbefugten Änderungen oder Befehlen.
- Schutz beginnt bei sicherer API-Nutzung und kontextgerechter Behandlung von Eingaben.



SICHERHEITSKONTEXT

- Diese Lernmindmap dient der Prävention und dem Verständens. Tests nur in ausdrücklich autorisierten Umgebungen durchführen.

1. WAS IST INJECTION?



Eine Anwendung baut aus Eingaben eine Abfrage oder Anweisung, Werden Eingaben dabei als Syntax statt nur als Daten behandelt, kann die beabsichtigte Bedeutung verändert werden.

- Betroffen sind etwa Datenbankabfragen, Betriebssystembefehle und Verzeichnisabfragen.
- Injection ist eine Schwachstellenklasse; konkrete Auswirkungen hängen von Anwendung, Rechten und Umgebung ab.
- Eingabevalidierung allein erstetzt keine sichere Trennung von Daten und Anweisung.

2. SQL INJECTION (SQLi)



SQLi betrifft Anwendungen, die SQL-Anweisungen unsicher zusammensetzen. Angreifende können unter Umständen Abfragen oder deren Ergebnisse beeinflussen.

- In-band: Angriff und Ergebnis laufen über denselben Kommunikationsweg; dazu zählt etwa sichtbare Ausgabe in der Anwendung.
- Blind: Ergebnisse werden indirekt aus Verhalten wie Antwortunterschieden oder Laufzeit erschlossen.
- Out-of-band: Datenübertragung erfolgt über einen separaten Kanal, sie setzt geeignete Datenbankfunktionen und erreichbare Kanäle voraus.
- Mögliche Folgen: unbefugtes Lesen, Ändern oder Löschen von Daten; Umfang hängt von Berechtigungen ab.

3. COMMAND INJECTION (CMDi)



CMDi tritt auf, wenn Eingaben unsicher in Betriebssystembefehle oder Shell-Ausdrücke gelangen. Shell-Metazeichen können je nach Shell und Kontext die Befehlsstruktur verändern.

- Besonders riskant: Befehle als Zeichenkette zusammensetzen und über eine Shell starten.
- Bevorzugt sichere Bibliotheksfunktionen statt Shell-Aufrufen verwenden.
- Wenn ein Prozess nötig: Argumente getrennt übergeben, Shell-Interpretation vermeiden und Rechte minimieren.
- Eine Positivliste zulässiger Werte kann ergänzen, ist aber kein Erstatz für sichere Prozessaufrufe.

4. LDAP- UND XPATH-INJECTION



LDAPi und XPath-Injection entstehen, wenn Eingaben unsicher in Filter oder Ausdrücke eingefügt werden. Die jeweilige Syntax unterscheidet sich, Schutz muss zum verwendeten Kontext passen.

- LDAPi kann Suchfilter oder Verzeichnisabfragen verändern.
- XPath-Injection kann XML-Abfragen beeinflussen.
- Sichere Bibliotheken und kontextspezifisches Escaping oder Parametrisierung nutzen.
- Keine SQL-Schutzmaßnahme ungeprüft auf LDAP oder XPath übertragen.

5. SCHADENSAUSMASS UND RISIKO



Das Risiko ergibt sich aus Ausnutzbarkeit und möglichen Folgen. Entscheidend sind unter anderem erreichbare Funktionen, Schutzbedarf der Daten und Rechte des betroffenen Dienstes.

- Vertraulichkeit: unbefugte Einsicht in Daten.
- Integrität: unerlaubte Änderungen oder Löschungen.
- Verfügbarkeit: Störung oder Ausfall von Funktionen und Diensten.
- Folgen begrenzen: minimale Datenbank- und Prozessrechte, Segmentierung und sichere Fehlerbehandlung.

6. PARAMETRISIERUNG UND PREPARED STATEMENTS

Bei SQL-Abfragen trennt Parametrisierung SQL-Struktur und Werte. Platzhalter werden über die Datenbankschnittstelle mit Werten belegt, statt Eingaben in den SQL-Text einzubauen.

- Bevorzugte Maßnahme gegen SQLi bei Werten in Abfragen.
- Platzhalter repräsentieren Werte, nicht beliebige SQL-Struktur wie Tabellennamen oder Sortierrichtung.
- Dynamische Struktur nur aus festgelegten, geprüften Optionen zusammensetzen.
- ORMs verwenden, ohne sich auf sie als automatische Sicherheitsgarantie zu verlassen.



7. VALIDIERUNG, ORM UND FEHLERBEHANDLUNG



Validierung prüft, ob eine Eingabe den erwarteten Regeln entspricht. Sie ergänzt sichere Abfrage- und Prozess-APIs, ersetzt sie aber nicht.

- Eingaben nach Zweck und Datentyp prüfen; möglichst enge Positivlisten verwenden.
- ORM-Abfragen sind nicht automatisch sicher: rohe SQL-Fragmente oder unsichere Stringverkettung bleiben riskant.
- Datenbankrechte nach Minimalprinzip vergeben.
- Fehler nach außen knapp halten: Details geschützt protokollieren und Protokolle gegen sensible Daten absichern.

8. TESTEN UND LERNPUNKTE



Sicherheitstests prüfen, ob Eingaben als Daten behandelt werden und Schutzmaßnahmen wirksam greifen. Ergebnisse sind im Kontext des Systems zu bewerten.

- Code-Review: Stringverkettung, rohe Abfragen und Shell-Aufrufe suchen.
- Unit- und Integrationstests: ungewöhnliche, leere und ungültige Werte abdecken.
- Autorisierte dynamische Tests: kontrollierte Testumgebung und klarer Umfang.
- Befunde dokumentieren, beheben und Regressionstests ergänzen.
- Merksatz: SQL parametrisieren, Shell-Aufrufe vermeiden, LDAP/XPath kontextgerecht behandeln und Rechte begrenzen.

XSS: WENN DATEN ZU CODE WERDEN

Cross-Site Scripting (XSS) ist eine Schwachstelle, durch die nicht vertrauenswürdige Inhalte im Browser als aktiver Code ausgeführt werden können. Betroffen ist der Kontext, in dem eine Webanwendung Daten darstellt oder verarbeitet.

- Angreifer bringen ausführbare Inhalte in eine Anwendung oder deren Browser-Ausgabe.
- Der Browser führt den Inhalt im Sicherheitskontext der betroffenen Website aus.
- Mögliche Folgen reichen von verfälschten Seiteninhalten bis zum Missbrauch einer angemeldeten Sitzung.

MERKSATZ

Nicht die Eingabe allein ist entscheidend, sondern wie sie später in einen Ausgabekontext gelangt.

1. REFLECTED XSS



Bei reflektiertem XSS verarbeitet die Anwendung eine Eingabe und gibt sie unmittelbar in einer Antwort zurück, ohne sie passend zum Ausgabekontext sicher zu behandeln. Häufig wird ein präparierter Link an ein Opfer weitergegeben.

- Die schädliche Eingabe kommt typischerweise mit einer Anfrage, etwa über einen URL-Parameter.
- Die Antwort enthält den Wert in einer Seite; der Browser interpretiert ihn dort als aktiven Inhalt.
- Ein erfolgreicher Angriff setzt meist voraus, dass das Opfer die präparierte Anfrage auslöst.
- Abwehr: Eingaben angemessen validieren und Ausgaben kontextgerecht kodieren.

2. STORED / PERSISTENT XSS



Bei gespeichertem XSS legt die Anwendung den schädlichen Inhalt dauerhaft ab, zum Beispiel in einem Kommentar oder Profilfeld. Spätere Seitenaufrufe können den Inhalt an andere Nutzer ausliefern.

- Der Inhalt wird gespeichert und nicht nur in einer einzigen Antwort reflektiert.
- Betroffene sehen ihn beim Laden der Seite, sofern er unsicher ausgegeben wird.
- Mögliche Speicherorte sind Datenbanken oder andere dauerhafte Inhaltsdienste.
- Abwehr: gespeicherte Daten als nicht vertrauenswürdig behandeln und bei jeder Ausgabe passend kodieren.

3. DOM-BASED XSS



DOM-basiertes XSS entsteht, wenn clientseitiger JavaScript-Code nicht vertrauenswürdige Daten unsicher in das Document Object Model (DOM) einfügt. Der entscheidende Fehler kann im Browsercode liegen, auch wenn der Server die Eingabe nicht in einer Antwort zurückspiegelt.

- Quelle: etwa URL-Fragment, Query-Parameter oder andere vom Nutzer beeinflussbare Daten.
- Unsichere Senke: eine DOM-Operation, die Daten als HTML oder ausführbaren Inhalt interpretiert.
- Sichere Alternative: Inhalte als Text einsetzen, statt sie als Markup interpretieren zu lassen.
- Clientseitige Quellen und Senken im Code gezielt überprüfen.

4. COOKIE-DIEBSTAHL UND SESSION HIJACKING



Kann eingeschleuster Code im Browser laufen, kann er möglicherweise auf für Skripte zugängliche Seitendaten zugreifen oder Aktionen im Namen des Nutzers auslösen. Ob ein Cookie lesbar ist, hängt unter anderem von seinen Schutzattributen ab.

- Das HttpOnly-Attribut verhindert den Zugriff auf das betreffende Cookie über clientseitige Skript-APIs.
- HttpOnly verhindert nicht, dass eingeschleuster Code andere Aktionen im angemeldeten Browser ausführt.
- Secure beschränkt die Cookie-Übertragung auf sichere HTTPS-Verbindungen.
- Session Hijacking bezeichnet die Übernahme oder missbräuchliche Nutzung einer Sitzung; XSS ist ein möglicher Angriffsweg, aber nicht der einzige.

WICHTIG

HttpOnly ist eine zusätzliche Schutzmaßnahme, kein Ersatz für die Behebung der XSS-Schwachstelle.

5. KONTEKT-SENSITIVES OUTPUT ENCODING



Ausgabekodierung stellt sicher, dass Daten in ihrem jeweiligen Kontext als Daten und nicht als ausführbare Syntax interpretiert werden. Eine Kodierung passt nicht für alle Stellen einer Webseite.

- HTML-Text, HTML-Attribute, JavaScript, CSS und URLs sind unterschiedliche Kontexte.
- Für den jeweiligen Kontext die passende Kodierung oder sichere API verwenden.
- Daten nicht durch selbst gebastelte Zeichen-Erstezungen absichern.
- Eingabevalidierung ist hilfreich, ersetzt aber keine sichere Ausgabebehandlung.

7. FRAMEWORK-SCHUTZMECHANISMEN



Viele moderne Webframeworks kodieren dynamische Werte bei der Darstellung standardmäßig als Text. Dieser Schutz kann verloren gehen, wenn Entwickler sichere Standardwege umgehen oder Roh-HTML zulassen.

- Standard-Templates und vorgesehene Textbiindungen bevorzugen.
- Funktionen für ungeprüftes oder rohes HTML nur mit sorgfältig bereinigten Inhalten verwenden.
- Sanitizing kann HTML bereinigen, muss aber zur erlaubten Inhaltspolitik passen.
- Framework-Schutz prüfen und nicht als Garantie gegen jeden XSS-Pfad verstehen.

8. KURZ ZUSAMMENGFASST



XSS beruht darauf, dass nicht vertrauenswürdige Daten im Browser als aktiver Inhalt interpretiert werden. Die passende Abwehr richtet sich nach dem Datenfluss und dem Ausgabekontext.

- Reflected XSS: Eingabe erscheint unmittelbar in einer Antwort.
- Stored XSS: unsicherer Inhalt wird gespeichert und später ausgeliefert.
- DOM-based XSS: clientseitiger Code setzt nicht vertrauenswürdige Daten unsicher ins DOM.
- Kontextgerecht kodieren, sichere Framework- und DOM-APIs nutzen und CSP sowie Cookie-Attribute ergänzend einsetzen.

032 BROKEN ACCESS CONTROL

1. ZUGRIFFSKONTROLLE VERSTEHEN

Zugriffskontrolle legt fest, welche angemeldete oder nicht angemeldete Person welche Ressource oder Aktion nutzen darf. Fehler entstehen, wenn eine Anwendung diese Regeln nicht zuverlässig durchsetzt.



- Authentifizierung prüft, wer jemand ist; Autorisierung prüft, was diese Person darf.
- Broken Access Control ist eine Kategorie der OWASP Top 10.
- Schutz gilt für Daten, Funktionen und einzelne Anfragen – nicht nur für sichtbare Menüs.

MERKSATZ
Eine ausgeblendete Schaltfläche ist keine Zugriffskontrolle. Die Berechtigung muss serverseitig geprüft werden.

2. IDOR UND OBJEKTZUGRIFF

Bei einer Insecure Direct Object Reference (IDOR) kann eine Anfrage ein Objekt über eine veränderbare Kennung ansprechen, ohne die Berechtigung für genau dieses Objekt ausreichend zu prüfen.



- Beispiel: Eine Rechnungs-ID wird in der URL geändert und zeigt fremde Daten.
- IDs können fortlaufend, zufällig oder anderweitig gestaltet sein; eine schwer erratbare ID ersetzt keine Autorisierung.
- Prüfe bei jedem Objektzugriff, ob die aktuelle Person dieses Objekt sehen oder ändern darf.

SICHERE PRÜFUNG
Berechtigung an der serverseitigen Ressource durchsetzen, nicht allein im Browser oder anhand der Kennung.

3. HORIZONTALE RECHTEAUSWEITUNG

Eine horizontale Rechteausweitung liegt vor, wenn eine Person auf Daten oder Funktionen einer anderen Person mit vergleichbarem Rollenstatus zugreift.



- Beispiel: Nutzer A liest durch manipulierte Objekt-ID das Profil von Nutzer B.
- Besonders wichtig bei Mandanten, Konten, Bestellungen und Dateien.
- Autorisierung muss Besitzer, Mandant oder passende Freigabe berücksichtigen.

4. VERTIKALE RECHTEAUSWEITUNG

Bei vertikaler Rechteausweitung erlangt eine Person Zugriff auf Funktionen oder Daten einer höheren Berechtigungsstufe.



- Beispiel: Ein normales Konto ruft eine nur für Administratoren gedachte Aktion direkt auf.
- Rollenprüfungen müssen serverseitig für jede privilegierte Aktion greifen.
- Rollenname allein genügt nicht, wenn zusätzlich Ressource oder Kontext entscheidend ist.

5. ROLLENPRÜFUNGEN UMGEHEN

Ein Zugriffskontrollfehler kann entstehen, wenn Prüfungen fehlen, uneinheitlich umgesetzt oder nur an einzelnen Anwendungswegen angewendet werden.



- Direkte HTTP-Anfragen können eine geschützte Oberfläche umgehen.
- Prüfe alternative Methoden, API-Routen und Zustandsübergänge.
- Vertraue nicht auf clientseitige Rollenangaben oder übermittelte Berechtigungswerte.

WICHTIG
Jede Anfrage ist separat zu autorisieren – auch wenn zuvor eine Seite oder ein Menü freigegeben wurde.

6. PATH TRAVERSAL

Path Traversal, auch Directory Traversal genannt, versucht durch manipulierte Pfadangaben aus dem vorgesehenen Verzeichnis auszubrechen und andere Dateien anzusprechen.



- Eingaben können etwa kodierte oder wiederholte Pfadsegmente enthalten.
- Einfache Zeichenfilter sind oft unzuverlässig, wenn Normalisierung und Kodierungen nicht berücksichtigt werden.
- Dateizugriff auf erlaubte Bereiche begrenzen und Pfade sicher auflösen.

ABWEHR
Nicht ungeprüfte Benutzereingaben als Dateipfad verwenden. Erlaubte Ressourcen möglichst über feste Kennungen zuordnen.

7. API-ENDPUNKTE UND ERZWUNGENES BROWSEN

Erzwungenes Browsen bezeichnet das direkte Aufrufen von Pfaden oder Endpunkten, die über die normale Navigation nicht erreichbar oder sichtbar sind. APIs sind ebenso zu schützen wie Webseiten.



- Nicht verlinkt bedeutet nicht geheim oder geschützt.
- Für jeden Endpunkt und jede HTTP-Methode Berechtigung und Ressourcenkontext prüfen.
- Fehlerantworten dürfen keine unnötigen sensiblen Details offenlegen.

Nicht verlinkte Endpunkte sind nicht automatisch geschützt. Jeder Endpunkt muss wie ein regulärer Zugriff geprüft werden.

8. LEAST PRIVILEGE UND AUDIT

Least Privilege bedeutet, Konten und Anwendungskomponenten nur mit den Rechten auszustatten, die sie für ihre Aufgabe benötigen. Regelmäßige Prüfungen helfen, Lücken und unnötige Rechte zu finden.



- Zugriff standardmäßig verweigern und benötigte Rechte gezielt vergeben.
- Rollen, Ressourcenbesitz und Mandantengrenzen systematisch testen.
- Autorisierungsereignisse protokollieren; Protokolle vor unberechtigtem Zugriff schützen.
- Tests mit erlaubten und verbotenen Zugriffen für jede relevante Rolle durchführen.

KURZCHECK
Kann ein Konto fremde Objekte lesen, fremde Objekte ändern oder eine höhere Rolle ausführen? Jede Antwort muss durch passende serverseitige Kontrollen verhindert werden.

033 CSRF & SSRF

CSRF: Browser/Sitzung.
SSRF: Server/Ziel.

1. CSRF-ABLAUF



Eine angemeldete Person besucht eine fremde oder manipulierte Seite. Diese löst eine Anfrage an die vertrauenswürdige Webanwendung aus; der Browser kann dabei die zugehörigen Cookies mitsenden.

- Kritisch sind zustandsändernde Aktionen wie Profiländerung oder Überweisung.
- Das Opfer muss angemeldet sein; der Browser muss passende Anmeldeinformationen mitsenden.
- Die fremde Seite kann die Antwort häufig nicht lesen – die unerwünschte Aktion kann dennoch ausgelöst werden.

2. SYNCHRONIZER-TOKEN



Bei sitzungsbasierter Anmeldung erzeugt der Server ein unvorhersagbares CSRF-Token und bindet es an die Sitzung. Die Anwendung erwartet es bei zustandsändernden Anfragen, etwa als Formularfeld oder Header.

- Fehlendes oder ungültiges Token: Aktion ablehnen.
- Token nicht in URLs oder Logs preisgeben; sicher vergelichen und erneuern, wenn die Sitzung wechselt.
- Ein Token ist kein Ersatz für Autorisierung oder Schutz vor XSS.

WICHTIG

Das Token muss für eine fremde Webiste unzugänglich sein; ein bloß vorhandenes, aber nicht geprüftes Token schützt nicht.

3. SAMESITE-COOKIES



Das Cookie-Attribut SameSite begrenzt, wann Browser Cookies bei websiteübergreifenden Anfragen mitsenden. Es ist eine hilfreiche zusätzliche CSRF-Abwehr, aber keine universelle Einzellösung.

- Strict: Cookies werden bei websiteübergreifenden Kontexten weitgehend zurückgehalten.
- Lax: Cookies werden in bestimmten Fällen, typischerweise bei sicheren Top-Level-Navigationen, mitgesendet.
- None: websiteübergreifendes Senden ist erlaubt; Secure ist erforderlich.
- Für sensible Aktionen weiterish Token oder geeignete gleichwertige Prüfung verwenden.

4. SSRF-ANGRIFFSFLÄCHE



Eine Anwendung nimmt ein Ziel entgegen und ruft es serverseitig ab – etwa für Webhooks, Vorschauen, Bildimport oder URL-Abruf. Unzureichend kontrollierte Ziele können interne Dienste erreichbar machen.

- Angreifer können interne Hostnamen oder private IP-Bereiche anvisieren.
- Cloud-Metadaten-Endpunkte sind ein mögliches Ziel; Zugriff kann vertrauliche Instanzdaten oder Zugangsdaten offenlegen.
- Auch DNS-Auflösung, Weiterleitungen und alternative IP-Schreibweisen erschweren Prüfungen.

GRENZE DER AUSSAGE

Eine SSRF-Möglichkeit bedeutet nicht automatisch, dass ein bestimmter interner Dienst erreichbar ist; das hängt von Netzwerkregeln und Anwendung ab.

5. ZIELVALIDIERUNG UND ALLOWLIST



Bevor der Server eine Verbindung aufbaut, muss die Anwendung das Ziel strikt begrenzen. Für bekannte Integrationen ist eine Allowlist erlaubter Hosts meist sicherer als der Versuch, alle gefährlichen Ziele auszuschließen.

- Schema erlauben, zum Besipiel nur https; unerwartete Schemata ablehnen.
- Host exakt gegen eine gepflegte Allowlist prüfen; Benutzereingaben nicht als volltsändige URL ungeprüft übernehmen.
- DNS-Auflösung und tatsächliche Ziel-IP prüfen; Schleißen über private, Loopback-, Link-Local- und interne Adressen verhindern.
- Weiterleitungen deaktivieren oder jedes Weiterleitungsziel erneut volltsändig prüfen.

NICHT AUSREICHEND

Nur eine Zeichenkettenprüfung oder ein einmaliger DNS-Check kann Umgehungen und DNS-Rebinding übersehen.

6. NETZWERKBEGRENZUNG



Anwendungscode und Netzarchitektur sollten gemeisnam verhindern, dass ein abrufender Dienst unnötige interne Ressourcen erreicht.

- Ausgehenden Verkehr auf erforderliche Ziele und Ports beschränken.
- Abrufdienste in einer isolierten Zone oder DMZ → betreiben; interne Vermaltungsnetze nicht routbar machen.
- Cloud-Metadatenzugriff durch Plattformkontrollen begrenzen und minimale Berechtigungen vergeben.
- Zeitlimits, Antwortgrößenlimits und Protokoll-beschränkungen setzen; unnötige Protokolle sperren.

7. CSRF ODER SSRF?



Zur Eiendrnung zuerst fragen: Wer sendet die gefälschte oder beeinflusste Anfrage, und wessen Netzwerkposition wird ausgenuntzt?

- Browser samt Sitzungcookie betroffen → CSRF prüfen.
- Server ruft vom Angreifer beeinflusstes Ziel ab → SSRF prüfen.
- Beide Risiken können unabhängig voneinander in derselben Anwendung vorkommen.
- Maßnahmen passend zur Ursache kombinieren: CSRF-Token und Cookie-Regeln einerseits, Zielvalidierung und Egress-Kontrollen andererseits.

Welche Komponente führt die unerwünschte Anfrage aus?

Der angemeldete Browser sendet sie mit Sitzungskontext.	Der Server verbindet sich mit einem beeinflussten Ziel.
↓	↓
CSRF-Schutz für zustands-ändernde Aktionen prüfen: Token, SameSite und sichere Anfragegestaltung.	SSRF-Schutz prüfen: Allowlist, Schema- und IP-Prüfung, Weiterleitungsbenadlung sowie Netzwerk-Egress-Regeln.

8. KURZÜBERBLICK



CSRF nutzt das Vertrauen der Anwendung in den Browser; SSRF nutzt die Netzwerkreichweite der Anwendung. Verteidigung beginnt mit der richtigen Einordrung und wird durch mehrere unabhängige Kontrollen gestärkt.

- CSRF: sitzungsgebundene Tokens; SameSite als zusätzliche Schutzschicht.
- SSRF: strikte Schema- und Host-Allowlist; IP-Prüfung auch nach DNS-Auflösung und Weiterleitungen.
- Beide: minimale Berechtigungen, sichere Voreinstellungen und gezielte Protokollierung.

MERKSATZ

CSRF zielt auf die Sitzung einer Person; SSRF zielt auf den Netzwerkzugriff des Servers.

Sichere Authentifizierung und Session-Management im Web



Eine Anmeldung weist eine Identität nach; eine Session hält den angemeldeten Zustand über mehrere HTTP-Anfragen hinweg. Diese Mindmap zeigt, wie Webanwendungen Sitzungen sicher erzeugen, begrenzen und beenden.

- Session-ID: zufälliger, nicht erratbarer Verweis auf serverseitigen Sitzungszustand
- Cookie: üblicher Transport der Session-ID zwischen Browser und Website
- Ziel: Sitzungsdiebstahl, Fixierung und missbräuchliche Anmeldeversuche erschweren



BEGRIFFE UNTERSCHEIDEN

Authentifizierung prüft, wer angemeldet ist. Autorisierung entscheidet, worauf diese Identität zugreifen darf. Eine gültige Session ersetzt keine Autorisierungsprüfung.

1. ABLAUF EINER WEB-SESSION



HTTP-Anfragen sind grundsättlich voneinander unabhängig. Eine Session verknüpft sie über einen Sitzungsbezeichner, den der Browser bei späteren Anfragen mitsendet.

- Anmeldung: Zugangsdaten oder ein anderer Faktor werden geprüft.
- Erfolg: Der Server erzeugt eine neue Session und gibt eine ID aus.
- Folgeanfrage: Der Browser sendet die ID üblicherweise im Cookie zurück.
- **Server: Prüft die ID und wendet für jede Aktion erneut Berechtigungen an.**

2. SICHERE SESSION-ID



Die Session-ID ist ein Geheimnis auf Zeit. Wer sie besitzt, könnte die zugehörige Sitzung übernehmen; sie darf deshalb nicht erratbar oder unnötig offengelegt sein.

- Mit einem kryptografisch sicheren Zufallszahlengenerator erzeugen.
- Ausreichend hohe Entropie verwenden; keine fortlaufenden oder vorhersgabaren IDs.
- ID nicht aus Benutzername, Zeitstempel oder anderen bekannten Daten ableiten.
- In URLs, Fehlermeldungen und unnötigen Logs vermeiden.
- Serverseitig prüfen und bei unbekannten oder abgelaufenen IDs ablehnen.

3. LEBENSZYKLUS UND TIMEOUTS



Sitzungen sollten nur so lange und so weit bestehen, wie es der Nutzung entspricht. Der Server muss Ablauf und Beendigung durchsetzen.

- Inaktive Sitzungen nach einer angemessenen Leerlaufzeit verfallen lassen.
- Zusatzilch eine absolute Höchstdauer festlegen, auch bei fortlaufender Aktivität.
- Nach erfolgreicher Anmeldung und Privilegienwechsel eine neue ID ausstellen.
- Nach Ablauf die alte ID serverseitig ungültig machen.
- Bei sensiblen Aktionen gegebenenfalls eine anneute Authentifizierung verlangen.



ABLAUF IST SERVERSEITIG

Ein abgelaufenes Cookie allein beendet eine Sitzung nicht zuverlässig. Die zugehörige Session muss auch serverseitig ablaufen oder widerrufen werden.

4. SCHUTZ VOR SESSION FIXATION



Bei einer Session-Fixierung versucht ein Angreifer, dem Opfer eine bekannte Session-ID unterzuschieben und diese nach dessen Anmeldung weiterzuverwenden.

- Nach erfolgreicher Anmeldung die Session-ID wechseln.
- Auch nach einem Wechsel zu höheren Rechten eine neue ID ausstellen.
- Den alten Bezeichner serverseitig ungültig machen.
- Keine vom Client frei gewählten IDs als angemeldete Sitzung übernehmen.

5. COOKIE-SICHERHEITSFLAGS



Cookie-Attribute begrenzen, wann und wie der Browser einen Sitzungsbezeichner sendet. Sie ergänzen serverseitige Schutzmaßnahmen, ersetzen sie aber nicht.

- Secure: Cookie nur über HTTPS senden.
- HttpOnly: Zugriff auf das Cookie über JavaScript verhindern; es schützt nicht vor allen Folgen von XSS.
- SameSite: Cross-Site-Senden einschränken und dadurch CSRF-Risiken mindern; passande Einstellung abhängig vom Ablauf wählen.
- Sitzungs-Cookie ohne unnötig lange persistente Laufzeit verwenden.
- HTTPS für die gesamte Sitzung einsetzen und Cookies nicht über ungeschützte Verbindungen übertragen.



SAMESITE IST KEIN ALLHEILMITTEL

SameSite kann CSRF erschweren, deckt aber nicht jeden Anwendungsfall ab. Kritische Aktionen brauchen ein zum System passendes CSRF-Schutzkonzept.

6. JWT SICHER EINSETZEN



Ein JSON Web Token (JWT) kann signierte Ansprüche enthalten. Eine Signatur schützt die Integrität, verschlüsselt den Tokeninhalt aber nicht automatisch.

- Signatur mit einem geeigneten Verfahren prüfen, Algorithmus nicht unkontrolliert aus dem Token übernehmen.
- Aussseller, Empfänger und Ablaufzeit passend zur Anwendung validieren.
- Keine Geheimnisse oder unnötig sensiblen Daten in lesbare Ansprüche schreiben.
- Token kurzlebig halten und sicher speichern; Browser-Speicherung hat unterschiedlide XSS- und CSRF-Risiken.
- Widerruf und Rotation berücksichtigen: Ein gültig signierter Token bleibt sonst bis zum Ablauf verwendbar.

7. BRUTE-FORCE-ABWEHR



Viele automatisierte Anmeldeversuche können Passwörter erraten oder Konten überlasten. Schutz sollte wirksam sein, ohne legitime Nutzer unnötig auszusperren.

- Versuche nach Konto und weiteren Signalen begrenzen; IP allein ist kein zuverlässiger Maßstab.
- Verzögerungen oder abgestufte Begrenzung einsetzen und verteilte Versuche berücksichtigen.
- Mehrfaktor-Authentifizierung anbieten, besonders für wichtige Konten.
- Keine leicht erratbaren Standardpasswörter; kompromittierte oder schwache Passwörter blockieren.
- Fehlermeldungen so gestalten, dass sie nicht verraten, ob ein Konto existiert.

8. WIDERRUF, LOGOUT UND KURZÜBERBLICK



Logout ist erst vollständig, wenn die Sitzung nicht mehr verwendbar ist. Das Löschen des Browser-Cookies allein widerruft keine serverseitig gültige Session.

- Beim Logout die Session serverseitig invalidieren und das Cookie im Browser löschen.
- Bei Passwortwechsel, Kontosperrung oder Sicherheitsvorfall betroffene Sessions widerrufen.
- Bei JWTs Widerrufsstrategie vorsehen, etwa kurze Laufzeit und kontrollierte Erneuerung.
- Merksatz: **starke ID, sichere Übertragung, Erneuerung bei Privilegienwechsel, begrenzte Laufzeit, wirksamer Widerruf.**
- Pro Anfrage sowohl Session-Gültigkeit als auch erforderliche Berechtigung prüfen.

035 API-SICHERHEIT

REST: Identität, Daten, Verfügbarkeit

1. BEDROHUNGSMODELL UND ANGRIFFSFLÄCHE



Erfasse, welche Akteure, Daten und Komponenten geschützt werden müssen. Eine API kann über Endpunkte, Identitäten, Abhängigkeiten und Fehlkonfigurationen angegriffen werden.

- Inventar: Hosts, Versionen, Routen, Methoden und Datenarten dokumentieren.
- Bedrohungen: unbefugter Objektzugriff, gestohlene Tokens, Injection, Missbrauch und Überlastung betrachten.
- OWASP-Risikofelder einbeziehen: Objekt- und Funktionsautorisierung, Authentifizierung, Ressourcenverbrauch, Fehlkonfiguration und unsichere Drittanbieter-APIs.
- Schutzbedarf und Auswirkungen priorisieren; Angriffsfläche regelmäßig aktualisieren.

2. API-SCHLÜSSEL SICHER VERWALTEN



API-Schlüssel identifizieren häufig einen Client oder eine Anwendung. Ein Schlüssel allein beweist nicht zuverlässig die Identität eines Menschen und ersetzt keine feingranulare Autorisierung.

- Nur über TLS übertragen; niemals in URLs, Quellcode-Repositories oder Logs offenlegen.
- Schlüssel zufällig und ausreichend lang erzeugen, sicher speichern und pro Anwendung oder Umgebung trennen.
- Berechtigungen eng begrenzen; Ablauf, Rotation und sofortigen Widerruf unterstützen.
- Bei vertraulichen Benutzeraktionen zusätzlich passende Authentifizierung und Autorisierung verlangen.

MERKE
Ein API-Key ist typischerweise ein Bearer-Geheimnis: Wer ihn besitzt, kann ihn möglicherweise verwenden. Behandle ihn entsprechend vertraulich.

3. OAUTH 2.0 UND SCOPES



OAuth 2.0 ermöglicht delegierte Autorisierung. Scopes grenzen ein, welche Arten von Zugriff ein Token erlaubt; sie ersetzen nicht die Prüfung, ob der Zugriff auf ein konkretes Objekt zulässig ist.

- Nur benötigte Scopes anfordern und erteilen (Least Privilege).
- Signatur, beziehungsweise Token-Integrität, Aussteller, Zielgruppe und Ablauf serverseitig passend zum Tokenformat prüfen.
- Scopes und Rollen mit Objekt- und Funktionsautorisierung kombinieren.
- Tokens nicht in URLs oder ungeschützten Speicherorten ablegen; Widerruf und kurze Gültigkeit passend zum Risiko planen.

ABGRENZUNG
OAuth 2.0 ist ein Autorisierungsrahmen. Für Anmeldeinformationen und Identitätsaussagen wird häufig OpenID Connect ergänzt.

4. AUTORISIERUNG PRO OBJEKT UND FUNKTION



Eine gültige Anmeldung bedeutet nicht, dass jede Aktion erlaubt ist. Die API muss Berechtigungen bei jeder Anfrage und für das jeweils adressierte Objekt prüfen.

- Objektzugriff: prüfen, ob der aktuelle Nutzer genau dieses Objekt lesen oder ändern darf.
- Funktionszugriff: privilegierte Aktionen und administrative Routen zusätzlich absichern.
- IDs nicht als Geheimnis behandeln; schwer erratbare Kennungen verhindern keine fehlende Autorisierung.
- Tests mit mehreren Konten und Rollen decken horizontale und vertikale Rechtefehler auf.

5. RATE LIMITING UND THROTTLING



Begrenze Anfragen und teure Operationen, damit Missbrauch und Lastspitzen die Verfügbarkeit nicht unverhältnismäßig beeinträchtigen. Limits sollten zur Funktion und zum Nutzungskontext passen.

- Budgets nach Identität, Route, IP oder Kombination festlegen; IP allein ist kein zuverlässiger Nutzerbezug.
- Burst-Limits und Zeitfenster erwägen; ressourcenintensive Operationen separat begrenzen.
- Bei Überschreitung konsistente Fehler liefern, etwa HTTP 429, und gegebenenfalls Retry-After nutzen.
- Limits überwachen und so abstimmen, dass legitime Nutzung möglich bleibt.

BETRIEB
Rate Limits sind eine Schutzschicht, kein Ersatz für Kapazitätsplanung, Authentifizierung oder Schutz gegen verteilte Angriffe.

6. EINGABE- UND SCHEMA-VALIDIERUNG



Behandle alle Eingaben als nicht vertrauenswürdig: Pfad- und Query-Parameter, Header sowie Request-Bodies. Validierung macht erwartete Datenformen explizit.

- Für jede Route erlaubte Felder, Datentypen, Formate, Grenzen und Pflichtangaben definieren.
- Unbekannte oder unzulässige Felder konsequent behandeln; Eingaben nicht bloß durch Zeichenfilter „bereinigen".
- Werte kontextgerecht weiterverwenden und sichere Datenbank- beziehungsweise System-APIs einsetzen.
- Fehlerantworten knapp halten und keine internen Details preisgeben.

7. MASS ASSIGNMENT UND ÜBERMÄSSIGE DATENAUSGABE



Mass Assignment entsteht, wenn ungeprüfte Request-Felder direkt interne Objekteigenschaften verändern. Übermäßige Datenausgabe liegt vor, wenn Antworten mehr sensible Informationen enthalten als der Client benötigt.

- Eingabefelder explizit erlauben (Allowlist) und sensible Attribute wie Rolle oder Kontostatus separat kontrollieren.
- Für Requests und Responses eigene Datenmodelle oder DTOs verwenden.
- Antworten auf notwendige Felder beschränken; sensible Eigenschaften nicht erst im Client ausblenden.
- [illegible]
- Objekt- und Feldberechtigungen serverseitig testen.

BEISPIEL
Ein Profil-Update sollte nicht ermöglichen, durch ein zusätzliches JSON-Feld eigenmächtig „role: admin" zu setzen.

8. API-GATEWAY, TLS UND SICHERE BETRIEBSFÜHRUNG



Ein API-Gateway kann gemeinsame Kontrollen zentral bereitstellen. Es entbindet Backend-Dienste nicht davon, Identität, Objektzugriff und Eingaben selbst angemessen zu prüfen.

- TLS für die gesamte Übertragung erzwingen; Zertifikate korrekt prüfen und HTTP auf HTTPS umleiten oder ablehnen.
- Gateway für Routing, Limits und zentrale Richtlinien einsetzen; Schutzwirkung und Umgehungspfade testen.
- Fehlermeldungen, CORS, Versionen und unnötige HTTP-Methoden bewusst konfigurieren.
- Protokollieren, ohne Tokens, API-Schlüssel oder unnötige personenbezogene Daten zu speichern; Alarme und regelmäßige Tests vorsehen.

ZUSAMMENSPIEL
Gateway-Kontrollen ergänzen die Autorisierung in der Anwendung. Ein interner Dienst darf nicht automatisch als vertrauenswürdig gelten.

036

Sicherheitsrelevante HTTP-Response-Header

1. CONTENT-SECURITY-POLICY (CSP)

CSP legt fest, aus welchen Quellen eine Seite bestimmte Inhalte laden oder ausführen darf.



- Wichtige Direktiven: default-src, script-src, style-src, img-src, connect-src.
- Nonce oder Hash können erlaubte Inline-Skripte gezielt freigeben.
- object-src 'none' und base-uri 'self' sind häufig sinnvolle Einschränkungen.
- report-only ermöglicht Beobachtung, ohne Verstöße zunächst zu blockieren.
- Zu breite Quellen oder unsichere Ausnahmen schwächen die Richtlinie.

2. STRICT-TRANSPORT-SECURITY (HSTS)

HSTS weist Browser an, eine Website für eine festgelegte Zeit ausschließlich über HTTPS aufzurufen.



- Header: Strict-Transport-Security: max-age=31536000
- Nur über eine sichere HTTPS-Antwort übernehmen Browser die Richtlinie.
- includeSubDomains betrifft auch Subdomains; vorher deren HTTPS-Betrieb prüfen.
- preload ist eine zusätzliche, schwer rückgängig zu machende Aufnahme in Browserlisten.
- HSTS schützt nicht den ersten Kontakt, wenn die Domain nicht bereits bekannt ist.

3. X-FRAME-OPTIONS

Dieser ältere Header steuert, ob eine Seite in einem Frame eingebettet werden darf, und kann Clickjacking erschweren.



- DENY verbietet Einbettung.
- SAMEORIGIN erlaubt Einbettung durch Seiten desselben Ursprungs.
- Für moderne Richtlinien ist CSP frame-ancestors die flexiblere Wahl.
- X-Frame-Options: ALLOW-FROM ist veraltet und wird nicht zuverlässig unterstützt.
- Widersprüche zwischen Richtlinien vermeiden und Verhalten im Zielbrowser testen.

4. X-CONTENT-TYPE-OPTIONS

Der Wert nosniff weist Browser an, den angegebenen MIME-Typ nicht durch Inhaltt-Sniffing zu ersetzen.



- Typischer Header: X-Content-Type-Options: nosniff
- Daten mit passendem Content-Type ausliefern.
- Für Skripte und Stylesheets besonders auf korrekte MIME-Typen achten.
- Der Header korrigiert keine falsch klassifizierten oder unsicheren Inhalte.

5. REFERRER-POLICY

Die Richtlinie begrenzt, welche Referrer-Informationen der Browser bei Navigationen und Anfragen weitergibt.



- Ein verbreiteter Standardwert ist strict-origin-when-cross-origin.
- Bei gleicher Herkunft kann mehr Referrer-Information übertragen werden als zwischen Ursprüngen.
- HTTPS zu HTTP kann die Referrer-Übermittlung unterdrücken.
- no-referrer minimiert Weitergabe, kann aber Funktionen beeinträchtigen.
- Referrer-Policy schützt nicht vor allen Formen der Datenweitergabe.

6. PERMISSIONS-POLICY

Permissions-Policy beschränkt die Nutzung bestimmter Browserfunktionen durch die eigene Seite und eingebettete Inhalte.



- Beispiele für Funktionen: Kamera, Mikrofon, Standort.
- Syntax und verfügbare Direktiven hängen von der Browserunterstützung ab.
- Zugriff nur dort erlauben, wo die Anwendung ihn benötigt.
- Ein Header wie Permissions-Policy: camera=(), microphone=() sperrt diese Funktionen für die Seite.
- Einträge mit eingebetteten Frames und deren Delegation gezielt prüfen.

7. CROSS-ORIGIN RESOURCE POLICY (CORP)

CORP steuert, ob andere Ursprünge eine Antwort in bestimmten Cross-Origin-Kontexten einbinden dürfen.



- Werte: same-origin, same-site oder cross-origin.
- same-origin beschränkt die Einbindung auf denselben Ursprung.
- same-site erlaubt passende Ressourcen derselben Website, auch über unterschiedliche Ursprünge.
- Falsche Einschränkungen können legitime Einbindungen blockieren.
- CORP ist von CORS zu unterscheiden: CORS regelt unter anderen, ob Skripte Cross-Origin-Antworten lesen dürfen.

8. IMPLEMENTIERUNGSPRÜFUNG

Prüfe Header an realen Antworten und im Browserkontext, statt nur eine Konfigurationsdatei anzusehen.



- HTTPS, Weiterleitungen, Fehlerseiten und unterschiedliche Anwendungspfade berücksichtigen.
- Doppelte oder widersprüchliche Header sowie unerwartete Proxies prüfen.
- CSP zunächst beobachten und Berichte auf legitime sowie verdächtige Verstöße auswerten.
- Änderungen in unterstützten Browsern und mit benötigten Funktionen testen.
- Dokumentieren, welche Richtlinie für welche Antwort gilt.

MERKSATZ

Ein gesetzter Header ist noch kein Nachweis, dass die gewünschte Schutzwirkung erreicht wird.

037 – SECURE SOFTWARE DEVELOPMENT LIFECYCLE

Sicherheit über den gesamten Softwarelebenszyklus.

SSDLC: SICHERHEIT VON ANFANG AN



Der Secure Software Development Lifecycle (SSDLC) integriert Sicherheitsarbeit in Planung, Entwurf, Entwicklung, Prüfung, Veröffentlichung und Betrieb einer Software. Ziel ist, Risiken früh zu erkennen und angemessene Schutzmaßnahmen über den gesamten Lebenszyklus hinweg umzusetzen.

- Sicherheit ist eine fortlaufende Aufgabe, kein einzelner abschließender Test.
- Maßnahmen richten sich nach Schutzbedarf, Bedrohungen und Einsatzkontext.

MERKE
Ein SSDLC ist ein Prozessrahmen: Teams passen Aktivitäten an Produkt, Risiken und Entwicklungsweise an.

1. SICHERHEITSANFFORDERUNGEN



Sicherheitsanforderungen beschreiben überprüfbar, was die Software schützen und leisten soll. Sie nennen:

- **Schutzgüter und Daten klassifizieren;** Vertraulichkeit, Integrität und Verfügbarkeit berücksichtigen.
- Zugriff, Authentisierung, Protokollierung, Datenschutz und Wiederherstellung konkret festlegen.
- Anforderungen an Bedrohungen, Nutzerrollen und geltende Vorgaben knüpfen.
- Jede Anforderung mit einem passenden Prüfverfahren verknüpfen.

2. BEDROHUNGSMODELLIERUNG: STRIDE



Threat Modeling macht mögliche Angriffe und Schutzbedarfe sichtbar, bevor Implementierungsdetails feststehen. STRIDE ordnet Bedrohungen **in sechs Kategorien.**

- Spoofing: Identität vortäuschen.
- Tampering: Daten oder Code unbefugt verändern.
- Repudiation: Handlungen bestreiten können, wenn belastbare Nachweise fehlen.
- Information Disclosure: Informationen offenlegen.
- Denial of Service: Verfügbarkeit beeinträchtigen.
- Elevation of Privilege: Rechte unbefugt ausweiten.

Systemgrenzen, Datenflüsse, Akteure und Vertrauensannahmen **dokumentieren; Gegenmaßnahmen ablieten.**

EINORDNUNG
STRIDE ist ein Denk- und Klassifikationsmodell, keine vollständige Liste aller Risiken und kein Nachweis, dass ein System sicher ist.

STRIDE ist ein Modell, keine vollständige Risikoliste.

3. SICHERE CODIERLEITLINIEN



Secure Coding reduziert häufige Fehler durch nachvollziehbare Regeln, sichere Standardwerte und geeignete Bibliotheken.

- Eingaben kontextgerecht validieren; Ausgaben sicher kodieren.
- Parametrisierte Datenbankabfragen statt unsicherer String-Verkettung verwenden.
- Fehler sicher behandeln; keine Geheimnisse oder sensiblen Daten in Logs ausgeben.
- Least Privilege und sichere Authentisierungs- und Autorisierungsprüfungen umsetzen.
- Abhängigkeiten und Geheimnisse kontrolliert verwalten; Codeänderungen prüfen.

4. SAST: STATISCHE ANALYSE



Static Application Security Testing (SAST) untersucht Quellcode oder andere Softwareartefakte, ohne das Programm wie ein laufendes System auszuführen.

- Früh im Entwicklungsprozess und wiederholt einsetzen.
- Befunde nach Risiko und Kontext prüfen; falsch-positive Meldungen sind möglich.
- Ergebnisse nachvollziehen, priorisieren und beheben; Scanner erstezt keine Codeprüfung.

5. DAST: DYNAMISCHE ANALYSE



Dynamic Application Security Testing (DAST) prüft eine laufende Anwendung über ihre beobachtbaren Schnittstellen.

- Testumgebung und Berechtigungen festlegen; nur autorisierte Systeme prüfen.
- Anwendung aus Sicht externer Anfragen untersuchen, etwa Eingaben und Antworten.
- DAST ergänzt SAST, deckt aber nicht jede Code- oder Laufzeitbedrohung ab.

SICHER TESTEN
Dynamische Tests können Daten verändern oder Dienste belasten. Verwende dafür eine kontrollierte, freigegebene Umgebung.

6. SCA: KOMPONENTEN UND ABHÄNGIGKEITEN



Software Composition Analysis (SCA) erfasst und bewertet verwendete Drittanbieter-Komponenten und Pakete.

- Abhängigkeiten und Versionen inventarisieren; eine Software-Stückliste kann dabei helfen.
- Bekannte Schwachstellen, Lizenzbedingungen und Wartungsstatus bewerten.
- Updates kontrolliert testen und einspielen; transitive Abhängigkeiten mitberücksichtigen.
- Ein Treffer bedeutet nicht automatisch Ausnutzbarkeit: Nutzung und Kontext bewerten.

7. PENETRATIONSTEST VOR VERÖFFENTLICHUNG



Ein **Penetrationstest** untersucht innerhalb eines vereinbarten Umfangs, ob sich Sicherheitslücken praktisch ausnutzen lassen. Er ergänzt automatisierte Prüfungen und laufende **Qualitätssicherung.**

- Umfang, Ziele, Zeitfenster und erlaubte Methoden vorab festlegen.
- Befunde mit Auswirkungen und nachvollziehbaren Reproduktionshinweisen dokumentieren.
- Risiken beheben oder bewusst behandeln; kritische Änderungen erneut prüfen.
- Ein bestandener Test ist eine Momentaufnahme, keine Garantie für Sicherheit.

GRENZE
Tests nur mit ausdrücklicher Autorisierung und innerhalb des vereinbarten Umfangs durchführen.

8. INCIDENT-RESPONSE-PLANUNG



Incident Response bereitet das Team darauf vor, Sicherheitsvorfälle zu erkennen, einzudämmen, zu beheben und daraus zu lernen.

- Rollen, Eskalationswege und Kontaktmöglichkeiten festlegen.
- Erkennung, sichere Beweissicherung und Kommunikationsabläufe planen.
- Eindämmung, Wiederherstellung und Nachbereitung berücksichtigen.
- Plan regelmäßig anhand von Übungen und Erkenntnissen verbessern.

038 DEVSECOPS & CI/CD-PIPELINE-SICHERHEIT

Sicherheit als wiederholbarer Teil des Entwicklungs- und Bereitstellungsablaufs: Risiken früh erkennen, Zugriffe begrenzen und Ergebnisse nachvollziehbar prüfen.

1. ZIEL UND ABLAUF



Eine CI/CD-Pipeline baut, testet und stellt Software automatisiert bereit. DevSecOps integriert Sicherheitskontrollen in diese Schritte, statt sie nur am Ende anzusetzen.

- Codeänderung → Build → Tests → Paketierung → Freigabe → Bereitstellung
- Prüfungen möglichst früh ausführen und Ergebnisse für Verantwortliche sichtbar machen
- Ein fehlgeschlagener Check kann den Ablauf stoppen oder eine begründete Ausnahme erfordern

2. PRE-COMMIT-HOOKS



Hooks prüfen Änderungen lokal vor dem Commit. Ein Secret-Scanner sucht beispielewiess nach möglichen API-Schlüsseln oder Zugangsdaten, bevor sie ins Versionsarchiv gelangen.

- Erkannte Treffer prüfen; Fehlalarme sind möglich
- Echte offengelegte Zugangsdaten widerrufen oder rotieren, nicht nur aus der aktuellen Datei löschen
- Hooks ergänzen serverseitige Prüfungen: lokale Kontrollen lassen sich umgehen

SICHERER UMGANG
Secrets gehören in geeignete Secret-Verwaltung oder geschützte Umgebungsvariablen, nicht in Quellcode oder Logs.

3. PIPELINE-RUNNER HÄRTEN



Runner führen Build- und Testaufträge aus und können Zugriff auf Quellcode, Artefakte oder Geheimnisse erhalten. Behandle sie als sicherheitsrelevante Systeme.

- Bevorzuge kurzlebige, isolierte Runner und begrenze Netzwerkzugriffe
- Trenne nicht vertrauenswürdige Pull-Requests von Jobs mit produktiven Geheimnissen
- Halte Betriebssystem, Runner-Agent und Build-Werkzeuge aktuell
- Vermeide privilegierte Container und unnötige Mounts oder Berechtigungen

4. INFRASTRUCTURE AS CODE SCANNEN



IaC beschreibt Infrastruktur beipielswiese in Terraform- oder Kubernetes-Konfigurationen. Scanner markieren mögliche Fehlkonfigurationen, bevor Änderungen ausgerollt werden.

- Beispiele: öffenllich erreichbare Speicher, zu offene Firewallregeln, fehlende Verschlüsselung
- Befunde anhand Kontext und Schwere priorisieren; Scanner können Fehlalarme erzeugen
- Policies und sichere Vorlagen helfen, wiederkehrende Fehlkonfigurationen zu verhindern

5. CONTAINER-IMAGES PRÜFEN



Image-Scanning sucht unter anderem nach bekannten Schwachstellen in Betriebssystempaketen und Abhängigkeiten. Ein Scan beschreibt Risiken, beweist aber nicht die Sicherheit des Images.

- Images aus vertrauenswürdigen Quellen beziehen und auf bekannte Schwachstellen prüfen
- Schweregrad, tatsächliche Nutzung und verfügbare Korretturen berücksichtigen
- Basissystem und Abhängigkeiten aktuell halten; unnötige Pakete entfenren
- Prüfergebnisse dem konkreten Image-Digest zuordnen

6. ARTEFAKTE SIGNIEREN



Eine Signatur kann hellen, Herkunft und Unverändertheit eines Artefakts zu prüfen. Cosign unterstützt das Signieren und Verifizieren von Container-Images und zugehörigen Nachweisen.

- Signieren ist nicht dasselbe wie ein Schwachstellenscan
- Verifikation muss gegen erwartete Identität oder Vertrauensrichtlinien erfolgen
- Signaturschlüssel und Identitäten geschützt verwalten; Prüfschritt vor Bereitstellung verankern

WICHTIG
Eine gültige Signatur allein bedeutet nicht, dass ein Artefakt fehlerfrei oder frei von Schadcode ist.

7. ZUGRIFFE UND POLICIES



Pipeline Access Controls legen fest, wer Workflows ändern, ausführen, freigeben oder auf Geheimnisse zugreifen darf. Policy as Code macht ausgewählte Regeln maschinenprüfbar.

- Least Privilege: nur notwendige Rechte und kurze Berechtigungsdauer
- Produktionsfreigaben und sensible Geheimnisse von untrusted Jobs trennen
- Regeln versionieren, testen und nachvollziehbar dokumentieren
- Ausnahmen begrenzen, begründen und mit Ablauf oder erneuter Prüfung versehen

8. FEEDBACK UND BETRIEB



Sicherheitskontrollen wirken besser, wenn Befunde verständlich, zuständigkeitskkar und zeitnah bearbeitet werden. Die Pipeline sollte aus Fehlern und Änderungen lernen.

- Befund → verantwortliche Person → Priorität → Korrektur oder dokumentierte Ausnahme
- Fehlalarme auswerten und Regeln gezielt verbessern, statt Checks pauschal abzuschalten
- Fehlgeschlagene Prüfungen und Freigaben protokollieren: Logs vor Geheimnisabfluss cchützen
- Regelmäßig prüfen, ob Scanner, Policies und Runner noch zum tatsächlichen Ablauf passen

ZUSAMMENFASSUNG
Früh prüfen, Runner und Rechte begrenzen, Infrastruktur und Images scannen, Artefakte verifizieren und Ergebnisse in einen verantwortlichen Rückkoppllungsprozess geben.

SOFTWARE-LIEFERKETTENSICHERHEIT

Schütze Software von der Auswahl ihrer Bausteine bis zu Build, Veröffentlichung und Betrieb. Ziel ist, Herkunft und Änderungen nachvollziehbar zu machen, Risiken zu verringern und bei Problemen gezielt reagieren zu können.



13. SLSA UND BUILD-PROVENANCE

SLSA ist ein Framework zur Verbesserung der Sicherheit von Softwarelieferketten. Provenance beschreibt nachvollziehbar, wie ein Artefakt erzeugt wurde. Aussagen über ein Niveau hängen von den jeweils erfüllten Anforderungen ab.

- Trenne Quellcode, Build-Ausführung und Veröffentlichungsrechte.
- Erzeuge Provenance für konkrete Artefakte und Builds.
- Prüfe Herkunft und Build-Identität vor der Weiterverwendung.
- Halte ungefähr [illegible] fest.
- Plane Verfügbarkeit, Backup und kontrollierte Bereinigung.
- nächsten Prüftermin.

12. CODE SIGNING UND HERKUNFT

Eine digitale Signatur kann die Integrität eines Artefakts und die Zuordnung zu einem Signaturschlüssel prüfbar machen. Sie beweist nicht, dass der Inhalt fehlerfrei oder ungefährlich ist.

- Signiere veröffentlichte Artefakte mit kontrollierten Schlüsseln.
- Prüfe Signatur und erwartete Identität vor der Installation.
- Schütze Schlüssel und beschränke Signierrechte.
- Plane Schlüsselrotation und den Umgang mit kompromittierten Schlüsseln.

10. VENDOR RISK ASSESSMENT

Eine Lieferantenbewertung betrachtet, wie ein Anbieter Software entwickelt, schützt, aktualisiert und Vorfälle kommuniziert. Sie sollte zum Schutzbedarf und zur Nutzung des Produkts passen.

- Welche Daten und Systeme erhält der Anbieterzugang?
- Wie werden Schwachstellen gemeldet und Updates bereitgestellt?
- Welche Nachweise, Kontrollen und Verantwortlichkeiten sind relevant?
- Wie funktionieren Kündigung, Datenexport und Reaktion auf Vorfälle?

9. UPSTREAM-MONITORING

Upstream-Monitoring beobachtet Projekte und Komponenten, von denen die eigene Software abhängt. So werden Sicherheitsmeldungen, Wartungsprobleme und relevante Änderungen früher sichtbar.

- Abonniere passende Sicherheitsmeldungen und Release-Informationen.
- Beobachte Projektaktivität, Maintainer-Wechsel und Archivierung.
- Ordne Meldungen dem eigenen Komponentenbestand zu.
- Definiere einen Prozess für Prüfung, Eskalation und Update.

9. UPSTREAM-MONITORING

Upstream-Monitoring beobachtet, wie ein Anbieter Komponenten entwickelt, denen, aktualisiert und Vorfälle So werden Sie Sicherheitsmeldungen, Wartungen und zur und relevante Änderungen früher sichtbar.

- Abonniere passende Sicherheitsmeldungen zugang?
- Wie werden Schwachstellen gemeldet und Updates bereichbarkeit.
- Pflanze Gebiete, Kontrollen und Verantwortlichkeite Risikoakzeptanz.
- Wie funktioniereine Kündigung, Datenexport und Reaktion auf Vorfälle?

8. SCHWACHSTELLEN UND PRIORISIERUNG

Ein Fund in einer Bibliotbeh bedeutet nicht automatisch, dass die eigene Anwendung verwundbar ist. Relevant sind unter andiemm betroffene Version, Nutzung des verwundbaren Codes und erreichbare Angriffswege.

- Ordne Fund, Paketversion und betroffene Funkiton ein.
- Berücksichtige Schweregrad und tatsächliche Erreichbarkeit.
- Plane Update, Workaround oder begründete Risikoakzeptanz.
- Dokumentiere Entscheidung, Zuständigkeit und nächsten Prüftermin.

2. SBOM: KOMPONENTEN SICHTBAR MACHEN

Eine Software Bill of Materials (SBOM) ist ein maschinenlesbares Inventar von Softwarekomponenten. Sie hilft, betroffene Produkte zu finden, beweist für sich allein aber weder Sicherheit noch Vollständigkeit.

- Erfasse Komponenten, Versionen und verfügbare Identifikatoren.
- Dokumentiere Erzeugungszeitpunkt, Werkzeug und betrachtetes Artefakt.
- Aktualisiere die SBOM nach relevanten Änderungen oder Builds.
- Nutze sie für Schwachstellen- und Lizenzprüfungen, nicht als Ersatz dafür.

14. BUILD-SYSTEME ABSICHERN

Kompromittierte Buildskripte und Geheimnisse können veröffentlichte Software maßgeblich beeinflussen. Ihre Absicherung gehört deshalb zur Lieferkettensicherheit.

- Nutze isolierte, regelmäßig aktualisierte Runner.
- Begrenze Geheimnisse auf erforderliche Jobs und Zeiträume.
- Schütze Build-Konfigurationen und Freigaben vor unbefugter Änderung.
- Halte Build-Protokolle und Artefaktbezüge nachvollziehbar.
- und dokumentiere [illegible]

7. VERSIONEN UND LOCKFILES

Versionsangaben steuern, welche Komponenten ein Build bezieht. Lockfiles halten die konkrete Auflösung fest und unterstützen reproduzierbare Installationen, ersetzen aber keine Vertrauens- oder Sicherheitsprüfung.

- Verwende für Anwendungen die vorgesehenen Lockfiles.
- Prüfe Änderungen daran im Code-Review.
- Aktualisiere Abhängigkeiten bewusst und nachvollziehbar.
- Beachte, dass nicht jeder Paketmanager identisches Versionsverhalten hat.

3. CYCLONEDX UND SPDX

CycloneDX und SPDX sind verbreitete Formate zum Austausch von Software-Komponenten- und Lieferketteninformationen. Die konkrete Aussagekraft hängt davon ab, welche Daten erzeugt und gepflegt werden.

- Wähle ein Format passend zu Werkzeugen und Austauschpartnern.
- Prüfe, ob Abhängigkeiten, Beziehungen und Lizenzen abgebildet sind.
- Bewahre SBOM und zugehöriges Release-Artefakt gemeinsam auf.
- Ein Format allein garantiert weder korrekte Daten, noch Konformität.

4. ABHÄNGIGKEITSRISIKEN ERKENNEN

Externe Bibliotheken sparen Entwicklungsaufwand, erweitern aber die Angriffsfläche. Risiken entstehen etwa durch bekannte Schwachstellen, aufgegebene Projekte, kompromittierte Konten oder unsichere Updates.

- Prüfe direkte und transitive Abhängigkeiten.
- Bevorzuge nachvollziehbare Herkunft und gepflegte Projekte.
- Bewerte Erreichbarkeit und Auswirkung einer Schwachstelle im eigenen Einsatz.
- Lege verantwortliche Personen und einen Aktualisierungsweg fest.

5. DEPENDENCY CONFUSION

Bei Dependency Confusion wird versucht, interne Paketnamen über öffentliche Registries mit gleichnamigen, vom Angreifer kontrollierten Paketen zu bedienen. Ob dies gelingt, hängt unter anderem von Namensauflösung und Paketmanager-[illegible]

- Vergib für interne Pakete kontrollierte, nicht erratbare Namensräume.
- Konfiguriere Paketquellen und Prioritäten ausdrücklich.
- Verhindere unbeabsichtigte Auflösung interner Namen aus öffentlichen Quellen.
- Überwache neue oder unerwartete Pakete mit internen Namen.

MERKE

Ein interner Name ist kein Zugriffsschutz. Entscheidend sind kontrollierte Quellen und eindeutige Auflösungsregeln.

6. TYPOSQUATTING UND PAKET-VERWECHSLUNG

Bei einem Verdacht nutzt ähnlich geschriebene Namen, damit Nutzende oder automatisierte Abläufe ein falsches Paket auswählen. Ähnliche Namen können auch durch Verwechslung entstehen, ohne dass ein Angriff vorliegt.

- Erfasse erkannte Lizenzen und Herkunft der Information.
- Erfasse erkannte Lizenzen und Herkunft der [illegible] Quelle abgleichen.
- Abhängigkeiten nach Möglichkeit exakt festlegen und [illegible] prüfen.
- Stoppe oder begrenze weitere Verteilung, wenn angemessen.
- Aktualisiere, widerrufe oder ersetze betroffene Artefakte und dokumentiere Maßnahmen.

6. REAKTION AUF F PAKET-VERWECHSLUNG

Bei einem Verdacht auf kompromittierte Komponente oder Build Nutzende oder[illegible] belegbare Reaktion. Zu falsche Paffang und betroffene Artefakte einigen auch durch Verwechslung entstehen, ohne dass ein Angriff vorliegt.

- Sichere Logs, SBOMs, Provenance und betroffene Artefakte.
- Identifiziere Builds, Releases und Umgebungen mit der Komponente.
- Bappe oder begrenze oder widersprüchliche Angaben bewerten.
- Nutze bei konkreten Rechtsfragen qualifizierte Rechtsberatung.

KRYPTOGRAFISCHE IMPLEMENTIERUNGSFEHLER UND BEST PRACTICES

Kryptografie schützt Daten nur dann zuverlässig, wenn Algorithmen, Zufall, Schlüssel und Betriebsverfahren korrekt eingesetzt werden. Die acht Konten zeigen häufige Fallstricke und sichere Grundentscheidungen.

- Kryptografie ersetzt keine Zugriffskontrolle oder sichere Softwareentwicklung.
- Nutze etablierte Bibliotheken und dokumentierte, aktuelle Verfahren statt eigener Kryptografie.

MERKE
Ein starker Algorithmus kann durch schwache Schlüsselverwaltung, unsichere Zufallswerte oder falsche Verwendung wirkungslos werden.

1. VERALTETE ALGORITHMEN VERMEIDEN

- Veraltete Verfahren bieten keinen angemessenen Schutz für neue Anwendungen.
- MD5 und SHA-1 gelten für kollisionsresistente Hash-Anwendungen als ungeeignet.
- DES hat einen zu kurzen Schlüssel; RC4 ist veraltet und sollte nicht neu eingesetzt werden.
- Für Passwortspeicherung: dafür vorgesehene adaptive Verfahren wie Argon2id, scrypt oder bcrypt verwenden.
- Für Datenintegrität und Signaturen geeignete moderne Hash- und Signaturverfahren aus einer aktuellen Bibliothek wählen.

2. SICHERE ZUFALLSWERTE

- Geheimschlüssel, Tokens und Nonces brauchen Zufallswerte aus einer kryptografisch sicheren Quelle.
- CSPRNG bedeutet kryptografisch sicherer Zufallszahlengenerator.
- Keine Standard- oder Simulations-PRNG für Geheimnisse verwenden.
- Betriebssystem- und Bibliotheksfunktionen für kryptografische Zufallswerte nutzen.
- Zufallswerte nicht selbst aus Uhrzeit, Prozess-ID oder Benutzername ableiten.

3. SALT UND IV RICHTIG EINSETZEN

- Salt und Initialisierungsvektor erfüllen unterschiedliche Aufgaben und sind nicht automatisch geheim.
- Ein Passwort-Salt ist pro Passwort einzigartig und wird mit dem gespeicherten Hash abgelegt.
- Ein IV oder Nonce muss die Anforderungen des gewählten Verschlüsselungsverfahrens erfüllen.
- Bei GCM darf ein Nonce mit demselben Schlüsssel nicht wiederverwendet werden.
- Werte nicht pauschal als geheim behandeln; Vertraulichkeit kommt vom Schlüsssel.

4. VERSCHLÜSSELUNGSMODI

- Der Modus bestimmt, wie ein Blockchiffre Daten verarbeitet und welche Sicherheitsmerkmale bereitgestellt werden.
- ECB verschlüsselt gleiche Klartextblöcke gleich und legt dadurch Muster offen; nicht für vertrauliche Daten verwenden.
- CBC benötigt einen geeigneten, unvorhersagbaren IV und separate Integritätssicherung.
- GCM bietet authentifizierte Verschlüsselung; Nonce-Eindeutigkeit ist entscheidend.
- Bevorzuge AEAD-Verfahren wie GCM oder ChaCha20-Poly1305, sofern sie zur Umgebung passen.

5. SCHLÜSSELVERWALTUNG

- Schlüssel müssen geschützt, zugänglich für Berechtigte und über ihren Lebenszyklus kontrolliert werden.
- Keine Schlüssel oder Passwörter hart codiert im Quelltext, Repository oder Container-Image ablegen.
- Geeignete Secret-Verwaltung oder Schlüsselmanagement-dienste nutzen.
- Zugriff nach dem Minimalprinzip begrenzen und Zugriffe überwachen.
- Rotation planen; bei Kompromittierung Schlüssel unverzüglich sperren oder ersetzen.
- Backups und Wiederherstellung sicher und nachvollziehbar organisieren.

6. TIMING-ANGRIFFE

- Laufzeitunterschiede können in bestimmten Situationen Informationen über geheime Werte preisgeben.
- Vergleich von MACs oder geheimen Tokens nicht mit gewöhnlichem frühzeitig abbrechendem Stringvergleich durchführen.
- Konstante-Zeit-Vergleichsfunktionen etablierter Bibliotheken verwenden.
- Fehlermeldungen und Antwortverhalten so gestalten, dass sie keine unnötigen Geheimnisinformationen offenbaren.
- Konstante Laufzeit allein garantiert keine Sicherheit; Gesamtsystem und Umgebung zählen.

7. BIBLIOTHEKSWAHL

- Bewährte Implementierungen reduzieren typische Fehler, müssen aber passend konfiguriert und gepflegt werden.
- Aktuelle, gepflegte Kryptografiebibliotheken und sichere Standardkonfigurationen verwenden.
- Keine eigene Implementierung Protokollen oder Schlüsselableitung schreiben.
- Protokollen verwenden.
- Fehlermeldungen und Antwortverhalten so gestalten, dass sie keine unnötigen Geheimnisinformationen offenbaren.
- Konstante Laufzeit allein garantiert keine Sicherheit; Gesamtsystem und Umgebung zählen.

8. PRÜF- UND BETRIEBSROUTINE

- Eine kurze systematische Prüfung hilft, Implementierungsfehler früh zu entdecken.
- Bedrohung und Schutzbedarf benennen: Was soll geheim, integer oder authentisch bleiben?
- Algorithmus, Modus, Schlüsselquelle und Lebenszyklus nachvollziehbar dokumentieren.
- Tests für Fehlerfälle, Nonce-Handling und Schlüsselwechsel vorsehen.
- Geheimnisse nicht in Logs, Fehlermeldungen oder Diagnoseausgaben schreiben.
- Bei Unsicherheit sichere Bibliotheksmuster und aktuelle Fachhinweise heranziehen.

KURZCHECK
Keine veralteten Verfahren, keine selbstgebauten Zufallswerte, keine Schlüssel im Code und keine wiederverwendeten GCM-Nonces.

KRYPTOGRAFISCHE IMPLEMENTIERUNGSFEHLER UND BEST PRACTICES

Sichere Kryptografie braucht passende Verfahren, Zufall, Schlüssel und sorgfältige Implementierung.

MINDMAP 041 – SIEM

041 SIEM

Sicherheitsereignisse sammeln, verknüpfen und untersuchen

1. ZWECK UND NUTZEN



SIEM unterstützt Sicherheitsüberwachung und Untersuchung, indem es Protokolldaten an einem Ort zusammenführt und analysierbar macht.

- Zentrale Sicht auf Ereignisse verschiedener Systeme
- Hinweise auf mögliche Sicherheitsvorfälle priorisieren
- Untersuchungen mit Zeitachsen und Suchabfragen unterstützen

2. DATENQUELLEN



Die Aussagekraft hängt davon ab, welche relevanten Systeme Ereignisse liefern und ob diese Daten verfügbar und verständlich sind.

- Betriebssysteme und Endgeräte: Anmeldungen, Prozesse, Sicherheitsmeldungen
- Netzwerkgeräte: Verbindungen, DNS- oder Firewall-Ereignisse
- Anwendungen und Cloud-Dienste: Zugriffe, Änderungen und Audit-Ereignisse
- Zeitstempel und Geräte- oder Benutzerbezüge erleichtern die Zuordnung

3. LOG-INGESTION



Ingestion bezeichnet den Weg von der Ereignisquelle bis zur Aufnahme in die SIEM-Plattform.

- Quellen über Agenten, Weiterleitung oder Schnittstellen anbinden
- Übertragungsweg und Zugriffsrechte angemessen absichern
- Verfügbarkeit, Verzögerungen und Ausfälle der Erfassung beobachten
- Datenumfang und Aufbewahrung nach Zweck und Richtlinien festlegen

Wichtig

Fehlende oder verspätete Logs können die Erkennung erschweren. Ein leeres Suchergebnis bedeutet nicht automatisch, dass nichts passiert ist.

4. NORMALISIERUNG UND KONTEXT



Quellen verwenden unterschiedliche Formate. Normalisierung ordnet Felder einer einheitlicheren Struktur zu; Kontext ergänzt Informationen, die bei der Bewertung helfen.

- Zeitangaben vereinheitlichen und Zeitzonen berücksichtigen
- Felder wie Benutzer, Host, Quell-IP und Ereignistyp zuordnen
- Anlagen- oder Kontoinformationen nur aus geeigneten Quellen ergänzen
- Originaldaten für Nachvollziehbarkeit gemäß Verfahren erhalten

5. KORRELATIONSREGELN



Korrelationsregeln verknüpfen Ereignisse nach definierten Bedingungen, Zeitfenstern und gegebenenfalls Entitäten wie Konto oder Gerät.

- Beispiel: mehrere fehlgeschlagene Anmeldungen, gefolgt von einer erfolgreichen
- Schwellenwerte, Zeitfenster und Ausnahmen an den Anwendungsfall anpassen
- Regeln mit sicheren Testdaten prüfen und Änderungen dokumentieren
- Zu breite Regeln erzeugen Fehlalarme; zu enge Regeln können Hinweise übersehen

Einordnung

Eine Regel liefert einen Prüfhinweis, keine sichere Feststellung eines Angriffs. Kontext und weitere Belege sind nötig.

6. ERKENNUNG UND ALARMIERUNG



Erkennung nutzt Regeln, Suchabfragen oder Analysefunktionen, um Ereignisse für eine weitere Prüfung hervorzuheben.

- Alarm mit Schweregrad, Zeit, betroffenen Systemen und Auslöser versehen
- Doppelte oder wenig aussagekräftige Alarme reduzieren und nachvollziehbar priorisieren
- Erkennungsabdeckung und bekannte Lücken regelmäßig prüfen
- Ein Alarm kann durch legitime Abläufe entstehen und muss validiert werden

7. UNTERSUCHUNG UND REAKTION

Analystinnen und Analysten prüfen den Alarm, sammeln Kontext und folgen dem festgelegten Vorfallprozess der Organisation.

- Zeitlinie aus relevanten Ereignissen erstellen
- Betroffene Konten, Geräte und Dienste abgrenzen
- Belege und Untersuchungsschritte nachvollziehbar dokumentieren
- Eindämmung oder weitere Maßnahmen nach Zuständigkeit und Verfahren veranlassen



Grenze der Mindmap

SIEM-Daten unterstützen Etscheindungen, ersetzen aber nicht die Untersuchung und die geltenden Reaktionsverfahren.

8. BETRIEB UND QUALITÄT



Ein SIEM braucht laufende Pflege: Quellen, Regeln, Zugriffe und Datenhaltung müssen zum Sicherheitsziel und zum Betriebsumfeld passen.

- Quellenabdeckung, Parser und Erfassungsfehler überwachen
- Regeln nach Vorfällen und Änderungen kontrolliert nachjustieren
- Zugriff auf Protokolle begrenzen; sensible Daten angemessen behandeln
- Suchleistung, Aufbewahrung und Zuständigkeiten regelmäßig überprüfen

042 · SOAR

SOAR verbindet Werkzeuge, Fälle und kontrollierte Reaktionen.

1. ZWECK UND EINSATZBEREICH



SOAR hilft Teams, Sicherheitsmeldungen in nachvollziehbare Fälle und koordinierte Arbeitsabläufe zu überführen. Es ergänzt bestehende Schutz- und Überwachungswerkzeuge, erstetzt aber weder deren Datenqualität noch menschliches Urteilsvermögen.

- Warnmeldungen bündeln und priorisieren
- Fallbearbeitung über Werkzeuge hinweg koordinieren
- Wiederkehrende, klar definierte Schritte automatisieren

2. ORCHESTRIERUNG



Orchestrierung ist das Zusammenspiel von Systemen, Daten und Aktionen. Eine SOAR-Plattform kann Informationen aus angebundenen Quellen zusammenführen und Aufgaben an andere Werkzeuge übergeben.

- Mögliche Quellen: SIEM, E-Mail-Sicherheit, Endpoint-Schutz und Threat-Intelligence-Dienste
- Integrationen verwenden je nach Produkt APIs, Webhooks oder andere Schnittstellen
- Datenfelder und Identitäten müssen vor der Zusammenführung sinnvoll abgeglichen werden

3. PLAYBOOKS



Ein Playbook ist ein dokumentierter Ablauf für einen bestimmten Falltyp. Es legt fest, welche Informationen gesammelt, welche Prüfungen durchgeführt und welche Aktionen unter welchen Bedingungen ausgelöst werden.

- Auslöser und Geltungsbereich definieren
- Schritte, Bedingungen und mögliche Abzweigungen festlegen
- Fehlerpfade, Zeitüberschreitungen und manuelle Übergaben berücksichtigen



KLEIN BEGINNEN

Ein klar begrenzter, risikoarmer Ablauf lässt sich leichter testen und überwachen als eine weitreichende automatische Reaktion.

4. TYPISCHER ABLAUF



Ein Fall durchläuft meist mehrere Phasen. Die konkrete Reihenfolge variiert je nach Organisation, eingesetzten Werkzeugen und Art des Sicherheitsereignisses.

- Erfassen: Warnmeldung und Kontext aufnehmen
- Anreichern: ergänzbare Informationen ergänzen
- Prüfen: Signalqualität und mögliche Auswirkungen bewerten
- Reagieren: freigegebene Maßnahmen ausführen oder zur Entscheidung vorlegen
- Abschließen: Ergebnis und Begründung dokumentieren

5. CASE MANAGEMENT



Case Management hält zusammengehörige Warnmeldungen, Befunde, Aufgaben und Entscheidungen in einem bearbeitbaren Fall fest. So kann das Team Fortschritt und Zuständigkeiten nachvollziehen.

- Eindeutige Fallkennung und Status verwenden
- Zuständige Personen und nächste Aufgaben festhalten
- Zeitstempel, relevante Belege und Entscheidungen dokumentieren
- Zugriff auf sensible Falldaten begrenzen

6. API-gestützte Reaktionsketten



Eine API ermöglicht den Austausch von Daten oder Befehlen zwischen Softwarediensten. Eine Reaktionskette kann beispielsweise einen Fall anlegen, Kontext abfragen und eine zur Freigabe vorgesehene Maßnahme übermitteln.

- Authentifizierung und Autorisierung für jede Integration passend konfigurieren
- Nur erforderliche Berechtigungen vergeben und Zugangsdaten schützen
- Fehler, Wiederholungen und doppelte Anfragen robust behandeln
- Anfragen und Ergebnisse nachvollziehbar protokollieren



KEINE BLINDEN AKTIONEN

Eine erfolgreiche API-Antwort beweist nicht automatisch, dass die gewünschte Sicherheitswirkung eingetreten ist. Ergebnis und Folgezustand müssen geprüft werden.

7. FREIGABEN UND SCHUTZMASSNAHMEN



Die Automatisierungstiefe sollte dem möglichen Schaden entsprechen. Informative oder leicht rückgängig zu machende Schritte können anders behandelt werden als Maßnahmen, die Konten, Systeme oder Geschäftsabläufe einschränken.

- Freigabe für folgenreiche Aktionen einplanen
- Berechtigungen auf den notwendigen Umfang begrenzen
- Abbruch- und Rücksetzverfahren vorsehen, soweit möglich
- Automatisierte Aktionen überwachen und regelmäßig überprüfen

Wie sollte der nächste Schritt ausgeführt werden?

- Nur Informationen sammeln oder eine reverisble, risikoarme Aufgabe erledigen
  → Im festgelegten Umfang automatisieren und Ergebnis protokollieren.
- Aktion kann Verfügbarkeit, Zugang oder Geschäftsbetrieb beeinträchtigen
  → Befund und vorgeschlagene Aktion zur menschlichen Freigabe vorlegen.
- Datenlage ist unvollständig oder widersprüchlich
  → Keine folgenreiche automatische Aktion auslösen; fehlenden Kontext ermitteln und Fall eskalieren.

8. QUALITÄT UND KONTINUIERLICHE VERBESSERUNG



SOAR-Abläufe benötigen Pflege. Tests und Auswertung helfen, fehlerhafte Bedingungen, veränderte Schnittstellen und unerwünschte Nebenwirkungen zu erkennen.

- Playbooks mit kontrollierten Testfällen prüfen
- Fehlalarme, Abbrüche und manuelle Übergaben auswerten
- Änderungen an APIs, Berechtigungen und Datenformaten nachverfolgen
- Abläufe nach Vorfällen und veränderten Anforderungen anpassen

043 · INCIDENT-RESPONSE-LEBENSZYKLUS

Vorbereiten · erkennen · eindämmen · beseitigen · wiederherstellen · lernen

1 · VORBEREITUNG

- Rollen, Kontaktwege, Werkzeuge und Abläufe vor einem Vorfall festlegen und üben.



2 · ERKENNUNG UND ANALYSE

- Hinweise prüfen, den möglichen Umfang einschätzen und relevante Informationen dokumentieren.



3 · EINDÄMMUNG

- Geeignete Maßnahmen zur Begrenzung weiterer Auswirkungen abstimmen.



4 · BESEITIGUNG UND WIEDERHERSTELLUNG

- Ursachen beseitigen, Systeme kontrolliert wieder in Betrieb nehmen. und beobachten.



5 · WIEDERHERSTELLUNG

- Kontrolliert zurück in den Betrieb.



6 · WIEDERHERSTELLUNG

- Systeme stabilisieren und den Normalbetrieb vollständig wiederherstellen.



7 · KOMMUNIKATION & DOKUMENTATION

- Zuständigkeiten, Zeitlinie und Entscheidungen festhalten.



8 · NACHBEREITUNG

- Ablauf und Wirkung der Maßnahmen auswerten und Verbesserungen in die Vorbereitung übernehmen.



INCIDENT RESPONSE: VIER PHASEN



Incident Response (IR) bezeichnet das organisierte Vorgehen bei vermuteten oder bestätigten Sicherheitsvorfällen. Das NIST-Modell unterscheidet Vorbereitung, Erkennung und Analyse, Eindämmung sowie Beseitigung und Wiederherstellung. Die Nachbereitung unterstützt Verbesserungen und verbindet den konkreten Vorfall mit der Vorbereitung auf künftige Ereignisse.

- Ziel: Auswirkungen begrenzen, Abläufe koordinieren und Erkenntnisse sichern.
- Die Phasen sind ein Orientierungsmodell; in der Praxis können Tätigkeiten parallel laufen oder erneut nötig sein.
- Arbeiten nach geltenden Zuständigkeiten, Richtlinien und rechtlichen Vorgaben.

BEGRIFFSHINWEIS



Diese Mindmap vermittelt ein allgemeines Lernmodell. Sie ist keine Diagnose eines konkreten Sicherheitsvorfalls und keine individuelle Handlungs- oder Rechtsberatung.

044 · CTI & BEDROHUNGSMODELLIERUNG

Cyber Threat Intelligence (CTI) ordnet Informationen über Bedrohungen ein, damit Organisationen Risiken besser verstehen und Entscheidungen treffen können. Bedrohungsmodellierung betrachtet systematisch, was geschützt werden soll, wer es bedrohen könnte und wie Risiken reduziert werden.

- CTI verbindet Beobachtungen mit Kontext und möglicher Bedeutung.
- Bedrohungsmodellierung ist ein strukturierter Prozess, keine Vorhersage mit Gewissheit.

1. ZWECK UND FRAGEN



Gute Analyse beginnt mit einer konkreten Entscheidungsfrage. Der Zweck bestimmt, welche Informationen wichtig sind und für wen sie aufbereitet werden.

- Welche Systeme oder Daten sollen geschützt werden?
- Welche Entscheidung soll die Analyse unterstützen?
- Wer benötigt das Ergebnis: Technikteam, Leitung oder weitere Partner?
- Welche Annahmen und Grenzen gelten?

2. QUELLEN UND DATENSAMMLUNG



Hinweise können aus internen Ereignisprotokollen, Sicherheitsmeldungen, Warnungen von Herstellern oder öffentlich zugänglichen Berichten stammen. Herkunft und Zuverlässigkeit müssen mitgedacht werden.

- Quelle, Zeitpunkt und Erhebungsmethode dokumentieren.
- Nur benötigte Daten sammeln und Zugriffsrechte beachten.
- Beobachtung von Schlussfolgerung unterscheiden.
- Unabhängige Hinweise können eine Einschätzung stärken.

EINORDNUNG
Ein einzelner Bericht oder Datenpunkt beweist nicht, dass ein Angriff stattugfunden hat.

3. INDIKATOREN UND KONTEXT



Ein Indikator für eine Kompromittierung (IOC) ist ein beobachtbares Merkmal, das auf verdächtige Aktivität hindeuten kann. Indikatoren sind kontextabhängig und können veralten oder auch harmlose Ursachen haben.

- Beispiele: bekannte schädliche Datei-Hashes, verdächtige Domains oder ungewöhnliche Netzwerkverbindungen.
- Zeitfenster, betroffene Systeme und Erfassungsquelle festhalten.
- Indikatoren mit weiteren Ereignissen korrelieren.
- Ein Treffer ist ein Anlass zur Prüfung, nicht automatisch ein Beweis.

4. ANALYSE UND EINSCHÄTZUNG



Analystinnen und Analysten verbinden einzelne Beobachtungen zu einer nachvollziehbaren Hypothese. Unsicherheit sollte sichtbar bleiben; alternative Erklärungen gehören in die Bewertung.

- Was wurde direkt beobachtet?
- Welche Erklärung passt zu den verfügbaren Hinweisen?
- Welche Gegenhypothese ist plausibel?
- Konfidenz und offene Fragen ausdrücklich bennenen.

5. DIAMOND-MODELL



Das Diamond-Modell strukturiert eine Aktivität anhand von vier Kernmerkmalen: Angreifer, Fähigkeit, Infrastruktur und Opfer. Es hilft, Beziehungen zwischen Beobachtungen zu beschreiben.

- Angreifer: vermuteter Akteur oder Gruppierung; Zuordnung bleibt oft unsicher.
- Fähigkeit: eingesetzte Werkzeuge, Techniken oder Kenntnisse.
- Infrastruktur: Systeme und Dienste, die bei der Aktivität genutzt werden.
- Opfer: betroffene Organisationen, Personen oder Systeme.

VORSICHT BEI ZUORDNUNGEN
Ähnliche Werkzeuge oder Infrastruktur reichen allein nicht aus, um einen Akteur sicher zu identifizieren.

6. BEDROHUNGSMODELLIERUNG



Ein Bedrohungsmodell macht Angriffsflächen, Schutzgüter und mögliche Risiken sichtbar. Es wird regelmäßig angepasst, wenn sich System, Nutzung oder Bedrohungslage ändern.

- Systemgrenzen und Datenflüsse skizzieren.
- Schutzgüter und Vertrauensgrenzen markieren.
- Mögliche Angreifer und Ziele betrachten.
- Risiken nach Auswirkung und Plausibilität priorisieren.

7. INFORMATIONSAUSTAUSCH



CTI kann innerhalb einer Organisation oder mit anderen Stellen geteilt werden, Austausch ist nur nützlich, wenn Empfänger den Inhalt verstehen und sicher verwenden können.

- Relevante Erkenntnisse knapp und handlungsorientiert formulieren.
- Quelle, Zeitbezug, Konfidenz und Einschränkungen beibehalten.
- Vertraulichkeit, Datenschutz und Freigaberegeln berücksichtigen.
- Maschinenlesbare Formate können strukturierte Weitergabe erleichtern.

8. REAKTION UND AKTUALISIERUNG



Erkenntnisse sollen in angemessene Schutzmaßnahmen und erneute Beobachtung münden. Nach einer Reaktion wird geprüft, ob die Annahmen weiterhin tragen.

- Passende Abwehrmaßnahme und Zuständigkeit festlegen.
- Ergebnis und Auswirkungen überwachen.
- Neue Hinweise in Analyse und Modell zurückführen.
- Lernen dokumentieren, ohne Vermutungen als Fakten auszugeben.

045 DIGITALE FORENSIK

Spuren nachvollziehbar sichern; Vorfälle sicher einordnen.

Wichtiger Hinweis

Bei akuter Gefahr haben Menschen- und Betrebssicherheit Vorrang. Eingriffe können flüchtige Spuren verändern; wenn möglich, vor Maßnahmen zuständige Incident-Response- oder Forensik-Fachleute einbeziehen.

1. AUFTRAG UND UMFANG

Vor der Untersuchung klären, was bekannt ist, welche Systeme betroffen sein könnten und welche Befugnisse gelten.



- Meldung, Zeitpunkt und beobachtete Symptome erfassen.
- Betroffene Geräte, Konten und Dienste eingrenzen.
- Zuständigkeiten, Freigaben und Datenschutz beachten.

2. LAGE PRIORISIEREN

Die erste Einechätzung dient der sicheren Priorisierung, nicht dem voreiligen Beweis einer Ursache.



- Gibt es laufenden unbefugten Zugriff oder Datenabfluss?
- Sind kritische Dienste oder weitere Systeme gefährdet?
- Welche unmittelbaren Schutzmaßnahmen sind verhältnismäßig?

3. FLÜCHTIGE SPUREN

Arbeitsspeicher, aktive Prozesse und Netzwerkverbindungen können nach dem Ausschalten verloren gehen. Die Erhebung solcher Daten erfordert geeignete Werkzeuge und geschultes Vorgehen.



- Zeitpunkt und Beobachtungen festhalten.
- Vor Neustart oder Ausschalten Auswirkungen abwägen.
- Erhebung und verwendete Werkzeuge dokumentieren.

4. BEWEISMITTEL SICHERN

Originale möglichst unverändert erhalten. Bei Datenträgern wird, soweit möglich und angemessen, ein forensisches Abbild erstellt; Analyse erfolgt an einer Arbeitskopie.



- Beweismittel eindeutig kennzeichnen und Zustand notieren.
- Zugriffe begrenzen und sichere Aufbewahrung gewährleisten.
- Hashwerte können die Übereinstimmung einer Kopie mit einer später geprüften Version unterstützen.

5. CHAIN OF CUSTODY

Die Beweismittelkette dokumentiert Besitz, Übergaben und relevante Handlungen an jedem Beweismittel. Sie unterstützt Nachvollziehbarkeit, erstetzt aber keine korrekte Erhebung.



- Kennung, Quelle, Datum und Uhrzeit mit Zeitzone erfassen.
- Jede Übergabe mit abgebender und übernehmender Person dokumentieren.
- Zugriffe, Verpackung, Speicherort und Abweichungen protokollieren.

6. INTEGRITÄT UND DOKUMENTATION

Eine nachvollziehbare Dokumentation trennt Beobachtungen von Interpretationen und hält Unsicherheiten sichtbar.



- Originaldaten und Arbeitskopien klar unterscheiden.
- Hashverfahren, Zeitpunkt und Ergebnis festhalten; ein Hash beweist allein weder Herknuft noch Vollständigkeit.
- Befunde, Zeitzonen, Werkzeugversionen und Entscheidungen notieren.

7. EINDÄMMUNG MIT BEDACHT

Eindämmung reduziert weiteren Schaden, kann jedoch Spuren verändern. Entscheidungen mit Verantwortlichen abstimmen und Auswirkungen dokumentieren.



- Zugänge oder Verbindungen gezielt begrenzen, wenn dies angemessen ist.
- Keine Beweise löschen oder Systeme unnötig bereinigen.
- Erforderliche Wiederherstellung und Geschäftskontinuität berücksichtigen.

8. ÜBERGABE UND NÄCHSTE SCHRITTE

Die Triage endet mit einem verständlichen Lagebild und klaren offenen Punkten. Vertiefte Analyse folgt einem autorisierten, dokumentierten Plan.



- Gesicherte Daten und ihre Aufbewahrung übergeben.
- Bekanntes, Vermutungen und offene Fragen getrennt berichten.
- Weitere Untersuchung, Schutzmaßnahmen und Kommunikation abstimmen.

046 · MEMORY FORENSICS

RAM-Analyse untersucht den Arbeitsspeicher eines laufenden oder zuvor erfassten Systems. Sie kann Spuren flüchtiger Aktivitäten sichtbar machen, die auf einem Datenträger fehlen.

1. ZIEL DER ANALYSE



Die Untersuchung sucht nach flüchtigen Hinweisen auf laufende Prozesse, Verbindungen und Veränderungen im Systemzustand.

- Prozesse und geladene Komponenten einordnen
- Verdächtige Abweichungen als Hypothesen prüfen
- Befunde mit weiteren Quellen abgleichen

2. FLÜCHTIGKEIT



RAM verliert seinen Inhalt typischerweise, wenn die Stromversorgung endet. Auch während des Betriebs werden Speicherbereiche verändert oder wiederverwendet.

- Ausschalten kann flüchtige Spuren beseitigen.
- Analysewerkzeuge verändern möglicherweises den untersuchten Zustand.
- Zeitpunkt und Vorgehen deshalb dokumentieren.

3. SICHERUNG



Ein Speicherabblid hält den Zustand zu einem bestimmten Zeitpunkt fest. Die Erhebung sollte autorisiert und nachvollziehbar erfolgen.

- System, Uhrzeit und Erhebungsumstände notieren
- Werkzeug und Version festhalten
- Abblld geschützt speichern und Integrität per Hashwert prüfen

4. PROVENIENZ



Forensische Aussagekraft hängt auch davon ab, woher die Daten stammen und wie sie verarbeitet wurden.

- Originalabblid unverändert aufbewahren
- Arbeitskopie für die Auswertung verwenden
- Schritte, Prüfsummen und Zuständigkeiten protokollieren

5. ANALYSEEBENEN



Die Auswertung verbindet unterschiedliche Ansichten auf dasselbe Abblid. Einzelne Artefakte sind selten allein beweiskräftig.

- Prozessliste und Prozessbeziehungen
- Module, Treiber und Speicherbereiche
- Netzwerkverbindungen und offene Handles
- Befunde im Kontext von Betriebssystem und Zeitlinie

6. CODE-INJEKTION



Bei einer Code-Injektion wird Code in einen Prozess oder dessen Speicher eingebracht oder dort ausgeführt. Auffälligkeiten können Anlass zur Prüfung sein, beweisen aber für sich keinen Angriff.

- Unerwartete ausführbare Speicherbereiche untersuchen
- Abweichungen zwischen Speicherinhalt und Datei berücksichtigen
- Prozessverhalten und weitere Telemetrie ergänzend prüfen



NICHT VORSCHNELL URTEILEN

Legitime Laufzeittechniken und Schutzsoftware können ungewöhnliche Speichermerkmale erzeugen. Ein einzelnes Signal ist kein sicherer Nachweis.

7. ROOTKIT-HINWEISE



Rootkits versuchen, ihre Präsenz oder Aktivitäten zu verbergen. Ein Vergleich unabhängiger Ansichten kann Inkonsistenzen aufdecken.

- Abweichungen zwischen Prozess- und Speicheransichten beachten
- Kernel- oder Treiberartefakte im Systemkontext prüfen
- Auch Fehlalarme, Beschädigung und Werkzeuggrenzen erwägen

8. BEFUND BEWERTEN



Ein belastbarer Befund trennt Beobachtung, Interpretation und Schlussfolgerung. Unsicherheit und Grenzen gehören ausdrücklich dazu.

- Beobachtung präzise beschreiben
- Alternative Erklärungen prüfen
- Zeitpunkt und Datenquelle angeben
- Ergebnis mit unabhängigen Belegen korrelieren

047 · WINDOWS-FORENSIK

Digitale Spuren sichern, einordnen und zeitlich verknüpfen – nachvollziehbar und mit Blick auf Grenzen der Daten.

- Ziel: Hinweise finden, nicht vorschnell Schuld zuweisen.
- Kontext, Integrität und Dokumentation sind entscheidend.



1 · VORGEHEN



Strukturiert arbeiten, bevor Daten verändert werden.

- Auftrag und Umfang klären; geltende Berechtigungen beachten.
- Zustand, Uhrzeit, Zeitzone und sichtbare Hinweise dokumentieren.
- Reihenfolge und Maßnahmen protokollieren; Änderungen vermeiden.
- Bei möglicher Beweissicherung zuständige Fachleute und Verfahren einbeziehen.

2 · FLÜCHTIGE DATEN



Manche Informationen gehen beim Ausschalten oder mit der Zeit verloren.

- Arbeitsspeicher kann laufende Prozesse und flüchtige Inhalte enthalten.
- Aktive Verbindungen und angemeldete Sitzungen liefern Momentaufnahmen.
- Erhebung kann Spuren verändern: Vorgehen und Zeitpunkt festhalten.
- Nicht pauschal herunterfahren; Entscheidung nach Auftrag und Verfahren.

3 · DATENTRÄGER UND DATEISYSTEM



Dateisystemdaten beschreiben Dateien und ihre Ablage, sind aber nicht die Dateien selbst.

- NTFS-Metadaten: unter anderem MFT-Einträge und Zeitstempel.
- USN Change Journal: protokolliert Änderungen, nicht zwingend deren Ursache.
- Papierkorb und Schattenkopien können zusätzliche Kontextspuren enthalten.
- Gelöschte Daten können überschrieben oder nicht mehr verfügbar sein.

4 · AUSFÜHRUNGSSPUREN



Artefakte können auf Programmstarts oder -präsenz hinweisen; keines beweist allein eine Handlung.

- Prefetch kann Hinweise auf Programmstarts liefern; Verfügbarkeit und Aussagekraft variieren.
- Amcache enthält Inventar- und Kompatiblitäts-informationen, keine sichere Ausführungsliste.
- UserAssist erfasst bestimmte GUI-Aktivitäten nutzerbezogen.
- Artefakte im Zusammenhang und mit Systemversion interpretieren.

5 · ANMELDUNG UND REGISTRY



Windows speichert Konfigurations- und Aktivitätshinweise an verschiedenen Stellen.

- Security-Ereignisprotokolle können Anmeldeereignisse enthalten; Richtlinien beeinflussen die Erfassung.
- Registry-Hives und ihre Transaktionsprotokolle zeigen Konfiguration und Änderungen.
- BAM/DAM können Hinweise auf jüngste Programmnutzung enthalten; Aussagekraft ist begrenzt.
- Benutzerkontext, Ereignisquelle und Zeitangaben genau zuordnen.

6 · ZEIT UND KORRELATION



Eine belastbare Zeitleiste verbindet unabhängige Spuren und macht Unsicherheiten sichtbar.

- Windows-Zeitstempel können unterschiedliche Semantik haben; Dateisystemzeit ist oft UTC-basiert.
- Zeitzone, Sommerzeit, Systemuhr und mögliche Abweichungen berücksichtigen.
- Ereignisprotokolle, Dateimetadaten und Netzwerkhinweise vergleichen.
- Widersprüche und fehlende Daten kennzeichnen, nicht weginterpretieren.

7 · INTEGRITÄT UND DOKUMENTATION



Nachvollziehbarkeit entsteht durch reproduzierbare Schritte und klare Herkunft der Daten.

- Quelle, Erhebungszeit, Werkzeug und Version dokumentieren.
- Kryptografische Hashwerte können die Unveränderheit einer erfassten Kopie prüfen.
- Hashgleichheit belegt nicht automatisch Vollständigkeit oder ursprüngliche Herkunft.
- Originale schützen; Auswertung möglichst an Arbeitskopien durchführen.

8 · GRENZEN UND ZUSAMMENFASSUNG



Windows-Artefakte sind kontextabhängige Hinweise, keine isolierten Beweise für Absicht oder Identität.

- Betriebssystemversion, Konfiguration, Aufbewahrung und Bereinigung beeinflussen Spuren.
- Fehlende Artefakte beweisen nicht, dass ein Ereignis nicht stattgefunden hat.
- Mehrere unabhängige Hinweise stärken eine Hypothese; alternative Erklärungen prüfen.
- Ergebnis, Unsicherheit und Grenzen verständlich festhalten.

048 LINUX-FORENSIK UND LOG-UNTERSUCHUNG

Dateisystem-Artefakte · Systemprotokolle · Audit-Spuren

1. ZIEL UND UMFANG

Vor Beginn den Untersuchungsauftrag und den zulässigen Umfang festhalten.



- Fragestellung und betroffene Systeme benennen.
- Zeitfenster, Datenquellen und Berechtigungen klären.
- Nur erforderliche Daten erfassen; Maßnahmen protokollieren.

2. FLÜCHTIGE DATEN

Manche Informationen ändern sich mit laufendem Betrieb oder gehen beim Ausschalten verloren.



- Bei autorisierter Untersuchung zuerst flüchtige Daten erfragen.
- Systemzeit und mögliche Abweichung zu einer Referenz dokumentieren.
- Erhebungsschritte, Werkzeug und Zeitpunkt festhalten.

3. DATENTRÄGER UND DATEISYSTEM

Dateisystemspuren können Abläufe und Veränderungen sichtbar machen, sind aber kontextabhängig.



- Metadaten wie Zeitstempel, Eigentümer und Berechtigungen prüfen.
- Zeitstempel nicht allein als Beweis für eine bestimmte Handlung behandeln.
- Soweit möglich mit schreibgeschütztem Zugriff oder forensischer Kopie arbeiten.

4. LINUX-PROTOKOLLE

Protokolle liegen je nach Distribution, Dienst und Konfiguration an unterschiedlichen Stellen.



- Journal-Daten mit journalctl untersuchen, sofern systemd-Journald genutzt wird.
- Textdateien unter /var/log prüfen. Namen und Umfang variieren.
- Rotation, Aufbewahrung und mögliche Lücken berücksichtigen.

5. AUTHENTIFIZIERUNG UND SITZUNGEN

Anmeldeereignisse liefern Hinweise, aber keine vollständige Erklärung der Ursache.



- Erfolgreiche und fehlgeschlagene Anmeldungen zeitlich vergleichen.
- Je nach System etwa auth.log, secure oder Journal-Einträge prüfen.
- Benutzer, Dienst, Quelladresse und Sitzungskontext gemeinsam betrachten.

6. AUDIT-SPUREN

Audit-Regeln bestimmen wesentlich, welche Ereignisse protokolliert werden.



- Vorhandene Audit-Konfiguration und Status erfassen.
- Audit-Ereignisse nach Zeit, Benutzerkennung und Ereignistyp filtern.
- Fehlende Einträge bedeuten nicht automatisch, dass kein Ereignis stattfand.

7. KORRELATION UND INTEGRITÄT

Mehrere unabhängige Quellen können eine belastbarere zeitliche Rekonstruktion ermöglichen.



- Zeitstempel mit Zeitzone und möglicher Uhrabweichung vergleichen.
- Dateisystem, Dienste, Anmeldungen und Audit-Spuren abgleichen.
- Originale unverändert bewahren; Kopien und Prüfsummen nachvollziehbar dokumentieren.

8. BEFUNDE UND GRENZEN

Ergebnisse klar von Hypothesen trennen und Unsicherheiten sichtbar machen.



- Beobachtung, Interpretation und offene Frage getrennt notieren.
- Quelle, Erhebungszeitpunkt und relevante Einschränkungen angeben.
- Nur Schlussfolgerungen ziehen, die von den vorhandenen Spuren getragen werden.

049 NETZWERK-FORENSIK & PCAP-ANALYSE

Mitschnitte sichern, Kommunikation rekonstruieren, Auffälligkeiten vorsichtig einordnen.

1. AUFTRAG UND UMFANG



Vor der Analyse klären, welche Frage beantwortet werden soll und welche Daten dafür zulässig sind.

- Fragestellung und Zeitraum festhalten.
- Erfassungspunkt, Zeitzone und bekannte Lücken notieren.
- Nur autorisierte Mitschnitte verwenden; Zugriffe und Kopien begrenzen.

2. BEWEISDATEN SICHERN



Originaldaten unverändert erhalten und jeden Arbeitsschritt nachvollziehbar dokumentieren.

- Datei, Herkunft, Erfassungszeit und übergebende Stelle protokollieren.
- Kryptografischen Hash, zum Beispiel SHA-256, zur Integritätskontrolle berechnen.
- Mit einer Arbeitskopie analysieren; Hash und Änderungen dokumentieren.

3. PCAP UND ERFASSUNG



Ein Paketmitschnitt speichert je nach Konfiguration Paket-Header und Teile oder den gesamten Paketinhalt.

- PCAP und PCAPNG sind verbreitete Formate für Paketaufzeichnungen.
- Snap length, Filter, Paketverluste und Schnittstellen beeinflussen die Aussagekraft.
- Verschlüsselter Verkehr bleibt auf Anwendungsebene meist unlesbar, sofern keine zulässigen Entschlüsselungsdaten vorliegen.

4. FLOW-DATEN UND ZEIT



Flows bieten Überblick über Kommunikationsbeziehungen, ersetzen aber nicht die Paketanalyse.

- Flow-Datensätze enthalten typischerweise Adressen, Ports, Transportprotokoll sowie Zeit- und Zählerwerte.
- NetFlow und IPFIX sind Formate für Flow-Informationen; konkrete Felder variieren.
- Zeitstempel, Zeitzone, Uhrenabweichung und Aggregationsintervall beim Vergleich berücksichtigen.

5. PROTOKOLLE REKONSTRUIEREN



Schichtenweise arbeiten: Link, IP, Transport und – soweit sichtbar – Anwendung.

- Ethernet oder WLAN, IP und TCP/UDP getrennt betrachten; Tunnel können zusätzliche Schichten einführen.
- TCP-Verbindungen anhand von Sequenznummern und Richtungen zusammensetzen; Verluste und Retransmits beachten.
- DNS-Abfragen, TLS-Handshakes und HTTP-Metadaten nur auswerten, wenn sie im Mitschnitt sichtbar sind.

6. AUFFÄLLIGKEITEN PRÜFEN



Hypothesen an mehreren Merkmalen testen, statt einzelne Pakete vorschnell als Angriff zu deuten.

- Ungewöhnliche Verbindungsraten, Ziele, Ports oder Fehlerbilder mit einer Baseline vergleichen.
- DNS-Antworten, Verbindungsabbrüche und wiederholte Anfragen im zeitlichen Zusammenhang untersuchen.
- Scanner, Fehlkonfigurationen und legitime Last können ähnliche Muster erzeugen.

7. WERKZEUGE UND FILTER



Filter erleichtern die Sichtung, doch ein enger Filter kann relevante Pakete ausblenden.

- Wireshark eignet sich für interaktive Paketinspektion; TShark ermöglicht Kommandozeilen-Auswertung.
- Capture-Filter wirken bei der Aufzeichnung; Display-Filter beschränken die angezeigte Analyse.
- Filterbedingungen und Werkzeugversion dokumentieren; Ergebnisse am Rohmitschnitt gegenprüfen.

8. BEFUNDE BERICHTEN



Bericht trennt Beobachtung, Interpretation und Unsicherheit klar voneinander.

- Zeit, Endpunkte, Protokoll und konkrete Paket- oder Flow-Belege nennen.
- Alternative Erklärungen, Erfassungslücken und Verschlüsselungsgrenzen aufführen.
- Nur belegte Schlussfolgerungen ziehen; sensible Nutzdaten minimieren oder schwärzen.

EINORDNUNG

Analyse dient der autorisierten Verteidigung und Untersuchung. Paketdaten können sensible Inhalte enthalten; Zugriff und Aufbewahrung müssen entsprechend geschützt werden.

MALWARE-ANALYSE – STATISCHE VERFAHREN

Untersuche verdächtige Dateien, ohne sie auszuführen. Ziel ist eine erste, nachvollziehbare Risikoeinschätzung – kein endgültiger Beweis für Sicherheit oder Schadwirkung.

- Datei unverändert sichern und Herkunft dokumentieren.
- Befunde aus mehreren Verfahren zusammenführen; Unsicherheit benennen.



1. ZIEL UND GRENZEN



Statische Verfahren untersuchen Eigenschaften und Inhalte einer Datei, ohne ihren Code zu starten.

- Hilft, Struktur, verdächtige Merkmale und mögliche Funktionen zu erkennen.
- Packen, Verschlüsselung oder Tarnung können Inhalte verbergen.
- Ein einzelner Hinweis beweist weder Schadwirkung noch Harmlosigkeit.

2. BEWEISSICHERUNG



Halte den Ausgangszustand fest, damit Analyseschritte nachvollziehbar bleiben.

- Dateiname, Größe, Herkunft und Erfassungszeit notieren.
- Original schreibgeschützt aufbewahren; Analysen an einer Kopie durchführen.
- Prüfsumme, etwa SHA-256, berechnen und protokollieren.
- Zugriffe und Veränderungen dokumentieren.

3. DATEITYP UND METADATEN



Prüfe, was die Datei tatsächlich enthält, statt dich allein auf die Dateiendung zu verlassen.

- Endung, erkannte Dateisignatur und Format miteinander vergleichen.
- Größe, Zeitstempel, eingebettete Ressourcen und ungewöhnliche Eigenschaften erfassen.
- Abweichungen sind Prüfhinweise, aber für sich kein Schadensnachweis.

4. STRINGS UND EINGEBETTETE HINWEISE



Lesbare Textfragmente können Anhaltspunkte für Funktion oder Herkunft liefern.

- Suche nach URLs, Domänen, Dateipfaden, Fehlermeldungen oder Befehlsfragmenten.
- Kodierte oder komprimierte Inhalte können in einer einhachen Suche fehlen.
- Ein gefundener Text kann ungenutzt, irreführend oder zufällig enthalten sein.

5. STRUKTUR UND KOMPONENTEN



Untersuche Aufbau und eingebettete Bestandteile passend zum Dateiformat.

- Bei ausführbaren Formaten: Abschnitte, Importe, Exporte und Ressourcen betrachten.
- Bei Archiven und Dokumenten: eingebettete Dateien, Makros oder Skripte erfassen.
- Ungewöhnliche Struktur oder fehlende Konsistenz als Hypothese prüfen, nicht als Urteil.

6. HASHES UND ERKENNUNG



Hashes dienen dem Abgleich exakt gleicher Inhalte; Erkennungsnamen liefern zusätzliche, kontextabhängige Hinweise.

- Ein kryptografischer Hash identifiziert den geprüften Datenstand, nicht dessen Sicherheit.
- Erkennungsdienste können Fehlalarme oder übersehene Varianten aufweisen.
- Datei-Hashse nicht mit kryptografischen Signaturen verwechseln.

7. ERGEBNISSE EINORDNEN



Bewerte Indizien gemeinasm und trenne Beobachtung von Schlussfolgerung.

- Beobachtung: etwa ein verdächtiger String oder eine unerwartete Komponente.
- Hypothese: mögliche Funktion, die durch weitere Befunde gestützt werden muss.
- Konfidenz und alternative Erklärungen angeben; keine Gewissheit aus einem Merkmal ableiten.

8. NÄCHSTE SCHRITTE



Leite aus Befundlage und Risiko einen angemessenen, sicheren Umgang ab.

- Bei konkreten Verdachtsmomenten: Datei nicht öffnen oder ausführen und zuständige Sicherheitsstelle einbeziehen.
- Falls erforderlich, dynamische Analyse nur kontrolliert und autorisiert durchführen.
- Ergebnis, Grenzen und offene Fragen verständlich dokumentieren.

SICHERHEITSGRENZE



Eine statische Analyse vermeidet die Ausführung der Datei, aber der Umgang mit verdächtigen Dateien kann dennoch riskant sein. Nutze dafür freigegebene, isolierte Analyseumgebungen und beachte die geltenden Regeln.

MALWARE-ANALYSE: DYNAMISCHE SANDBOX-VERFAHREN

Eine Sandbox führt verdächtige Dateien kontrolliert aus, damit ihr Verhalten beobachtet werden kann. Sie ergänzt die statische Analyse, ersetzt aber keine umfassende Sicherheitsprüfung.

- Ziel: Verhalten sichtbar machen, ohne produktive Systeme zu gefährden.
- Ergebnisse hängen von Umgebung, Beobachtungszeit und beobachteten Aktivitäten ab.

SICHERHEITSGRENZE

Nur autorisierte, isolierte Analyseumgebungen verwenden. Eine Sandbox ist kein absoluter Schutz vor Ausbruch oder Folgeschäden.

1. VORBEREITUNG

Eine reproduzierbare Ausgangslage schafft vergleichbare Beobachtungen.

- Betriebssystem, Version und installierte Programme notieren.
- Uhrzeit, Netzwerkmodus und Sandbox-Konfiguration festhalten.
- Datei-Hash berechnen und Probe eindeutig kennzeichnen.

2. ISOLIERTE UMGEBUNG

Die virtuelle Maschine oder Analyseumgebung von produktiven Netzen und sensiblen Daten trennen.

- Snapshots erleichtern das Zurücksetzen auf einen sauberen Zustand.
- Netzwerkzugriff standardmäßig sperren oder kontrolliert simulieren.
- Gemeinsame Ordner, Zwischenablage und Host-Integration minimieren.

3. KONTROLLIERTE AUSFÜHRUNG

Die Probe nur innerhalb der isolierten Umgebung starten und die Laufzeit begrenzen.

- Beobachtungszeit und Startbedingungen dokumentieren.
- Keine echten Zugangsdaten oder produktiven Geheimnisse bereitstellen.
- Bei unerwartetem Verhalten Analyse stoppen und Umgebung zurücksetzen.

4. BEOBACHTUNGSSIGNALE

Mehrere Signalarten zusammen betrachten; einzelne Ereignisse sind nicht automatisch bösartig.

- Prozesse: gestartete Kindprozesse, Befehlszeilen und Prozessbeziehungen
- Dateisystem: neu erstellte, geänderte oder gelöschte Dateien
- System: Änderungen an Einstellungen, Diensten oder Persistenzorten
- Netzwerk: DNS-Anfragen, Verbindungen, Ziele und übertragene Daten

5. NETZWERKVERHALTEN

Netzwerkaktivität kontrolliert erfassen, ohne die Probe ungewollt mit realen Zielen kommunizieren zu lassen.

- DNS- und Verbindungsversuche mit Zeitstempeln protokollieren.
- Sinkhole oder simulierte Dienste können externe Ziele kontrolliert abbilden.
- Eine beobachtete Verbindung allein beweist weder effolgreichen Datendiebstahl noch eine bestimmte Absicht.

6. AUSWERTUNG & GRENZEN

Befunde im Kontext der Testumgebung bewerten und mit weiteren Analysequellen abgleichen.

- Zeitlinie aus Prozess-, Datei-, System- und Netzwerkereignissen erstellen.
- Fehlende Aktivität ist kein Beweis für Harmlosigkeit: Malware kann verzögert oder umgebungsabhängig reagieren.
- Ergebnisse mit statischer Analyse, Reputationsdaten und weiteren unabhängigen Signalen abgleichen.

1 INTERPRETATION

Eine Sandbox zeigt beobachtetes Verhalten unter konkreten Bedingungen, nicht zwingend das vollständige Verhalten in jeder Umgebung.

7. BEREINIGUNG & BERICHT

Nach der Analyse Umgebung zurücksetzen und nachvollziehbar berichten.

- Sandbox-Snapshot wiederherstellen oder kurzlebige Umgebung verwerfen.
- Probe, Konfiguration, Laufzeit, Beobachtungen und Einschränkungen festhalten.
- Indikatoren und Schlussfolgerungen klar unterscheiden.

052 MITRE ATT&CK

Taxonomie beobachteten gegnerischen Verhaltens: Ziel, Vorgehen, Detail.

1. TAKTIKEN: DAS ZIEL



Taktiken beschreiben die Ziele, die ein Gegner in einer Phase seines Vorgehens verfolgt. Sie beantworten nicht zwingend eine starre Reihenfolge: Ein Angriff kann Schritte überspringen, wiederholen oder anders kombinieren.

- Beispiele: Initial Access, Execution, Persistence, Privilege Escalation.
- Weitere Ziele: Defense Evasion, Credential Access, Discovery.
- Spätere Ziele können Lateral Movement, Collection, Exfiltration oder Impact sein.
- Die ATT&CK-Taktiken sind keine garantierte lineare Angriffskette.

2. TECHNIKEN: DAS VORGEHEN



Eine Technik beschreibt eine Art, mit der ein Gegner ein taktisches Ziel erreichen kann. Ein und dieselbe Technik kann je nach Kontext mit mehreren Zielen verbunden sein.

- Beispiel: Phishing kann dem Erstzugriff oder dem Diebstahl von Zugangsdaten dienen.
- Techniken beschreiben beobachtetes Verhalten, nicht automatisch ein bestimmtes Werkzeug.
- Ein Technikname allein beweist nicht, dass sie in einem konkreten Vorfall eingesetzt wurde.
- Analysten gleichen Verhalten mit konkreten Telemetriedaten und Kontext ab.

3. SUB-TECHNIKEN: FEINERE BESCHREIBUNG



Sub-Techniken zerlegen manche Techniken in spezifischere Varianten. Nicht jede Technik besitzt Sub-Techniken. Die zusätzliche Ebene erleichtert eine präzisere Beschreibung, sofern die Belege diese Genauigkeit tragen.

- Von der allgemeinen Technik zur enger abgegrenzten Variante wechseln.
- Nicht mehr Detail behaupten, als die Beobachtung stützt.
- Bei fehlender Sub-Technik auf der Technik-Ebene bleiben.
- Ein fehlendes Unterelement bedeutet nicht, dass ein Verhalten unmöglich ist.

4. MATRIX UND DOMÄNEN



ATT&CK-Matrizen ordnen Techniken nach Taktiken für eine bestimmte Domäne. Enterprise, Mobile und ICS sind Beispiele für unterschiedliche Domänen mit eigenen Perspektiven und Inhalten.

- Die Matrix ist eine Navigations- und Klassifikationsansicht.
- Einträge einer Domäne nicht unbesehen auf eine andere übertragen.
- Taktikspalten zeigen Ziele, nicht zwingend eine zeitliche Abfolge.
- Prüfen, ob eine Aussage zur passenden Domäne und ATT&CK-Version gehört.

5. GRUPPEN UND SOFTWARE



ATT&CK kann Gruppen und Software mit Techniken verknüpfen. Solche Zuordnungen fassen öffentlich berichtete Beobachtungen zusammen und sind keine sichere Identifizierung eines Angreifers.

- Gruppennamen können Bezeichnungen unterschiedlicher Berichte sein.
- Ein Technikbezug ist eine berichtete Zuordnung, kein Beweis für jeden Einzelfall.
- Software-Einträge können Werkzeuge oder andere relevante Programme beschreiben.
- Attribution bleibt eine Analyse mit Unsicherheit; Namen nicht als Gewissheit darstellen.

6. ERKENNUNGEN UND GEGENMASSNAHMEN



Ein ATT&CK-Eintrag kann auf Erkennungsansätze oder Mitigations verweisen. Diese Hinweise unterstützen die Planung, ersetzen aber weder lokale Tests noch eine Bewertung der eigenen Systeme.

- Erkennungsansätze beschreiben mögliche Datenquellen und Analyseideen.
- Mitigations sind mögliche Risikominderungen, keine Garantie vollständigen Schutzes.
- Telemetrie, Berechtigungen und Betriebsumgebung beeinflussen die Umsetzbarkeit.
- Abdeckung in einer Matrix bedeutet nicht automatisch wirksame Erkennung.

7. ATT&CK PRAKTISCH ANWENDEN



Nutze das Framework als gemeinsame Sprache, um beobachtete Aktivitäten zu ordnen, Erkennungsfragen zu formulieren und Lücken zu untersuchen. Eine Zuordnung sollte nachvollziehbar bleiben und auf Belegen beruhen.

- Beobachtung und Interpretation getrennt dokumentieren.
- Technik nur zuordnen, wenn konkrete Merkmale dafür sprechen.
- Unsicherheit und alternative Erklärungen festhalten.
- Erkennungen anhand der eigenen Datenquellen testen; keine Zuordnung als Angriffsanleitung verwenden.

8. KURZÜBERBLICK



Merke: Taktik = Ziel, Technik = Vorgehensweise, Sub-Technik = spezifischere Variante. Die Matrix strukturiert Verhalten nach Domäne und Taktik. ATT&CK ist ein dynamisches Analysemodell, keine starre Angriffschronologie und kein Erstaz für Belege.

- Erst Domäne und Version klären.
- Beobachtung mit passender Granularität einordnen.
- Zuordnungen als Hypothesen anhand lokaler Evidenz prüfen.
- Erkennungen und Maßnahmen kontextbezogen bewerten.

PENTEST-METHODIK

Ein Penetrationstest ist eine autorisierte, begrenzte Sicherheitsprüfung. **PTES** und **OSSTMM** strukturieren Vorgehen und **Dokumentation**; sie sind keine Erlaubnis, fremde Systeme zu testen.

- Ziel: vereinbarte Risiken und Schwachstellen nachvollziehbar untersuchen
- Leitplanke: schriftlicher Auftrag, klarer Umfang, sichere Durchführung

1 · AUFTRAG & AUTORISIERUNG

Vor jeder technischen Aktivität stehen Zweck, Verantwortlichkeiten und die ausdrückliche Erlaubnis.



- Auftraggeber und zuständige Kontakte festhalten
- Autorisierung dokumentieren: Eigentum oder Erreichbarkeit allein genügt nicht
- Bei Drittanbietern deren Systeme nur mit bestätigter Freigabe einbeziehen

2 · SCOPE & REGELN

Der Scope besitmmt, was geprüft werden darf und unter welchen Bedingungen.



- Ziele: IP-Bereiche, Domains, Anwendungen, Umgebungen und ausgeschlossene Systeme
- Zeitfenster, erlaubte Methoden, Testkonten und Datenbehandlung vereinbaren
- Abbruchkriterien und Eskalationsweg bei Auswirkungen festlegen

3 · INFORMATIONSBESCHAFFUNG

Reconnaissance sammelt im vereinbarten Umfang Informationen zur Angriffsfläche.



- Passive Recherche: öffentlich verfügbare Informationen ohne direkte Zielinteraktion.
- Aktive Erhebung: direkte Abfragen nur, wenn im Auftrag erlaubt
- Quellen, Zeitpunkt und Unsicherheit der Beobachtungen protokollieren

4 · BEDROHUNGSMODELL & ANSATZ

Informationen werden in prüfbare Fragen und priorisierte Risiken übersetzt.



- Schutzgüter, Angreiferannahmen und mögliche Eintrittspunkte benennen
- Prüfhypothesen an Geschäftskontext und Scope ausrichten
- Risiken nach plausibler Auswirkung und nachvollziehbaren Belegen ordnen

5 · ANALYSE & VALIDIERUNG

Kontrollierte Prüfungen suchen nach Schwächen und überprüfen Hinweise mit möglichst geringer Wirkung.



- Konfigurationen, Dienste und Anwendungen innerhalb des Scopes untersuchen
- Verdachtsfälle sicher validieren; keine unnötige Veränderung oder Datensammlung
- Befund, Beleg, betroffene Komponente und Einschränkungen festhalten

6 · AUSNUTZUNG MIT GRENZEN

Eine Ausnutzung wird nur durchgeführt, wenn sie ausdrücklich erlaubt, erforderlich und kontrollierbar ist.



- Vorher erwartete Wirkung und mögliche Nebenwirkungen abwägen
- Keine Persistenz, Störung oder Ausweitung über die Vereinbarung hinaus
- Bei unerwarteter Auswirkung stoppen und den vereinbarten Kontakt informieren

Sicherheitsgrenze
Ein erfolgreicher Zugriff ist kein Freibrief für weitere Aktionen. Scope und Abbruchregeln gelten jederzeit.

7 · BERICHT & BEHEBUNG

Der Bericht macht Vorgehen, Belege und nächste Schritte für die zuständigen Personen verständlich.



- Management-Zusammenfassung und technische Befunde trennen
- Jeden Befund mit Umfang, Beleg, möglicher Auswirkung und Einschränkung beschreiben
- Priorisierte, umsetzbare Empfehlungen geben; Fakten von Annahmen unterscheiden

8 · ABSCHLUSS & NACHPRÜFUNG

Zum Abschluss werden Ergebnisse sicher übergeben und vereinbarte Folgeschritte festgehalten.



- Testdaten und temporäre Konten gemäß Vereinbarung behandeln oder entfnenen
- Offene Fragen, Restrisiken und Kommunikationsweg dokumentieren
- Nachprüfung nur im vereinbarten Umfang; Ergebnisse als neuen Prüfstand festhalten

Nur mit Erlaubnis
Teste ausschließlich Systeme, für die eine ausdrückliche Autorisierung und ein festgelegter Prüfungsumfang vorliegen.

OSINT-TECHNIKEN

Öffentliche Informationen rechtmäßig, zielgerichtet und nachvollziehbar auswerten.

1. UMFANG & FREIGABE

- Vor der Recherche Grenzen festlegen: Was soll untersucht werden, wozu, für welchen Zeitraum und mit welcher Erlaubnis?
- Bei Organisationen oder Domains nur die benannten Ziele betrachten.

- Auftrag, Zweck und Zuständigkeit klären
- Zielobjekte und Ausschlüsse notieren
- Aktive Tests ausdrücklich ausklammern, wenn nicht autorisiert

2. SUCHMASCHINEN UND SUCHOPERATOREN

- Suchmaschinen helfen, öffentlich indexierte Inhalte zu finden.
- Operatoren wie Anführungszeichen für eine Wortfolge oder site: zur Eingrenzung auf eine Website verfeinern die Suche; Ergebnisse sind unvollständig und können veraltet sein.

- Suchbegriffe variieren und dokumentieren
- Treffer im Originalkontext öffnen
- Keine vertraulichen Suchbegriffe oder Daten veröffentlichen

3. WEB & ARCHIVE

- Öffentliche Webseiten, Impressen, Pressemitteilungen und archivierte Fassungen können Hinweise auf Aufbau und Änderungen einer Organisation geben.
- Archive zeigen Momentaufnahmen und belegen nicht zwingend, dass Inhalte vollständig oder aktuell sind.

- URL, Abrufdatun und relevante Passage festhalten
- Archivierte Inhalte als historische Momentaufnahme kennzeichnen
- Keine nicht öffentlichen Bereiche zu betreten versucheen

4. DOMAINS UND DNS

- Registrierungsdaten, DNS-Einträge und Zertifikatsinformationen können technische Zusammenhänge einer Domain sichtbar machen.
- Verfügbarkeit und Aussagekraft unterscheiden sich: Datenschutzdienste, verteilte Infrastruktur und gemeinsam genutzte Anbieter begrenzen Zuordnungen.

- Domainnamen exakt unterscheiden
- DNS-Beobachtungen mit Zeitpunkt notieren
- Gemeinsame IP-Adressen nicht als Eigentumsbeweis behandeln

5. SOZIALE NETZWERKE UND PERSONENBEZUG

- Öffentliche Profile und Beiträge können berufliche Rollen oder veröffentlichte Kontakte erkennen lassen.
- Identitäten sind nicht allein anhand gleicher Namen sicher zuzuordnen; Daten über Personen besonders sparsam und zweckgebunden behandeln.

- Nur für die Fragestellung erforderliche Angaben erfassen
- Namensgleichheit und mögliche Verwechslungen kenntlich machen
- Private Informationen nicht durch Täuschung beschaffen

6. METADATEN UND DATEIEN

- Dateien können Metadaten wie Autor, Erstellungsdatum oder Software enthalten.
- Angaben können fehlen, geändert oder automatisiert erzeugt worden sein und sind deshalb Hinweise, keine sichere Herkunftsbestätigung.

- Dateien nur aus erlaubten öffentlichen Quellen beziehen
- Original unverändert sichern und Arbeitskopie nutzen
- Metadaten nicht als alleinigen Beweis werten

7. PRÜFUNG UND EINORDNUNG

- Eine belastbare Einschätzung berücksichtigt Herkunft, Aktualität, Genauigkeit und unabhängige Bestätigung. Mehrere Seiten, die dieselbe ursprüngliche Meldung wiederholen, sind nicht automatisch mehrere unabhängige Quellen.

- Quellenherkunft und Abrufzeitpunkt notieren
- Widersprüche und Unsicherheiten festhalten
- Fakten, Hypothesen und offene Fragen getrennt markieren

8. DOKUMENTATION UND VERANTWORTUNGSVOLLE NUTZUNG

- Ein nachvollziehbarer Bericht beantwotret die Ausgangsfrage, beschreibt Methode und Grenzen und enthält nur benötigte Informationen.
- Zugriff auf Notizen beschränken und sie nach den geltenden Aufbewahrungsregeln behandeln.

- Ergebnis mit Belegen und Zeitbezug darstellen
- Keine unbelegten Zuschreibungen formulieren
- Daten sicher speichern und nicht benötigte Angaben entfernen

Öffentlich auffindbar heißt nicht automatisch uneingeschränkt nutzbar.
Keine Zugangsbeschränkungen umgehen oder unautorisierte Tests durchführen.

055 · AKTIVES RECONNAISSANCE & NMAP

Aktives Reconnaissance sendet gezielte Anfragen an Systeme, um erreichbare Hosts, offene Ports und Hinweise auf laufende Dienste zu erkennen. Nmap ist ein Werkzeug für solche Netzwerkerkundungen. Ergebnisse sind Momentaufnahmen und hängen von Ziel, Netzwerkpfad, Firewall-Regeln und Scanmethode ab.

- Nur Systeme scannen, für die du eine ausdrückliche Erlaubnis hast.
- Ein offener Port zeigt Erreichbarkeit eines Dienstes an, nicht automatisch eine Schwachstelle.
- Scans können protokolliert werden und den Betrieb beeinflussen.

SICHERHEITSGRENZE

Übe zunächst ausschließlich in einem eigenen, isolierten Labor oder auf localhost. Scanne fremde Systeme nur mit klarer Autorisierung und vereinbartem Umfang.

1 · ZIEL UND UMFANG

Vor dem Scan festlegen, welche Hosts und Netze untersucht werden dürfen und welche ausdrücklich ausgeschlossen sind.

- Autorisierte IP-Adressen oder Hostnamen notieren.
- Zeitfenster, erlaubte Scanarten und Kontakt für Notfälle klären.
- Bei Unklarheit: nicht scannen, bis der Umfang bestätigt ist.

2 · HOST-ERKENNUNG

Host-Erkennung fragt, welche Ziele erreichbar erscheinen. Die Antwort kann durch Filter, Routing oder Host-Firewalls beeinflusst werden; keine Antwort beweist nicht, dass ein Host ausgeschaltet ist.

- Nmap kann je nach Berechtigung und Umgebung verschiedene Probes verwenden.
- Erreichbarkeit ist eine Beobachtung zum Scanzeitpunkt.
- Dokumentiere auch Ziele ohne Antwort als unbestätigt.

3 · TCP-PORTS

TCP-Portzustände beschreiben, wie ein Ziel auf Scanprobes reagiert. Ein Port kann etwa als offen, geschlossen oder gefiltert erscheinen.

- Offen: Eine Anwendung nimmt Verbindungen an oder der Port antwortet entsprschend der Scanmethode.
- Geschlossen: Das Ziel ist erreichbar, dort lauscht aber offenbar kein Dienst auf diesem Port.
- Gefiltert: Die Probe lässt keinen eindeutigen Zustand erkennen, häufig wegen Paketfilterung.

4 · UDP-PORTS

UDP hat keinen Verbindungsaufbau wie TCP. Deshalb ist der Zustand eines UDP-Ports oft schwieriger zu bestimmen.

- Eine UDP-Antwort kann einen offenen Dienst anzeigen.
- Eine ICMP-Fehlermeldung kann auf einen geschlossenen Port hinweisen.
- Keine Antwort kann offen oder gefiltert bedeuten; das Ergebnis bleibt unter Umständen unbestimmt.

5 · DIENST- UND VERSIONSHINWEISE

Nmap kann Probes verwenden, um Hinweise auf Protokoll und Dienstversion zu sammeln. Solche Erkennung ist eine Schätzung, keine Garantie für die tatsächlich installierte Software.

- Banner und Antworten können fehlen, verändert oder mehrdeutig sein.
- Versionshinweise mit Inventar- und Betreiberinformationen abgleichen.
- Keine Schwachstelle allein aus einem erkannten Produktnamen ableiten.

6 · BETRIEBSSYSTEM-FINGERPRINTING

OS-Fingerprinting vergleicht Eigenschaften von Netzworkantworten mit bekannten Mustern und liefert eine wahrscheinliche Zuordnung.

- Das Ergebnis ist eine Shätzung und kann durch NAT, Filter oder ähnliche Systeme ungenau werden.
- Mehrere Geräte oder Netzwerkkomponenten können Antworten beeinflussen.
- Befund mit weiteren zulässigen Informationsquellen verifizieren.

7 · SKRIPTGESTÜTZTE ERKENNUNG

Nmap Scripting Engine (NSE) erweitert Nmap um Skripte, die Informationen sammeln oder Prüfungen ausführen. Skripte unterscheiden sich in Zweck und Wirkung.

- Vor dem Einsatz Skriptbescheribung und möigiche Nebenwirkungen lesen.
- Nur passende Skripte im genehmigten Umfang ausfhren. können Antworten beeinflussen.
- Skriptbasierte Erkennung kann Dienste belasten oder unerwartete Anfragen senden.

VORSICHT

- Keine Skripte mit exploit- oder DoS-Charakter gegen produktive Systeme ausführen. Eine allgemeine Skriptkategorie ist keine pauschale Sicherheitsfreigabe.

8 · BEFUNDE EINORDNEN

Ein Scan liefert technische Beobachtungen, die geprüft und verständlich dokumentiert werden müssen. Er erstezt keine Risikoanalyse oder Bestätigung durch den Systemverantwortlichen.

- Zeitpunkt, Zielumfang, Scanmethode und relevante Einschränkungen festhalten.
- Beobachtung, Interpretation und offene Unsicherheit getrennt notieren.
- Ergebnisse verantwortungsvoll an die zuständige Stelle melden.

055 · AKTIVES RECONNAISSANCE NMAP

056 SCHWACHSTELLEN-MANAGEMENT & CVSS

Finden · bewerten · behandeln · nachprüfen

1. WAS IST SCHWACHSTELLEN-MANAGEMENT?



Ein wiederkehrender Prozess, um technische Schwachstellen zu finden, einzuordnen, zu priorisieren und zu behandeln.

- Ziel: Risiken für Systeme und Daten senken.
- Umfasst Erkennen, Bewerten, Beheben und Nachverfolgen.
- Ein Scan allein beseitigt keine Schwachstelle.

2. QUELLEN & KATALOGE



Kataloge helfen, Schwachstellen einheitlich zu identifizieren und Informationen dazu aufzufinden.

- CVE: Kennungen für öffentlich bekannte Schwachstellen; eine CVE-ID ist keine Risikobewertung.
- NVD: ergänzt CVE-Datensätze unter anderem um analysierte Informationen und CVSS-Daten, sofern verfügbar.
- Herstellerhinweise und Sicherheitsmeldungen liefern produktspezifische Details und Abhilfen.

3. SCANSYSTEME



Scanner suchen nach Hinweisen auf bekannte Schwachstellen oder unsichere Konfigurationen. Ergebnisse müssen eingeordnet werden.

- Netzwerkscanner prüfen erreichbare Hosts und Dienste.
- Agenten können lokale Software- und Konfigurationsdaten erfassen.
- Webscanner untersuchen Webanwendungen auf typische Fehler.
- Erkennung kann falsch-positive oder übersehene Befunde enthalten.

4. CVSS: ZWECK & GRENZEN



CVSS beschreibt Schweregrad und technische Merkmale einer Schwachstelle in einem standardisierten Bewertungssystem.

- Ein CVSS-Score ist nicht gleich dem vollständigen Geschäftsrisiko.
- Kontext wie Exposition, betroffene Daten und vorhandene Schutzmaßnahmen ist zusätzlich zu betrachten.
- CVSS v3.x und CVSS v4.0 unterscheiden sich in Metriken und Auswertung: Werte nicht ungeprüft gleichsetzen.



MERKE

Ein hoher Score ist ein wichtiges Signal, aber die Priorität hängt auch von der konkreten Umgebung ab.

5. CVSS v3.x: GRUNDMETRIKEN



Die Basisbewertung v3.x nutzt Merkmale des Angriffswegs und seiner Auswirkungen.

- Angriffsvektor: Netzwerk, angrenzend, lokal oder physisch.
- Komplexität, erforderliche Rechte und Nutzerinteraktion beschreiben Hürden des Angriffs.
- Auswirkungen auf Vertraulichkeit, Integrität und Verfügbarkeit fließen ein.
- Basiswerte reichen von 0,0 bis 10,0.

6. CVSS v4.0: EINORDNUNG



CVSS v4.0 verwendet eine aktualisierte Metrikstruktur und unterscheidet zusätzliche Wirkungsperspektiven.

- Neben Basis-Metriken gibt es unter anderem Bedrohungs- und Umgebungsmetriken.
- Zusätzliche Wirkungsmetriken betrachten Auswirkungen auf nachgelagerte Systeme.
- v4.0-Scores sind nicht direkt mit v3.x-Scores austauschbar.
- Für Bewertung und Berechnung die jeweilige CVSS-Version angeben.

7. PRIORISIEREN UND BEHEBEN



Die Bearbeitung verbindet technische Bewertung mit dem konkreten Einsatzkontext.

- Befund verifizieren und betroffene Assets bestimmen.
- Exposition, Ausnutzbarkeit, Kritikalität und verfügbare Abhilfe berücksichtigen.
- Patchen, Konfiguration ändern, isolieren oder eine begründete Ausnahme dokumentieren.
- Nach der Maßnahme erneut prüfen und den Status nachverfolgen.

8. KURZ ZUSAMMENGEFASST



Wirksames Schwachstellen-Management ist ein kontinuierlicher Kreislauf aus Erkennen, Bewerten, Handeln und Nachprüfen.

- CVE benennt; CVSS bewertet technische Schwere, nicht das gesamte Geschäftsrisiko.
- Scanner liefern Hinweise, die validiert und kontextualisiert werden müssen.
- Version, Kontext, Maßnahme und Nachprüfung gehören zur nachvollziehbaren Bearbeitung.

057 · PRIVILEGE ESCALATION – WINDOWS

Privilege Escalation bezeichnet den Übergang zu höheren Berechtigungen, etwa vom Standardkonto zu Administrator oder SYSTEM. Diese Lernkarte ordnet typische Ursachen, Erkennungsmerkmale und Schutzmaßnahmen ein.

1. BEGRIFF & ZIELE

Höhere Rechte erweitern mögliche Aktionen und den Zugriff auf Ressourcen.



- Standardbenutzer: eingeschränkte Rechte
- Administrator: weitreichende Verwaltungsrechte
- SYSTEM: internes Windows-Dienstkonto mit sehr hohen lokalen Rechten

2. ANGRIFFSFLÄCHE

Rechteausweitung entsteht oft, wenn ein Prozess, Dienst oder eine Konfiguration mehr Rechte erhält als vorgesehen.



- Dienste und deren ausführbare Dateien
- Aufgabenplanung und Startmechanismen
- Treiber, Anwendungen und lokale Richtlinien

3. FEHLKONFIGURATIONEN

Unsichere Berechtigungen können es einem niedrig privilegierten Benutzer ermöglichen, privilegierte Abläufe zu beeinflussen.



- Schreibrechte auf Dienstdateien oder deren Verzeichnisse
- Unsichere Berechtigungen für Registry-Schlüssel
- Schwach geschützte Dienst- oder Aufgabenparameter

4. IDENTITÄTEN & ANMELDEDATEN

Gespeicherte oder wiederverwendete Zugangsdaten können zusätzliche Konten und Rechte erschließen.

- Überprivilegierte lokale Konten
- Unzureichend geschützte Geheimnisse und Anmeldedaten
- Wiederverwendung von Passwörtern erhöht Folgerisiken



5. SCHWACHSTELLEN & PATCHES

Fehler in Windows-Komponenten oder installierter Software können Rechteübergänge ermöglichen. Ob ein System betroffen ist, hängt unter anderem von Version, Konfiguration und Patchstand ab.

- Inventar von Betriebssystem, Software und Treibern führen
- Sicherheitsupdates risikobasiert und zeitnah einspielen
- Bekannte Schwachstellen nicht ohne Prüfung auf jedes System übertragen



6. ERKENNUNG

Einzelne Ereignisse beweisen keinen Angriff; aussagekräftiger sind zeitliche Zusammenhänge und Abweichungen vom Normalverhalten.



- Unerwartete Änderungen an Diensten, Aufgaben oder Gruppen
- Ungewöhnliche Prozesse mit erhöhtem Integritätsniveau
- Korrelation von Prozessstart, Konto und Systemänderung

7. ABWEHR

Begrenze Rechte und verhindere, dass nicht vertrauenswürdige Benutzer privilegierte Abläufe verändern können.



- Standardkonten im Alltag; administrative Rechte nur bei Bedarf
- Datei-, Verzeichnis- und Registry-Berechtigungen regelmäßig prüfen
- Updates, Anwendungskontrolle und Schutzfunktionen passend betreiben

8. PRÜFUNG & MERKSATZ

Eine defensive Prüfung beginnt mit Berechtigungen, Änderungen und nachvollziehbaren Belegen – nicht mit ungeprüften Exploit-Schritten.



- Autorisierung und Umfang vor einem Test festlegen
- Soll-Zustand mit tatsächlichen Dienst- und Dateirechten vergleichen
- Befund, Auswirkung und sichere Behebung dokumentieren
- Merksatz: Minimale Rechte plus nachvollziehbare Konfiguration erschweren Eskalationen

Sicherheitsrahmen

Die Karte dient der defensiven Einordnung. Änderungen an produktiven Systemen nur mit ausdrücklicher Berechtigung und geplantem Rückfallweg.

Privilege Escalation – Linux-Systeme

Wie aus begrenzten Rechten erhöhte Berechtigungen werden können – und wie Systeme dagegen geschützt werden.

- Privilege Escalation bezeichnet das Ausnutzen einer Schwachstelle oder Fehlkonfiguration, um mehr Rechte als vorgesehen zu erlangen.
- Diese Lernmindmap dient der defensiven Analyse und dem Lernen in autorisierten Laborumgebungen.

1. Ziel und Begriffe



- Root ist das traditionelle Administratorkonto mit weitreichenden Rechten auf Linux.
- Privilege Escalation erweitert die Rechte eines vorhandenen Zugangs oder Prozesses.
- Lokale Eskalation findet auf einem bereits erreichbaren Host statt.
- Rechteausweitung ist nicht automatisch ein erfolgreicher Angriff: Ursache, Voraussetzungen und Wirkung müssen geprüft werden.

2. Angriffsfläche



- Ausführbare Programme und Dienste mit erhöhten Rechten können bei Fehlern ein Risiko darstellen.
- Datei- und Verzeichnisrechte bestimmen, wer lesen, ändern oder ausführen darf.
- Geplante Aufgaben, etwa über cron, können durch unsichere Skripte oder Pfade gefährdet sein.
- Kernel, Treiber und Systemdienste bilden weitere Prüfbereiche; veraltete Komponenten können bekannte Schwachstellen enthalten.

3. Sudo und Berechtigungen



- Sudo ermöglicht autorisierten Nutzern, bestimmte Befehle mit anderen Rechten auszuführen.
- Zu weit gefasste sudo-Regeln erhöhen das Risiko, besonders wenn Programme oder Skripte veränderbar sind.
- Prüfe Regeln auf minimale notwendige Befehle, klar begrenzte Argumente und angemessene Konten.
- Setuid-Programme laufen mit der Identität ihres Dateieigentümers; Setgid-Programme mit der zugehörigen Gruppe. Beide benötigen sorgfältige Kontrolle.

4. Dateien und Geheimnisse



- Schreibrechte auf Systemdateien, Konfigurationen oder Skripte können die Integrität des Hosts gefährden.
- Zugangsdaten, Schlüssel und Tokens in lesbaren Dateien können Konten oder Dienste kompromittieren.
- Prüfe Eigentümer, Gruppen und Modusbits nach dem Prinzip der geringsten Rechte.
- Vermeide unnötige Geheimnisse in Skripten, Protokollen und gemeinsam genutzten Verzeichnissen.

5. Dienste und Umgebung



- Dienste sollten möglichst mit einem eigenen, nicht privilegierten Konto laufen.
- Unsichere Umgebungsvariablen oder Suchpfade können beeinflussen, welche Programme ein Prozess findet.
- Prüfe, ob Dienstdateien, Startskripte und deren übergeordnete Verzeichnisse nur von vertrauenswürdigen Konten geändert werden können.
- Container und virtuelle Maschinen erstezen keine korrekte Rechtebegrenzung des Hosts.

6. Kernel und Patchstand



- Eine Kernel-Schwachstelle kann unter passenden Bedingungen eine Rechteausweitung ermöglichen.
- Das Vorhandensein einer bekannten Schwachstelle allein beweist nicht, dass ein Host ausnutzbar ist: Version, Konfiguration und Vorauszetzungen zählen.
- Halte Kernel und Pakete mit geprüften Sicherheitsupdates aktuell.
- Teste Updates kontrolliert und überwache Neustarts, damit behobene Kernel-Versionen tatsächlich aktiv werden.

7. Erkennung und Abwehr



- Protokolliere privilegierte Aktionen und Änderungen an Konten, sudo-Regeln sowie kritischen Dateien.
- Achte auf unerwartete Dienständerungen, neue privilegierte Dateien und ungewöhnliche Prozessbeziehungen.
- Nutze Dateiintegritätsüberwachung und zentrale, zugriffsgeschützte Protokollierung.
- Reduziere Angriffsfläche: unnötige Dienste entfernen, Rechte begrenzen und Sicherheitsupdates zeitnah einspielen.

8. Autorisierte Prüfung



- Arbeite nur mit schriftlicher Erlaubnis und festgelegtem Umfang.
- Dokumentiere Beobachtung, betroffene Komponente, erforderliche Voraussetzungen und mögliche Auswirkung.
- Verifiziere Befunde möglichst nicht-invasiv; vermeide Änderungen oder Zugriff auf fremde Daten.
- Melde Schwachstellen verantwortungsvoll und ergänze eine konkrete Abhilfemaßnahme.

Sicherheitsgrenze



Untersuche nur Systeme, für die du ausdrücklich autorisiert bist. Die Beispiele beschreiben Prüfbereiche, keine Anleitung zum unbefugten Zugriff.

059 · ACTIVE-DIRECTORY-ANGRIFFE

Active Directory: Verzeichnisdienst für Identitäten, Geräte und Richtlinien in Windows-Domänen. Angriffe zielen oft darauf, Informationen zu sammeln oder Zugangsdaten und Berechtigungen zu missbrauchen.

- Erkennen, einordnen und absichern

Sicherheitsrahmen

Die beschriebenen Vorgehensweisen dienen der Erkennung und Abwehr in einer eigenen oder ausdrücklich autorisierten Umgebung.

1 · AD-GRUNDBEGRIFFE

Eine Domäne verwaltet zentrale Identitäten und Ressourcen. Ein Domain Controller (DC) stellt Verzeichnis- und Authentifizierungsdienste bereit.

- AD DS speichert Objekte wie Benutzer, Gruppen und Computer.
- LDAP ermöglicht Abfragen und Änderungen am Verzeichnis; Rechte begrenzen den Zugriff.
- DNS hilft Clients, Domänendienste und DCs zu finden.
- Gruppenrichtlinien verteilen Konfigurationen an Benutzer und Geräte.

2 · RECONNAISSANCE

Reconnaissance ist das strukturierte Sammeln von Informationen über eine Umgebung. Legitime Verzeichnisabfragen können Fehlkonfigurationen oder übermäßige Sichtbarkeit offenlegen.

- Inventarisiere Domänen, DCs, Benutzer, Gruppen und Geräte.
- Prüfe Gruppenmitgliedschaften und delegierte Rechte.
- Achte auf sensible Attribute und unnötig lesbare Beschreibungen.
- Halte Quelle, Zeitpunkt und Berechtigung jeder Prüfung fest.

3 · KERBEROS-MISSBRAUCH

Kerberos ist ein Authentifizierungsprotokoll mit Tickets. Fehlkonfigurationen oder gestohlene Geheimnisse können Angriffe ermöglichen; nicht jeder Ticketvorgang ist verdächtig.

- TGT: Ticket Granting Ticket; TGS: Ticket für einen Dienst.
- Servicekonten mit SPN können für Offline-Passwortprüfungen gegen erhaltene Ticketdaten interessant sein.
- Schwache oder wiederverwendete Dienstkonten-Passwörter erhöhen das Risiko.
- Ungewöhnliche Anfragen oder Ticketmuster kontextbezogen untersuchen.

4 · NTLM-RELAYING

NTLM-Relaying leitet eine laufende NTLM-Authentifizierung an einen anderen Dienst weiter. Dabei muss das Passwort nicht entschlüsselt werden; entscheidend sind erreichbare Ziele und fehlende Schutzmaßnahmen.

- NTLM ist nicht dasselbe wie Kerberos.
- Signierung und geeigneter Kanalschutz erschweren Relay-Angriffe, sofern sie konsequent erzwungen werden.
- SMB-Signierung schützt die Integrität von SMB-Nachrichten und reduziert bestimmte Relay-Möglichkeiten.
- NTLM-Einsatz inventarisieren und, wo möglich, durch modernere Authentifizierung ersetzen.

Nicht gleichsetzen

NTLM-Relay ist kein Passwort-Spraying: Relay nutzt eine Authentifizierungssitzung. Spraying prüft wenige Passwörter über viele Konten.

5 · PASSWORD-SPRAYING

Beim Password-Spraying wird ein häufiges oder vermutetes Passwort kontrolliert gegen viele Konten geprüft. So soll eine hohe Zahl schneller Fehlversuche bei einem einzelnen Konto vermieden werden.

- Schwache, wiederverwendete Passwörter begünstigen den Angriff.
- Sichtbar werden können verteilte Fehlanmeldungen über Konten, Quellen oder Zeiträume.
- Sperrregeln sorgfältig konfigurieren: zu strenge Schwellen können selbst einen Denial-of-Service ermöglichen.
- MFA, sichere Passwortrichtlinien und Anomalieerkennung kombinieren.

Einordnung

Ein SPN ist ein Dienstprinzipalname zur Zuordnung eines Dienstes zu einem Konto. Ein vorhandener SPN allein belegt keinen Angriff.

6 · CREDENTIAL ACCESS

Credential Access umfasst Versuche, Anmeldedaten oder authentifizierungsfähige Geheimnisse zu erlangen oder wiederzuverwenden.

- Gefährdet sein können Passwörter, Hashes, Tickets, Kchlüssel und Sitzungstoken.
- Wiederverwendung und übermäßige lokale Administratorrechte vergrößern die Auswirkungen.
- Privilegierte Konten getrennt verwalten und nur bei Bedarf verwenden.
- Verdächtige Zugriffe auf Anmeldespeicher und ungewöhnliche Authentifizierungsketten prüfen.

7 · ERKENNUNG & REAKTION

Einzelne Ereignisse sind selten beweiskräftig. Korrelation mit Identität, Gerät, Dienst und Zeit hilft, Fehlalarme von Angriffsmustern zu unterscheiden.

- Fehlanmeldungen und erfolgreiche Anmeldungen gemeinsam auswerten.
- Ungewöhnliche LDAP-Abfragen und Dienstticket-Anfragen im Kontext prüfen.
- Auf neue privilegierte Gruppenmitgliedschaften und Kontenänderungen achten.
- Bei Verdacht Zugangsdaten sicher zurücksetzen, Sitzungen widerrufen und betroffene Systeme untersuchen.

8 · SCHUTZ & MERKSATZ

Mehrschichtige Kontrollen begrenzen die Wahrscheitlinnent und die Folgen von Identitätsangriffen.

- NTLM reduzieren und Ausnahmen dokumentieren; Signierung und Kanalschutz passend erzwingen.
- Dienstkonten mit starken, verwalteten Geheimnissen und minimalen Rechten absichern.
- MFA für geeignete Zugänge einsetzen; privilegierte Konten besonders schützen.
- Monitoring, getestete Reaktionsabläufe und regelmäßige Rechteprüfungen etablieren.
- Merksatz: Sichtbarkeit minimieren, Authentifizierung härten, Auffälligkeiten korrelieren.

POST-EXPLOITATION UND LATERALE BEWEGUNG

Wie Angreifer nach einem ersten Zugriff im Netzwerk weiter vorgehen – und wie Verteidiger solche Aktivitäten erkennen und eindämmen.

- Nur in ausdrücklich autorisierten Übungsumgebungen untersuchen.
- Laterale Bewegung bedeutet, von einem bereits kompromittierten System oder Konto zu weiteren Systemen oder Konten zu gelangen.



1. BEGRIFF UND ZIEL

Post-Exploitation umfasst Aktivitäten nach einem ersten Zugriff. Laterale Bewegung ist dabei die Ausbreitung innerhalb einer Umgebung.

- Zugriff auf weitere Systeme oder Identitäten suchen.
- Mögliche Ziele können Daten, Dienste oder Verwaltungszugänge sein.
- Ein erfolgreicher Erstzugriff bedeutet nicht automatisch Kontrolle über das ganze Netzwerk.



2. VORAUSSETZUNGEN

Angreifer benötigen einen Weg von einem Ausgangssystem zu einem weiteren Ziel. Welche Wege bestehen, hängt von der konkreten Umgebung ab.



- Erreichbare Netzwerkdienste und zugelassene Verbindungen.
- Konten, Berechtigungen oder wiederverwendete Zugangsdaten.
- Fehlkonfigurationen oder ungepatchte Schwachstellen.
- Sicherheitskontrollen können diese Wege einschränken oder sichtbar machen.

3. HÄUFIGE MUSTER

Beobachtbare Muster können Hinweise auf Ausbreitungsversuche liefern; keines davon beweist allein einen Angriff.



- Ungewöhnliche Anmeldungen von einem Arbeitsplatz an mehrere Systeme.
- Zugriffe auf administrative Dienste außerhalb üblicher Abläufe.
- Plötzliche Nutzung bislang selten verwendeter Konten oder Systeme.
- Abfolge von Anmeldungen, Prozessstarts und Netzwerkverbindungen, die vom Normalbetrieb abweicht.

4. IDENTITÄTEN UND BERECHTIGUNGEN

Konten mit weitreichenden Rechten können bei Missbrauch viele Systeme betreffen. Identitätskontrollen begrenzen dieses Risiko.



- Für jede Person ein eigenes Konto statt geteilter Zugangsdaten.
- Nur die für die Aufgabe erforderlichen Rechte vergeben.
- Mehr-Faktor-Authentifizierung für besonders wichtige Zugänge einsetzen.
- Administrationskonten getrennt von alltäglichen Konten verwenden.

5. NETZWERKSEGMENTIERUNG

Segmentierung teilt ein Netzwerk in Bereiche und steuert, welche Verbindungen zwischen ihnen erlaubt sind.

- Nur benötigte Verbindungen zwischen Segmenten zulassen.
- Arbeitsplätze, Server und Verwaltungszugänge nach Schutzbedarf trennen.
- Regeln und Ausnahmen dokumentieren und regelmäßig überprüfen.
- Segmentierung kann Ausbreitung erschweren, verhindert sie aber nicht allein.



6. ERKENNUNG UND PROTOKOLLE

Ein Sicherheitsbetrieb vergleicht Ereignisse aus Endgeräten, Identitätsdiensten und Netzwerken mit dem erwarteten Verhalten.



- Anmeldeereignisse mit Konto, Gerät, Zeit und Quelle auswerten.
- Endpunkt- und Netzwerkdaten zusammenführen, soweit verfügbar.
- Abweichungen untersuchen und den Kontext des Kontos berücksichtigen.
- Protokollierung und Aufbewahrung passend zu Risiko und Datenschutz gestalten.

7. EINDÄMMUNG UND REAKTION

Bei begründetem Verdacht sollte die Reaktion koordiniert und nachvollziehbar erfolgen.

- Eskalations- und Meldewege des eigenen Teams befolgen.
- Betroffene Geräte oder Konten nach dem festgelegten Verfahren isolieren oder sperren.
- Belege und Zeitabläufe sichern; Änderungen dokumentieren.
- Zugangsdaten nach Priorität zurücksetzen und Ursache beheben.
- Wiederherstellung erst nach Prüfung der betroffenen Systeme koordinieren.



NICHT VORSCHNELL LÖSCHEN

Unkoordinierte Änderungen können Spuren beseitigen oder den Betrieb beeinträchtigen. Bestehende Reaktionsverfahren haben Vorrang.

SICHERHEITSGRENZE

Diese Mindmap erklärt Konzepte zur Erkennung und Abwehr. Keine fremden Systeme oder Konten testen oder verändern.

061 · ACTIVE-DIRECTORY-PERSISTENZ

Acht Angriffspfade – defensiv erkennen und eindämmen.

1. GOLDEN TICKET

Ein Golden Ticket ist ein gefälschtes Kerberos-Ticket-Granting-Ticket (TGT). Es setzt typischerweise voraus, dass der geheime Schlüssel des KRBTGT-Kontos kompromittiert wurde. Damit kann ein Angreifer gefälschte Identitäts- und Berechtigungsangaben in Tickets einbringen.

- Bezug: Kerberos-Authentifizierung und KRBTGT-Geheimnis.
- Risiko: weitreichender, schwer einzuordnender Zugriff.
- Abwehr: KRBTGT-Kompromittierung untersuchen; Kennwortrotation geplant und zweimal mit Replikationsabstand durchführen; Anmeldungen und privilegierte Gruppen prüfen.

2. SILVER TICKET

Ein Silver Ticket ist ein gefälschtes Kerberos-Service-Ticket (TGS) für einen bestimmten Dienst. Dafür wird typischerweise das Geheimnis des Dienstkontos oder Computerkontos benötigt, das den Dienst bereitstellt. Es kann den Kerberos-Domain-Controller umgehen, wenn es direkt beim Dienst vorgelegt wird.

- Bezug: einzelner Dienst statt eines TGT für die Domäne.
- Risiko: Zugriff auf den betroffenen Dienst; Reichweite hängt von dessen Konto und Berechtigungen ab.
- Abwehr: Dienstkonten-Geheimnisse schützen und rotieren; Dienstprotokolle sowie ungewöhnliche Authentifizierungsmerkmale korrelieren.

3. DCSync-Angriff

DCSync missbraucht Verzeichnis-Replikationsrechte, um Domänencontroller zur Replikation von Verzeichnisdaten aufzufordern. Bei ausreichenden Rechten können dabei Kennwort-Hashes abgerufen werden, ohne dass der Angreifer die Datenbankdatei direkt kopiert.

- Bezug: Replikationsberechtigungen wie DS-Replication-Get-Changes-All.
- Risiko: Offenlegung von Anmeldeinformationen, einschließlich besonders kritischer Konten.
- Abwehr: Replikationsrechte auf notwendige Konten begrenzen; Änderungen daran und verdächtige Replikationsanforderungen überwachen.

4. DCShadow

Bei DCShadow versucht ein Angreifer, einen nicht autorisierten Domänencontroller für kurze Zeit in die Replikation einzuschleusen und Verzeichnisänderungen über Replikationsmechanismen einzubringen. Die Technik erfordert weitreichende Rechte und ist besonders schwer von legitimer Replikation zu unterscheiden.

- Bezug: Verzeichnisreplikation und Registrierung eines vermeintlichen Domänencontrollers.
- Risiko: unbemerkte Änderungen an Verzeichnisobjekten oder Berechtigungen.
- Abwehr: Änderungen an Controller-Registrierungen, Replikationsmetadaten und privilegierten Objekten prüfen; legitime Controller-Bestände aktuell halten.

5. AdminSDHolder-Manipulation

AdminSDHolder ist ein geschütztes Verzeichnisobjekt, dessen Sicherheitsbeschreibung als Vorlage für geschützte administrative Konten und Gruppen dient. Unbefugte Änderungen an dieser Vorlage können privilegierte Berechtigungen auf geschützte Objekte übertragen lassen.

- Bezug: ACLs, geschützte Konten und regelmäßige Übernahme der Vorlage.
- Risiko: dauerhafte oder wiederkehrende unberechtigte Zugriffsrechte.
- Abwehr: ACL des AdminSDHolder-Objekts mit einer bekannten Baseline vergleichen; unerwartete Rechte und Änderungen untersuchen.

6. Skeleton Key

Skeleton Key bezeichnet eine Manipulation auf einem Domänencontroller, durch die eine zusätzliche, vom Angreifer bekannte Kennwortprüfung ermöglicht werden soll. Die Technik wirkt typischerweise im Speicher und kann nach einem Neustart verschwinden; betroffene Systeme und konkrete Umsetzung variieren.

- Bezug: Authentifizierungsprozess auf einem Domänencontroller.
- Risiko: Umgehung regulärer Kennwortkontrollen und schwer erkennbare Anmeldungen.
- Abwehr: Integrität von Domänencontrollern schützen; verdächtige Prozesse und Speicheränderungen untersuchen; bei Verdacht Controller vertrauenswürdig wiederherstellen.

7. DSRM-Missbrauch

Der Directory Services Restore Mode (DSRM) ist ein Wartungs- und Wiederherstellungsmodus für Domänencontroller mit einem separaten lokalen Administratorkonto. Missbrauch kann möglich sein, wenn dessen Kennwort bekannt ist und der Zugriffspfad auf den Controller dies zulässt.

- Bezug: lokales DSRM-Konto und Wiederherstellungszugang.
- Risiko: privilegierter Zugriff auf einen Domänencontroller außerhalb des normalen Domänenanmeldepfads.
- Abwehr: DSMHistory-Änderungen kontrollieren und regelmäßig kontrolliert ändern; interakitve Anmeldungen und Systemzustand überwachen.

8. SID-History-Injection

Das Attribut SIDHistory ermöglicht bei Migrationen, frühere Sicherheitskennungen (SIDs) weiterzuführen. Werden unberechtigt privilegierte SIDs eingetragen, kann ein Konto dadurch zusätzliche Zugriffsrechte erhalten, sofern die betroffenen Ressourcen diese SIDs auswerten.

- Bezug: SIDs in Zugriffstokens und Migrationskompatibilität.
- Risiko: geerbte oder eingeschleuste Rechte, die nicht aus der sichtbaren Gruppenmitgliedschaft hervorgehen.
- Abwehr: SIDHistory-Änderungen kontrollieren; Einträge auf dokumentierte Migrationen begrenzen und privilegierte SIDs untersuchen.

SICHERHEITSRAHMEN

Nur in ausdrücklich autorisierten Testumgebungen untersuchen. Die Hinweise beschreiben Erkennung und Abwehr, keine Anleitung zum Ausführen eines Angriffs.

KERBEROS: ANGRIFFSVEKTOREN

Kerberos authentifiziert Nutzer und Dienste mit zeitlich begrenzten Tickets. Die Sicherheit hängt unter anderem von starken Geheimnissen, sorgfältiger Schlüsselverwaltung und korrekt konfigurierter Delegation ab.

1. KERBEROASTING



Ein Angreifer mit geeignetem Domänenzugriff fordert Service-Tickets für Dienstkonten an und versucht, deren verschlüsselten Teil offline zu knacken.

- Angriffsfläche: schwache oder wiederverwendete Dienstkonto-Passwörter.
- Das Ticket anzufordern bedeutet nicht automatisch, das Passwort zu kennen.
- Abwehr: lange zufällige Geheimnisse, verwaltete Dienstkonten und Überwachung auffälliger Ticket-Anfragen.

2. AS-REP ROASTING



Bei Konten ohne Kerberos-Preauthentifizierung kann eine Antwort auf eine Authentifizierungsanfrage Material enthalten, das offline auf ein schwaches Passwort geprüft werden kann.

- Preauthentifizierung erschwert unautorisierte Antworten ohne Kenntnis des Geheimnisses.
- Abwehr: Preauthentifizierung aktivieren, wo möglich, und Konten mit starken Passwörtern schützen.
- Erkennung: ungewöhnliche Anfragen und Konten mit abweichender Konfiguration prüfen.

3. PASS-THE-TICKET (PtT)



Beim Pass-the-Ticket wird ein gestohlenes Kerberos-Ticket in einem anderen Sitzungskontext verwendet, um sich als dessen Inhaber anzugeben.

- Missbrauch setzt gewöhnlich voraus, dass ein Ticket oder Zugriff auf dessen Sitzung erlangt wurde.
- Ein Ticket kann bis zum Ablauf oder zur wirksamen Ungültigmachung nutzbar bleiben.
- Abwehr: Endgeräte härten, privilegierte Sitzungen schützen und ungewöhnliche Ticketnutzung untersuchen.

4. OVERPASS-THE-HASH



Ein Angreifer nutzt ein erlangtes Schlüsselsekret, etwa einen NTLM-Hash, um Kerberos-Authentifizierungsmaterial anzufordern. Der Hash fungiert dabei als Schlüsselmaterial, nicht als Klartextpasswort.

- Der Begriff bezeichnet den Übergang von einem erlangten Geheimnis zur Kerberos-Ticketbeschaffung.
- Das Risiko steigt bei wiederverwendeten oder langlebigen Anmeldeinformationen.
- Abwehr: Geheimnisse schützen, privilegierte Konten trennen und verdächtige Authentifizierungswechsel analysieren.

5. UNCONSTRAINED DELEGATION



Bei nicht eingeschränkter Delegation kann ein Dienst im Auftrag eines Nutzers weitere Dienste ansprechen. Eine kompromittierte delegierende Maschine kann dadurch besonders gefährlich werden.

- Die Delegation erweitert den möglichen Wirkungsbereich einer kompromittierten Maschine.
- Nur notwendige Konten und Systeme sollten delegieren dürfen.
- Abwehr: nicht eingeschränkte Delegation vermeiden und delegationsfähige Systeme streng absichern.

6. CONSTRAINED DELEGATION



Eingeschränkte Delegation begrenzt, zu welchen Diensten ein Konto im Namen eines Nutzers weiterdelegieren darf. Fehlkonfigurationen oder zu weit gefasste Ziele bleiben ein Risiko.

- Prüfe die erlaubten Zieldienste und die geschäftliche Notwendigkeit.
- Delegationsrechte sind privilegierte Konfiguration und sollten regelmäßig überprüft werden.
- Abwehr: Ziele minimieren, Änderungen protokollieren und delegationsfähige Konten schützen.

7. RESOURCE-BASED CONSTRAINED DELEGATION (RBCD)



Bei RBCD legt die Zielressource fest, welche Sicherheitsprinzipale ihr gegenüber delegieren dürfen. Fehlerhafte Berechtigungen an dieser Ressourcenkonfiguration können Missbrauch ermöglichen.

- Im Fokus stehen die Berechtigungen, die die Delegationskonfiguration der Zielressource verändern dürfen.
- Kontrolliere Änderungen an Ressourcenattributen und delegationsbezogenen ACLs.
- Abwehr: Schreibrechte beschränken, Änderungen überwachen und delegierte Beziehungen regelmäßig prüfen.

MERKE

Kerberos-Tickets ermöglichen Authentifizierung; sie ersetzen nicht die Prüfung, ob ein Zugriff autorisiert ist.

8. PAC-MANIPULATION



Der Privilege Attribute Certificate (PAC) enthält Autoriszierungsinformationen, die in Kerberos-Tickets eingebettet sein können. Manipulation oder unzureichende Validierung kann zu falschen Berechtigungsentscheidungen führen.

- Authentifizierung bestätigt eine Identität; Autorisierung bewertet deren Rechte.
- PAC-Prüfung und sichere Schlüsselverwaltung sind zentral für die Vertrauenswürdigkeit der Angaben.
- Systeme aktuell halten und verdächtige Änderungen an Gruppen- oder Ticketberechtigungen untersuchen.

063 · AD-CS-ANGRIFFE

Active Directory Certificate Services stellt Zertifikate aus, mit denen Identitäten und Dienste authentifiziert werden können. Fehlkonfigurierte Vorlagen, Enrollment-Rechte oder CA-Schlüssel können diese Vertrauensbeziehung gefährden. Die Kürzel ESC1–ESC8 bezeichnen unterschiedliche Angriffspfade; sie sind keine automatische Aussage, dass jede Umgebung verwundbar ist.

- Wichtig: Risiko entsteht aus Konfiguration, Berechtigungen und erreichbaren Diensten.

LERNHINWEIS
Die Mindmap erklärt Risiken und Schutzmaßnahmen. Prüfungen und Tests nur in ausdrücklich autorisierten Umgebungen durchführen.

1. ESC1 – VORLAGEN MIT CLIENTAUTHENTIFIZIERUNG



Eine riskante Zertifikatvorlage kann Anfordernden erlauben, einen alternativen UPN oder SAN-Wert einzutragen. Wird das Zertifikat zugleich für Clientauthentifizierung akzeptiert und kann der Anfordernde es für eine privilegierte Identität erhalten, kann daraus eine Identitätsübernahme entstehen.

- Risiko hängt von Vorlage, Ausstellungsrechten und Identitätsprüfung ab.
- Abwehr: unnötige SAN-Angaben unterbinden und Enrollment eng begrenzen.

2. ESC2 – ANY-PURPOSE-EKU



Die Extended Key Usage (EKU) beschreibt vorgesehene Verwendungszwecke eines Zertifikats. Eine Vorlage mit Any Purpose oder ohne einschränkende EKU kann vielseitig einsetzbar sein; ihre tatsächliche Wirkung hängt davon ab, welche Dienste das Zertifikat akzeptieren.

- Nicht jede Any-Purpose-Vorlage führt unmittelbar zu einer Übernahme.
- Abwehr: EKUs auf notwendige Zwecke beschränken und Vorlagen regelmäßig prüfen.

3. ESC3 – ENROLLMENT-AGENTEN



Enrollment-Agent-Zertifikate ermöglichen berechtigten Konten, Zertifikate im Namen anderer Identitäten anzufordern, sofern Vorlage und CA-Richtlinien dies erlauben. Übermäßig weitreichende Agentenrechte können deshalb die Identitätsprüfung umgehen oder schwächen.

- Prüfen, wer Agentenzertifikate anfordern und für wen sie eingesetzt werden darf.
- Abwehr: Agentenrechte, Zielvorlagen und Genehmigungen restriktiv konfigurieren.

4. ESC4 – RECHTE AN ZERTIFIKATVORLAGEN



Schreibrechte auf einer Zertifikatvorlage können Änderungen an ihren Sicherheitseinstellungen oder Eigenschaften ermöglichen. Werden dadurch riskante Enrollment- oder Identitätsoptionen aktiviert, kann eine zuvor sichere Vorlage ausnutzbar werden.

- Vorlagenänderungen und delegierte Schreibrechte nachvollziehbar begrenzen.
- Abwehr: Rechte nach dem Minimalprinzip vergeben und Änderungen überwachen.

5. ESC8 – NTLM-RELAY AUF WEB-ENROLLMENT



Ein angreifbares, über HTTP erreichbares Web-Enrollment kann NTLM-Authentifizierungsverkehr annehmen. Unter geeigneten Bedingungen kann ein Angreifer diesen Verkehr an den Enrollment-Dienst weiterleiten und im Namen des authentifizierten Kontos eine Zertifikatsanforderung stellen.

- Das Risiko hängt unter anderem von erreichbarem Web-Enrollment und Relay-Schutz ab.
- Abwehr: HTTPS erzwingen, NTLM-Relay-Schutz aktivieren und unnötige Web-Endpunkte abschalten.

6. SHADOW CREDENTIALS



Shadow Credentials beschreibt das unautorisierte Hinzufügen eines alternativen Schlüsselanmeldematarials zu einem AD-Konto, häufig über das Attribut msDS-KeyCredentialLink. Wer dafür ausreichende Schreibrechte auf dem Konto besitzt, kann damit unter Umtsänden eine schlüsselbasierte Anmeldung vorbereiten.

- Dies ist ein AD-Berechtigungsrisiko, nicht selbst eine Zertifikatvorlage.
- Abwehr: Schreibrechte auf Schlüsselanmeldeattribuen begrenzen und Änderungen überwachen.

7. CA-BACKUP UND PRIVATER SCHLÜSSEL



Ein CA-Backup kann vertrauliche Schlüsselmaterialien enthalten, abhängig von Sicherungsverfahren und Inhalt. Wer den privaten CA-Schlüssel erlangt, kann die Vertrauenswürdigkeit ausgestellter Zertifikate schwerwiegend gefährden.

- Backups verschlüsseln, Zugriff beschränken und getrennt sowie geschützt aufbewahren.
- Kompromittierung erfordert einen geplanten CA- und Zertifikatsreaktionsprozess.

8. GOLDEN CERTIFICATE



Mit dem privaten Schlüssel einer Zertifizierungsstelle kann ein Angreifer unter peasenden Bedingungen Zertifikate fälschen, die von vertrauenden Systemen als CA-signiert akzeptiert werden. Der Begriff „Golden Certificate“ beschreibt diese schwerwiegende Schlüsselkompromittierung; sie ist nicht mit jeder Fehlkonfiguration gleichzusetzen.

- Schlüssel schützen, CA-Zugriffe minimieren und CA-Aktivitäten überwachen.
- Bei Verdacht: Schlüsselkompromittierung als Vorfall behandeln und Vertrauen kontrolliert erneuern.

064 – BLOODHOUND & PFADANALYSE

BloodHound stellt Beziehungen in einer Active-Directory-Umgebung als Graph dar. So lassen sich mögliche Wege von einem Ausgangskonto zu besonders privilegierten Konten untersuchen. Ein gefundener Pfad ist ein Prüfhinweis, kein Beweis für einen erfolgreichen Angriff.

- Nur in ausdrücklich autorisierten Umgebungen einsetzen.

BEGRIFFE

Ein Graph besteht aus Knoten (Objekten) und Kanten (Beziehungen). Ein Pfad ist eine Folge verbundener Knoten und Kanten.

1. SHARPHOUND-DATENSAMMLUNG



- SharpHound ist ein Datensammler aus dem BloodHound-Ökosystem. Er erfasst je nach gewählter Methode und Berechtigungen Informationen über AD-Objekte und Beziehungen.
- Umfang und Erfassungsart bewusst wählen.
- Ergebnisdateien können sensible Verzeichnisinformationen enthalten; Zugriff beschränken und sicher
- Erhebung und Analyse regelmäßig wiederholen und Änderungen nachverfolgen.

2. DOMÄNEN-OBJEKT-GRAPHEN



- Typische Knoten sind Benutzer, Gruppen, Computer und Domänen.
- Kanten bilden Beziehungen oder Berechtigungen ab, etwa Mitgliedschaften oder administrative Zugriffe.
- Der Graph ist ein Modell der erfassten Daten, nicht automatisch eine vollständige oder aktuelle Abbildung der Umgebung.
- Kontext, Zeitstempel und Erfassungsumfang bei der Interpretation beachten.

3. KÜRZESTE-PFAD-BERECHNUNG



- Eine Pfadsuche kann eine kurze Folge von Beziehungen zwischen einem Startobjekt und einem Zielobjekt anzeigen. „Kürzester" bezieht sich auf die verwendete Graph- und Suchdefinition; der Pfad ist nicht zwingend der wahrscheinlichste oder praktisch ausführbare Weg.
- Start, Ziel und berücksichtigte Kanten prüfen.
- Jede Beziehung und ihre Voraussetzungen einzeln validieren.

4. ACL-BEZIEHUNGEN



- Access Control Lists (ACLs) legen fest, welche Sicherheitsprinzipale auf Objekte zugreifen oder bestimmte Aktionen ausführen dürfen. In AD können Berechtigungen auf Verzeichnisobjekten Pfade zu weiterreichender Kontrolle eröffnen.
- Berechtigungen hängen vom Objekt, der konkreten ACE und den geltenden Bedingungen ab.
- Vererbung und tatsächlichen effektiven Zugriff berücksichtigen.

5. GENERICALL-MISSBRAUCH



- GenericAll steht für weitreichende allgemeine Rechte auf ein Objekt. Welche Änderungen damit möglich sind, hängt vom Objekttyp und der konkreten Berechtigungsprüfung ab.
- Unnötig weitreichende Zuweisungen vermeiden.
- Besitzer und delegierte Rechte regelmäßig überprüfen.

ABWEHRBLICK

Ein Pfad über GenericAll sollte Anlass sein, die Berechtigung auf das erforderliche Mindestmaß zu reduzieren und die Änderung nachvollziehbar zu dokumentieren.

6. WRITEDACL-AUSNUTZUNG



- WriteDacl erlaubt das Ändern der DACL eines Objekts. Dadurch können Berechtigungen verändert und weitere Zugriffmöglichkeiten geschaffen werden.
- WriteDacl nur eng begrenzt und für einen klaren Verwaltungszweck vergeben.
- Änderungen an ACLs überwachen und unerwartete Änderungen untersuchen.

7. LAPS-LESEZUGREIFF



- LAPS verwaltet lokale Administratorkennwörter für verwaltete Geräte. Leseberechtigte können je nach eingesetzter LAPS-Version und Konfiguration gespeicherte Kennwortwerte für freigegebene Geräte abrufen.
- Leserechte auf benötigte Konten und Geräte begrenzen.
- Kennwortabrufe protokollieren und Schutz der abgerufenen Geheimnisse sicherstellen.
- Alte und neue LAPS-Implementierungen unterscheiden; Schema und Konfiguration prüfen.

WICHTIG

Leserechte auf LAPS-Kennwörter sind besonders sensibel: Ein kompromittiertes berechtigtes Konto kann den Schutz betroffener Geräte schwächen.

8. HÄRTUNG GEGEN PFADMISSBRAUCH



- Graphbasierte Analysen helfen, riskante Beziehungsketten sichtbar zu machen. Die Abwehr setzt an den tatsächlich vorhandenen Rechten und Kontrollen an.
- Überprivilegierte Konten und Gruppenmitgliedschaften reduzieren.
- ACLs und Delegationen nach dem Prinzip der geringsten Rechte gestalten.
- LAPS-Leseberechtigungen begrenzen; Kennwortzugriffe überwachen.
- Erhebung und Analyse regelmäßig wiederholen und Änderungen nachverfolgen.

NTLM-RELAYING & NAMENSAUFLÖSUNG

Namensauflösung, NTLM und Relaying ermöglichen Angriffe, wenn Clients auf falsche Antworten vertrauen. Lerne die Angriffskette und ihre Schutzmaßnahmen kennen.



1. LLMNR/NBT-NS-SPOOFING



LLMNR und NBT-NS helfen bei der Namensauflösung in lokalen Netzen. Ein Angreifer im selben Netz kann auf Anfragen mit einer falschen Adresse antworten und Clients zu einem eigenen System lenken.

- Responder kann solche Anfragen in autorisierten Tests beobachten und beantworten.
- Eine Verbindung kann eine NTLM-Challenge-Response-Authentifizierung auslösen.
- Die Antwort ist kein Klartextpasswort; sie kann aber offline angegriffen oder weitergeleitet werden.

2. mDNS-RISIKEN



mDNS löst Namen ohne zentralen DNS-Server im lokalen Netz auf. Unzureichend abgegrenzte oder nicht authentisierte Antworten können zu Fehlleitungen führen.

- Prüfe, welche Geräte mDNS benötigen und auf welchen Schnittstellen es aktiv ist.
- Netzsegmentierung begrenzt, wer Anfragen und Antworten erreichen kann.
- mDNS ist nicht dasselbe wie LLMNR oder NBT-NS; Schutzmaßnahmen müssen zum Dienst passen.

3. SMB-RELAYING



Beim Relaying wird eine eingehende NTLM-Authentifizierung an einen anderen Dienst weitergereicht. Ist der Zielserver nicht ausreichend geschützt, kann der Angreifer im Kontext des authentisierten Kontos handeln.

- SMB-Signing erschwert Relaying zu SMB, wenn es verpflichtend aktiviert und korrekt durchgesetzt wird.
- Signing verschlüsselt nicht den Datenverkehr und verhindert nicht jedes Relaying zu anderen Protokollen.
- Erfolg hängt unter anderem von erreichbaren Diensten, Kontorechten und Konfigurationen ab.

4. LDAP-SIGNING UND KANALBINDUNG



LDAP-Signing schützt die Integrität signierter LDAP-Verbindungen. LDAP-Kanalbindung kann die Authentifizierung an den TLS-Kanal binden. Diese Schutzmechanismen erschweren bestimmte NTLM-Relaying-Szenarien zu LDAP.

- Signing und Kanalbindung sind unterschiedliche Schutzmechanismen.
- Ob ein Relaying gelingt, hängt von Protokoll, Serverrichtlinien und Clientverhalten ab.
- Richtlinien kontrolliert und schrittweise durchsetzen; Auswirkungen auf Anwendungen prüfen.

5. WEBDAV-MISSBRAUCH



WebDAV erweitert HTTP um Funktionen zur Verwaltung von Dateien. Wenn ein Client eine WebDAV-Ressource anspricht, kann dies in bestimmten Umgebungen eine NTLM-Authentifizierung auslösen.

- Unerwartete Dateipfade oder Links können Verbindungen zu nicht vertrauenswürdigen Servern provozieren.
- Ausgehende Verbindungen und WebDAV-Nutzung nach Bedarf begrenzen.
- NTLM-Ausgangsauthentifizierung zu nicht vertrauenswürdigen Zielen einschränken.

6. WPAD-ANGRIFFE



WPAD ermöglicht Clients, eine Proxy-Konfiguration automatisch zu finden. Manipulierte Namensauflösung oder eine nicht vertrauenswürdige Konfigurationsquelle kann Clients zu einem falschen Proxy führen.

- WPAD nur einsetzen, wenn es kontrolliert und tatsächlich benötigt wird.
- Namensauflösung, DHCP-Konfiguration und Proxy-Vertrauenswürdigkeit absichern.
- Unnötige automatische Erkennung deaktivieren und Proxy-Einstellungen zentral verwalten.

7. NTLMv2-CRACKING



NTLMv2 verwendet eine Challenge-Response-Antwort. Ein abgefangener Antwortwert kann offline gegen Passwortkandidaten geprüft werden; das ist kein direktes Wiederherstellen des Passworts.

- Lange, einzigracte Passwötrer erhöhen den Aufwand für Wörterbuch- und Brute-Force-Angriffe.
- Eine Antwort kann sensible Authentifizierungsdaten enthalten, bleibt aber von einem Passwort-Hash im üblichen Sinn zu unterscheiden.
- Weitergabe oder Prüfung selcher Daten ausschließlich in genehmigten Tests.

8. GEGENMASSNAHMEN



Reduziere Gelegenheiten für Fehlleitung, begrenze NTLM und erzwinge Protokollschutz. Mehrere abgestimmte Kontrollen sind wirksamer als eine einzelne Einstellung.

- LLMNR und NBT-NS deaktivieren, sofern sie nicht benötigt werden; verbleibende Namensauflösung kontrollieren.
- SMB-Signing verpflichtend aktivieren, wo möglich; LDAP-Signing und Kanalbindung pessend zur Umgebung durchsetzen.
- NTLM-Nutzung und ausgehende Authentifizierung beschränken; mDNS, WPAD und WebDAV gezielt konfigurieren.
- Netze segmentieren, ungewöhnliche Namensauflösungsantworten und Authentifizierungsereignisse überwachen.

SICHERHEITSGRENZE



Diese Mindmap erklärt Angriffsmuster zu Lernzwecken. Tests nur in ausdrücklich autorisierten Umgebungen durchführen. Ein Diagramm ist keine Diagnose oder Handlungsempfehlung für einen konkreten Vorfall.

Mindmap 066 · C2-Architekturen und Egress-Techniken

066 · COMMAND AND CONTROL (C2)

C2 bezeichnet die Kommunikation, mit der Angreifer kompromittierte Systeme steuern. Die folgenden Punkte ordnen typische Architektur- und Tarnkonzepte ein. Nutzung ausschließlich zur Erkennung, Analyse und autorisierten Sicherheitsarbeit.

1. TIERED C2-ARCHITEKTUR



Mehrere Ebenen verteilen die Kommunikation zwischen implantierten Systemen und Steuerungspunkten.

- Endpunkte kommunizieren gegebenenfalls zunächst mit einem Zwischenknoten.
- Zwischenknoten können Kommunikation weiterleiten oder bündeln.
- Eine Hierarchie kann direkte Verbindungen zur zentralen Steuerung verringern.
- Jede zusätzliche Ebene schafft Abhängigkeiten und mögliche Ausfallpunkte.

2. HTTP/HTTPS-BEACONING



Ein Beacon ist eine wiederkehrende Kontaktaufnahme eines kompromittierten Systems mit einer Steuerungsinfrastruktur.

- HTTP oder HTTPS kann für die Übertragung verwendet werden.
- Regelmäßige Kontakte können ein zeitliches Muster bilden.
- HTTPS verschlüsselt den Transport, macht den Verkehr aber nicht automatisch legitim oder unauffällig.
- Abwehr: Ziele, Prozesskontext, Häufigkeit und ungewöhnliche Verbindungen gemeinsam prüfen.

3. DNS-TUNNELING



DNS-Tunneling missbraucht DNS-Anfragen oder -Antworten, um Daten in DNS-Verkehr zu transportieren.

- Auffällige Abfragemuster oder ungewöhnlich lange Namen können Hinweise sein.
- Hohe Anfragevolumen und seltene Domains verdienen Kontextprüfung.
- DNS ist für Namensauflösung vorgesehen; Tunnelverkehr kann diese Funktion zweckentfremden.

Abwehr: DNS-Protokollierung, erlaubte Resolver und Anomalieanalyse einsetzen.

4. DOMAIN FRONTING



Bei Domain Fronting können äußere Verbindungs-Informationen und ein verschlüsselter Anwendungsinhalt unterschiedliche Ziele nahelegen.

- Die Methode nutzt Eigenschaften von HTTPS und Infrastruktur wie Vermittlungsdiensten.
- Beobachtbarkeit hängt davon ab, welche Metadaten und Inhalte verfügbar sind.
- Konfigurationen und Unterstützung durch Dienste ändern sich; die Technik ist nicht überall möglich.
- Abwehr: erlaubte Dienste, Endpunkttelemetrie und Netzwerkindikatoren zusammen auswerten.

5. UMGEHUNG VON EGRESS-FILTERN



Egress-Filtering begrenzt ausgehende Verbindungen. Angreifer können versuchen, erlaubte Wege oder Dienste zweckzuentfremden.

- Zu breite Regeln können unerwünschte ausgehende Verbindungen ermöglichen.
- Erlaubter Web- oder DNS-Verkehr ist nicht automatisch vertrauenswürdig.
- Least Privilege: Ziele, Ports und Protokolle auf den erforderlichen Bedarf beschränken.
- Abweichungen von erwarteten Zielen und Prozessen überwachen.

6. C2-FRAMEWORKS



Cobalt Strike und Sliver sind Beispiele für Frameworks, die in autorisierten Tests eingesetzt werden können und auch von Angreifern missbraucht werden.

- Ein Werkzeugname allein beweist keinen Angriff.
- Kontext wie Prozess, Host, Netzwerkziel und Testfreigabe ist entscheidend.
- Erkennungsregeln sollten Verhalten und technische Indikatoren betrachten.
- Einsatz nur in ausdrücklich autorisierten Umgebungen.

7. VERSCHLÜSSELUNG VON PAYLOADS



Verschlüsselung kann Inhalte während der Übertragung vor Mitlesen schützen. Sie ist kein Beleg für bösartige oder harmlose Aktivität.

- Transportverschlüsselung schützt Daten auf dem Übertragungsweg.
- Verschlüsselung erschwert Inhaltsprüfung, beseitigt aber nicht alle Metadaten.
- Schlüsselverwaltung und Endpunktsicherheit bleiben wesentlich.
- Abwehr: Metadaten, Endpunktverhalten und zulässige Ziele ergänzend berücksichtigen.

8. TRAFFIC-JITTERING



Jittering variiert Zeitabstände wischen wiederkehrenden Verbindungen und kann starre zeitliche Muster verändern.

- Unregelmäßige Intervalle können einfache Takt-Erkennung erschweren.
- Jittering ändert nicht automatisch Ziel, Prozess oder Zweck der Verbindung.
- Zeitliche Analyse sollte mit weiteren Netzwerk- und Endpunktindikatoren kombiniert werden.
- Baselines und Abweichungen helfen, auffällige Kommunikation einzuordnen.

SICHERHEITSKONTEXT



Die genannten Techniken können missbraucht werden. Diese Übersicht beschreibt sie auf konzeptioneller Ebene; sie enthält keine Anleitung zum Aufbau oder Betrieb einer C2-Infrastruktur.

MINDMAP 067 · EDR-EVASION UND DEFENSE EVASION-TECHNIKEN

DEFENSE EVASION

Angreifer versuchen, Erkennung zu umgehen. Für Verteidiger zählt das Zusammenspiel von Telemetrie, Schutzkontrollen und Kontext — kein einzelnes Signal beweist einen Angriff.

- Defense Evasion bezeichnet Techniken, mit denen Aktivitäten verborgen oder Sicherheitskontrollen umgangen werden sollen.

1. API UNHOOKING



Sicherheitssoftware kann Betriebssystem-APIs beebachten. Manipulationen an solchen Beobachtungspunkten können die Sichtbarkeit verändern.

- Achte auf unerwartete Änderungen an geladenen Modulen oder API-Einstiegspunkten.
- Vergleiche Prozessverhalten mit vertrauenswürdiger Systemtelemetrie.

2. DIRECT SYSTEM CALLS



Systemaufrufe vermitteln Anwendungen den Zugang zu Betriebssystemfunktionen. Abweichende Aufrufmuster können für die Analyse relevant sein.

- Erkenne: direkte Systemaufrufe sind nicht automatisch bösartig.
- Bewerte sie zusammen mit Herkunft, Prozessverhalten und weiteren Ereignissen.

3. PARENT-PID-SPOOFING

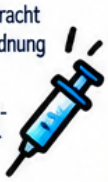


Ein Prozess kann einen irreführenden Elternprozess-Bezug aufweisen. Dadurch wirkt seine Prozesskette auf den ersten Blick plausibler.

- Prüfe Prozessbaum, Startzeit, Benutzerkonto und gestartete Befehle gemeinsam.
- Ungewöhnliche Eltern-Kind-Beziehungen sind ein Hinweis, kein Beweis.

4. PROCESS INJECTION

Bei Process Injection wird Code in den Kontext eines anderen Prozesses eingebracht oder dort ausgeführt. Das kann die Zuordnung von Aktivitäten erschweren.



- Untersuche auffällige Speicher-, Thread- und Prozessbeziehungen mit geeigneter Endpoint-Telemetrie.
- Berücksichtige legitime Software, die ebenfalls Prozesse instrumentiert.

5. LIVING-OFF-THE-LAND BINARIES (LOLBINS)

Legitime, bereits vorhandene Systemprogramme können für unerwartete Zwecke missbraucht werden.



- Bewerte Programm, Argumente, Elternprozess, Benutzer und Zielaktivität im Kontext.
- Eine yültige Signatur oder ein bekannter Systempfad macht eine Ausführung nicht automatisch harmlos.

6. AMSI-BYPASS



AMSI ist eine Windows-Schnittstelle, über die Anwendungen Inhalte zur Prüfung an Sicherheitsprodukte übergeben können. Umgehungsversuche können diese Sicht beeinträchtigen.

- Achte auf Veränderungen an Prüfpfaden, auffällige Skriptaktivität und korrelierte Prozessereignisse.
- AMSI ist eine Schutzschicht, kein vollständiger Ersatz für weitere Kontrollen.

7. ETW-PATCHING



Event Tracing for Windows (ETW) stellt Windows-Komponenten Ereignisverfolgung bereit. Manipulationen können bestimmte Beobachtungen beeinträchtigen.

- Suche nach unerwarteten Änderungen an Prozessen oder Komponenten, die Ereignisse erfassen.
- Korreliere Endpoint-Ereignisse mit unabhängigen Protokollquellen, soweit verfügbar.

8. MASQUERADING



Bei Masquerading werden Namen, Pfade oder andere Merkmale so gewählt, dass eine Datei oder ein Prozess vertrauenswürdig erscheint.

- Vergleiche Dateiname und Pfad mit Signatur, Metadaten, Herkunft und tatsächlichem Verhalten.
- Achte auf kleine Abweichungen bei Namen sowie unerwartete Speicherorte.

EINORDNUNG

Diese Mindmap dient der Sicherheitsbildung und Verteidigung. Sie enthält keine Anleitung zum Umgehen von Schutzsystemen.

068 · CLOUD-SICHERHEITSARCHITEKTUR

Shared Responsibility · IaaS · PaaS · SaaS

1 · GETEILTE VERANTWORTUNG

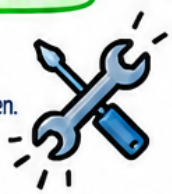
Anbieter und Kunde sichern unterschiedliche Teile des Dienstes. Die genaue Aufteilung variiert je Angebot.



- IaaS: Anbieter schützt Rechenzentrum und Hardware; Kunde meist Betriebssystem, Identitäten und Daten.
- PaaS: Anbieter verwaltet zusätzlich Laufzeit und Plattform; Kunde schützt Anwendung, Zugriffe und Daten.
- SaaS: Anbieter betreibt die Anwendung; Kunde bleibt für Benutzer, Konfiguration und den Umgang mit Daten verantwortlich.

2 · CLOUD-CONTROL-PLANE

Die Control Plane ist die Verwaltungsebene zum Erstellen und Ändern von Cloud-Ressourcen. Sie ist von den laufenden Workloads zu unterschieiden.



- Zugriff streng begrenzen und privilegierte Konten besonders schützen.
- Änderungen protokollieren und regelmäßig auf ungewöhnliche Aktivitäten prüfen.
- Fehlkonfigurationen können Ressourcen oder Daten unbeabsichtigt exponieren.

3 · MANDANTENTRENNUNG

Multi-Tenancy bedeutet, dass mehrere Kunden Ressourcen einer Cloud-Plattform nutzen. Logische Isolation soll verhindern, dass ein Mandant auf Ressourcen eines anderen zugreift.



- Die Trennung wird durch Plattform- und Virtualisierungsmechanismen umgesetzt.
- Kunden sollten Zugriffsrechte und exponierte Dienste dennoch korrekt konfigurieren.
- Mandantentrennung ist kein Ersatz für Verschlüsselung oder Zugriffskontrollen.

4 · CSA CLOUD CONTROLS MATRIX

Die Cloud Controls Matrix (CCM) der Cloud Security Alliance ordnet Cloud-Sicherheitskontrollen in Themenbereiche ein. Sie kann zur Orientierung und zum Vergleich von Kontrollanforderungen dienen.



- Kontrollbereiche umfassen unter anderem Governance, Identität und Datenschutz.
- Die Matrix erstezt keine Prüfung des konkreten Dienstes oder seiner Konfiguration.
- Anwendbarkeit und Nachweise müssen zum eigenen Risiko und Kontext passen.

5 · VIRTUAL PRIVATE CLOUD

Eine VPC ist ein logisch abgegrenztes virtuelles Netzwerk in einer Cloud-Umgebung. Sie ermöglicht die Strukturierung von IP-Bereichen und Netzwerkzugriffen.



- Subnetze teilen den Adressraum in Bereiche auf.
- Routing bestimmt mögliche Netzwerkpfade.
- Firewall- oder Sicherheitsregeln begrenzen den Verkehr nach Quelle, Ziel und Dienst.

6 · SERVICE-ENDPUNKTE

Endpunkte verbinden Workloads mit Cloud-Diensten. Je nach Dienst und Konfiguration erfolgt der Zugriff über öffentliche oder private Netzwerkpfade.



- Öffenlliche Erreichbarkeit nur aktivieren, wenn sie benötigt wird.
- Private Endpunkte können den Zugriff auf interne Netzwerkpfade beschränken.
- Zugriffsrichtlinien und Authentisierung bleiben auch bei privater Erreichbarkeit nötig.

7 · IDENTITÄTSGRENZEN

Identitäts- und Zugriffsverwaltung (IAM) legt fest, welche Personen und Dienste auf welche Ressourcen zugreifen dürfen.



- Rollen und Rechte nach dem Prinzip der geringsten Berechtigung vergeben.
- Menschliche Konten und Dienstidentitäten getrennt betrachten.
- Starke Anmeldung, regelmäßige Rechteprüfung und nachvollziehbare Protokolle unterstützen die Kontrolle.

8 · CLOUD-COMPLIANCE-STANDARDS

Compliance-Anforderungen hängen von Branche, Daten und Rechtsraum ab. Standards und Prüfberichte können Hinweise auf Kontrollen geben, garantieren aber nicht automatisch die Sicherheit einer konkreten Nutzung.



- Anwendbare Anforderungen und Verantwortlichkeiten ermitteln.
- Nachweise des Anbieters passend zum Dienst und Geltungsbereich prüfen.
- Eigene Konfiguration, Zugriffe und Datenverarbeitung weiterhin kontrollieren.

AWS IAM DEEP DIVE

Identitäten, temporäre Zugriffe und Richtlinien wirksam begrenzen und überwachen.

1. IAM-ROLLEN UND POLICIES



Rollen stellen temporäre Berechtigungen bereit. Eine Vertrauensrichtlinie bestimmt, wer eine Rolle annehmen darf; Berechtigungsrichtlinien bestimmen, was die Rolle tun kann.

- Policies beschreiben erlaubte oder ausdrücklich verweigerte Aktionen und Ressourcen.
- Rollen eignen sich für AWS-Dienste, verbundene Nutzer und kontoübergreifenden Zugriff.
- Vergib möglichst eng begrenzte Rechte nach dem Least-Privilege-Prinzip.

2. ASSUMEROLE UND STS



AWS Security Token Service (STS) stellt temporäre Sicherheitsanmeldedaten bereit. AssumeRole fordert solche Anmeldedaten für eine Rolle an, sofern deren Vertrauensrichtlinie den Aufruf zulässt.

- Temporäre Anmeldedaten umfassen Zugriffsschlüssel, geheimen Schlüssel und Sitzungstoken.
- Die Sitzung ist zeitlich begrenzt; ihre Rechte ergeben sich aus der Rolle und weiteren Berechtigungsgrenzen.
- Schütze die Anmeldedaten und vermeide ihre Ablage in Quellcode oder öffentlichen Dateien.

3. SERVICE CONTROL POLICIES (SCPs)



SCPs gehören zu AWS Organizations und legen die maximal verfügbaren Berechtigungen für Konten oder Organisationseinheiten fest.

- Eine SCP erteilt selbst keine Berechtigung; IAM-Richtlinien müssen den Zugriff weiterhin erlauben.
- Eine SCP kann Aktionen begrenzen, selbst wenn eine IAM-Richtlinie sie erlaubt.
- Prüfe Richtlinienänderungen sorgfältig, damit notwendige Verwaltungszugriffe nicht versehentlich blockiert werden.

4. PERMISSION BOUNDARIES



Eine Permission Boundary setzt die Obergrenze für die Berechtigungen, die einer IAM-Identität durch identitätsbasierte Policies gewährt werden können.

- Sie erteilt keine Rechte für sich allein.
- Effektive Berechtigungen hängen vom Zusammenspiel der anwendbaren Richtlinien und expliziten Verweigerungen ab.
- Nützlich, um delegierte Verwaltung oder die Erstellung von Rollen und Nutzern einzuschränken.

5. IAM ACCESS ANALYZER



IAM Access Analyzer hilft, Richtlinien zu prüfen und Zugriffe zu erkennen, die über eine Vertrauensgrenze hinausgehen können.

- Externe Zugriffsanalysatoren melden etwa Ressourcenfreigaben an externe Konten oder öffenliche Zugriffe.
- Richtlinienprüfungen können unerwartete oder übermäßig weit gefasste Berechtigungen aufzeigen.
- Ein Befund ist ein Prüfhinweis: Bewerte, ob der Zugriff beabsichtigt ist, und passe Richtlinien bei Bedarf an.

6. INSTANCE METADATA SERVICE (IMDSv2)



IMDS ermöglicht einer EC2-Instanz den Zugriff auf Metadaten und gegebenenfalls auf temporäre Anmeldedaten ihrer Instanzrolle. IMDSv2 verwendet einen sitzungsorientierten Token-Ablauf.

- Konfiguriere nach Möglichket IMDSv2 als erforderlich und begrenze den Hop-Limit passend zur Umgebung.
- Gib Instanzrollen nur die Berechtigungen, die die Arbeitslast benötigt.
- Verhindere, dass nicht vertrauenswürdige Anwendungen oder Nutzer auf den Metadienst zugreifen.

7. ROOT-KONTO SCHÜTZEN



Das Root-Konto einer AWS-Kontoidentität hat weitreichende Rechte. Nutze es nur für Aufgaben, die Root ausdrücklich erfordern.

- Aktiviere MFA und sichere Wiederherstellungsinformationen.
- Verwende keine Root-Zugriffsschlüssel für alltägliche Aufgaben.
- Erstelle individuelle Rollen oder Identitäten mit begrenzten Rechten für den laufenden Betrieb.

8. AUDIT MIT AWS CLOUDTRAIL



AWS CloudTrail zeichnet AWS-API-Aktivitäten als Ereignisse auf und unterstützt damit Nachvollziehbarkeit und Untersuchungen.

- Prüfe, ob Trails und Ereignisdatenspeicher die benötigten Konten und Regionen abdecken.
- Schütze gespeicherte Protokolle vor unbefugtem Zugriff und Änderungen.
- Nutze CloudTrail-Ereignisse zusammen mit weiteren Protokollen und Kontext; ein einzelnes Ereignis erklärt nicht immer die gesamte Ursache.

070 MICROSOFT ENTRA ID

Identitäten, Anmeldungen und Zugriff im Cloud-Ökosystem.

Cloudbasierter Identitätsdienst für Anmeldung, Identitätsverwaltung und Zugriff auf Anwendungen und Ressourcen. Sicherheit entsteht durch zusammenwirkende Kontrollen; verfügbare Funktionen hängen von Konfiguration und Lizenz ab.

- **Identität:** Wer oder was meldet sich an?
- **Authentifizierung:** Wie wird die Identität nachgewiesen?
- **Autorisierung:** Worauf darf sie zugreifen?



1. CONDITIONAL ACCESS POLICIES



Richtlinien für bedingten Zugriff werten Sign-in-Signale aus und setzen Kontrollen durch, bevor Zugriff gewährt wird.

- Bedingungen können Benutzer, Anwendung, Standort, Gerät und Risiko umfassen.
- Mögliche Kontrollen: MFA verlangen, Zugriff blockieren oder Gerätezustand verlangen.
- Richtlinien sorgfältig testen; Ausschlüsse und Notfallkonten bewusst begrenzen.

2. PRIVILEGED IDENTITY MANAGEMENT (PIM)



PIM unterstützt die zeitlich und organisatorisch kontrollierte Vergabe privilegierter Rollen.

- Berechtigungen können „eligible" (bei Bedarf aktivierbar) statt dauerhaft aktiv sein.
- Aktivierung kann Begründung, MFA, Genehmigung oder zeitliche Begrenzung erfordern.
- Aktivierungen und Rollenänderungen überwachen; kleinste nötige Rechte vergeben.

3. ENTRA ID PROTECTION



Identitätsbasierte Risikoerkennung kann auffällige Anmeldungen und kompromittierungsverdächtige Benutzer erfassen.

- Sign-in-Risiko bezieht sich auf eine Anmeldung; Benutzerrisiko auf die Identität.
- Erkennungen sind Signale, keine sichere Feststellung eines Angriffs.
- Untersuchen, angemessen reagieren und Fehlalarme berücksichtigen; Funktionen variieren nach Lizenz.

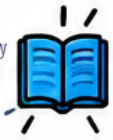
4. APP REGISTRATIONS & SERVICE PRINCIPALS



Eine App Registration beschreibt eine Anwendung in einem Mandanten. Ein Service Principal ist ihre konkrete Identität in einem Mandanten und kann dort Zugriffsrechte erhalten.

- Anwendungsberechtigungen und delegierte Berechtigungen unterscheiden.
- Geheimnisse und Zertifikate schützen, Ablaufdaten überwachen und unnötige Berechtigungen entfernen.
- Zustimmung zu Berechtigungen kontrollieren; weitreichende App-Rechte erhöhen das Risiko.

5. ENTRA CONNECT & PASS-THROUGH AUTHENTICATION



Entra Connect synchronisiert ausgewählte Identitätsdaten zwischen lokalem Active Directory und Entra ID. Pass-through Authentication (PTA) prüft Kennwörter bei der Anmeldung über lokale Agenten.

- Synchronisierung ist nicht dasselbe wie Föderation oder Kennwort-Hash-Synchronisierung.
- PTA-Agenten und ihre Server benötigen Schutz, Verfügbarkeit und Überwachung.
- Änderungen an Synchronisierung und Authentifizierung geplant testen; Abhängigkeiten dokumentieren.

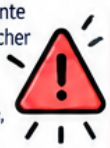
6. B2B- UND B2C-IDENTITÄTEN



B2B ermöglicht Zusammenarbeit mit externen Personen, meist über deren Identität im eigenen Entra-Mandanten. Kundenidentitäten werden je nach Architektur über passende externe Identitätslösungen verwaltet.

- Gastzugriff gezielt auf Ressourcen und Gruppen begrenzen.
- Einladung, Annahme, Zugriff und regelmäßige Überprüfung berücksichtigen.
- Kundenidentitäten von Mitarbeitendenidentitäten und deren Lebenszyklen unterscheiden.

7. RISIKOBASIERTE SIGN-IN-REGELN



Risikobasierte Regeln verknüpfen erkannte Risiken mit einer Reaktion, etwa zusätzlicher Authentifizierung oder Zugriffssperre.

- Risikosignale können durch IP-Adresse, Anmeldeverhalten oder andere Erkennungen entstehen.
- Reaktion nach Schutzbedarf festlegen und Auswirkungen auf legitime Benutzer beachten.
- Risikostufen und Erkennungen regelmäßig prüfen; Verfügbarkeit ist lizenzabhängig.

8. CONTINUOUS ACCESS EVALUATION (CAE)



CAE ermöglicht unterstützten Diensten, bestimmte Änderungen oder Ereignisse während einer Sitzung zeitnah zu berücksichtigen. statt ausschließlich auf den Ablauf eines Tokens zu warten.

- Unterstützung hängt von Client, Dienst und Ereignis ab; nicht jede Sitzung wird kontinuierlich neu bewertet.
- Ereignisse können etwa Kontodeaktivierung oder Änderungen an Richtlinien umfassen.
- CAE ergänzt weitere Zugriffskontrollen und ist kein Ersatz für MFA oder sichere Rollenvergabe.

EINORDNUNG

Microsoft Entra ID hieß früher Azure Active Directory (Azure AD). Es ist kein klassischer Domänencontroller und ersetzt nicht automatisch jede lokale Verzeichnisverwaltung.

071 CSPM & COMPLIANCE

Cloud-Konfigurationen kontinuierlich erfassen, prüfen und Abweichungen nachvollziehbar behandeln.

1. KONFIGURATIONS-DRIFT-ERKENNUNG



Drift bezeichnet eine Veränderung gegenüber einem festgelegten Sollzustand. CSPM kann aktuelle Cloud-Konfigurationen wiederholt mit Baselines oder Richtlinien vergleichen.

- Sollzustand dokumentieren, etwa erlaubte Speicherfreigaben oder Netzwerkregeln.
- Abweichungen mit Ressource, Zeitpunkt und Regelbezug erfassen.
- Änderungen prüfen: Sie können beabsichtigt, genehmigt oder riskant sein.

2. CIS CLOUD BENCHMARKS



CIS Benchmarks sind veröffentlichte Empfehlungen zur sicheren Konfiguration bestimmter Technologien. Für Cloud-Umgebungen können passende Benchmarks als Orientierung für Prüfregeln dienen.

- Benchmark und Version passend zu Anbieter und Dienst auswählen.
- Empfehlungen mit eigener Architektur und Betriebsanforderungen abgleichen.
- Abweichungen begründen und dokumentieren; nicht jede Empfehlung passt unverändert zu jedem Einsatz.

3. MULTI-CLOUD-ASSET-INVENTARISIERUNG



Ein Asset-Inventar erfasst Cloud-Ressourcen über Konten, Projekte oder Anbieter hinweg. Es schafft eine Grundlage dafür, Ressourcen Sicherheitsverantwortlichen und Prüfregeln zuzuordnen.

- Ressourcentyp, Konto oder Projekt und zuständiges Team erfassen.
- Inventar regelmäßig aktualisieren und verwaiste oder unbekannte Ressourcen untersuchen.
- Zugriff auf Inventardaten begrenzen und sensible Metadaten schützen.

4. AUTOMATISIERTE REMEDIERUNG



Remediierung bedeutet, eine erkannte Abweichung zu beheben. Automatisierung kann klar begrenzte Änderungen beschleunigen, birgt aber das Risiko unbeabsichtigter Auswirkungen.

- Nur eindeutige, risikoarme Fälle automatisiert behandeln.
- Berechtigungen möglichst eng begrenzen und Änderungen protokollieren.
- Vor produktiven Änderungen testen; für kritische Eingriffe Freigabe und Rückweg vorsehen.



SICHERHEITSGRENZE
Eine automatisierte Korrektur ist keine Garantie für eine sichere Gesamtkonfiguration. Änderungen müssen nachvollziehbar und überprüfbar bleiben.

5. FEHLKONFIGURATIONS-SCANS



Scans prüfen Cloud-Konfigurationen anhand definierter Regeln, zum Beispiel auf öffentlich erreichbare Speicher oder zu weit gefasste Netzwerkzugriffe. Ein Fund ist ein Hinweis, der im Kontext bewertet werden muss.

- Prüfumfang und verwendete Regeln kennen.
- Fundstelle, betroffene Ressource und mögliche Auswirkung nachvollziehen.
- Fehlalarme und Ausnahmen begründen, begrenzen und regelmäßig erneut prüfen.

6. POLICY-AS-CODE



Policy-as-Code beschreibt Sicherheitsregeln in maschinenlesbarer Form, damit sie automatisiert geprüft werden können, etwa bei Änderungen an Cloud-Ressourcen.

- Regeln versionieren und Änderungen nachvollziehbar reviewen.
- Prüfungen früh in Bereitstellungs- oder Änderungsabläufe integrieren.
- Ausnahmen mit Zweck, Zuständigkeit und Ablaufdatum dokumentieren.

7. COMPLIANCE-REPORTING



Compliance-Berichte fassen Prüfergebnisse und Abweichungen zu ausgewählten Anforderungen zusammen. Ein Bericht bildet nur den erfassten Umfang und den Prüfzeitpunkt ab, er ist kein automatischer Nachweis vollständiger Compliance.

- Anforderung, Prüfregel, Ressourcenumfang und Zeitstempel kenntlich machen.
- Abweichungen, Ausnahmen und offene Maßnahmen getrennt darstellen.
- Berichte mit Verantwortlichen abstimmen und Belege nachvollziehbar aufbewahren.

8. SCHATTEN-IT-IDENTIFIKATION



Schatten-IT umfasst Cloud-Dienste oder Ressourcen, die außerhalb etablierter Freigabe- oder Inventarprozesse genutzt werden. Erkennung kann Hinweise liefern, die anschließend geklärt werden müssen.

- Unbekannte Konten, Dienste oder Ressourcen anhand verfügabeer Inventar- und Nutzungsdaten untersuchen.
- Eigentümerschaft und legitimen Geschäftszweck klären, bevor Maßnahmen erfolgen.
- Ressourcen in geeignete Verwaltungs- und Sicherheitsprozesse aufnehmen oder kontrolliert stilllegen.

072 · CWPP

Schutzfunktionen für Cloud-Workloads – von virtuellen Maschinen bis Containern und Serverless.

1. Runtime-Schutz für Instanzen



Beobachtet laufende virtuelle Maschinen und Server, um verdächtige Aktivitäten während des Betriebs zu erkennen oder zu blockieren.

- Prozesse, Systemaufrufe oder Netzwerkverbindungen überwachen.
- Warnungen anhand von Regeln oder Verhaltensmustern erzeugen.
- Eingriffe können je nach Konfiguration automatisch erfolgen.

2. Systemintegritätsprüfung



Prüft, ob wichtige Dateien, Einstellungen oder Systemkomponenten verändert wurden – insbesondere im Vergleich zu einem bekannten Sollzustand.

- Änderungen an kritischen Dateien und Konfigurationen erfassen.
- Erwartete Wartungsänderungen von unerwarteten Abweichungen unterscheiden.
- Eine Abweichung ist ein Hinweis zur Untersuchung, nicht automatisch ein Angriff.

3. Host-basierte IPS und Firewalls



Host-Schutzregeln kontrollieren Netzwerkverkehr und bestimmte Aktivitäten direkt auf einer Instanz. Eine Firewall filtert Verbindungen nach Regeln; ein Intrusion Prevention System (IPS) kann verdächtige Muster erkennen und gegebenenfalls blockieren.

- Regeln auf erlaubte Quellen, Ziele und Dienste begrenzen.
- Regelsätze regelmäßig prüfen und unnötige Freigaben entfernen.
- Verschlüsselter Verkehr und Fehlkonfigurationen können die Erkennung begrenzen.

4. Schwachstellenprüfung für Serverless-Funktionen



Serverless-Funktionen laufen in einer vom Anbieter verwalteten Ausführungsumgebung, Prüfungen konzentrieren sich häufig auf Funktionscode, Abhängigkeiten, Berechtigungen und Konfigurationen.

- Bekannte Schwachstellen in verwendeten Bibliotheken suchen.
- Übermäßige Berechtigungen und unsichere Einstellungen identifizieren.
- Prüfergebinsse priorisieren und vor dem Einsatz beheben oder begründet behandeln.

5. Anwendungs-Sandboxing



Eine Sandbox führt Anwendungen oder Komponenten in einer stärker isolierten Umgebung aus, um deren Einfluss auf andere Ressourcen zu begrenzen.

- Zugriffe auf Dateien, Prozesse und Netzwerk einschränken.
- Unbekanntes Verhalten kontrolliert beebachten.
- Sandboxing reduziert Risiken, garantiest aber keine vollständige Eindämmung.

6. Micro-Segmentierung



Teilt Workloads in kleinere Kommunikationsbereiche und setzt Regeln dafür, welche Systeme miteinander sprechen dürfen.

- Kommunikationsbedarf erfassen, bevor Regeln erstellt werden.
- Verbindungen standardmäßig möglichst eng erlauben.
- Regeln testen und überwachen, damit legitime Dienste erreichbar bleiben.

7. Container-Workload-Monitoring



Überwacht Container-Workloads zur Laufzeit und kann Informationen aus Container- und Orchestrierungsumgebungen berücksichtigen.

- Prozesse, Netzwerkaktivität und auffällige Änderungen beobachten.
- Container-Images vor dem Einsatz auf Risiken prüfen, sofern die Plattform dies unterstützt.
- Laufzeitüberwachung und Image-Prüfung decken unterschiedliche Phasen ab.

8. Verhaltensanomalieerkennung



Sucht nach Aktivitäten, die vom üblichen Verhalten eines Workloads abweichen, etwa ungewöhnlichen Prozessen oder Kommunikationsmustern.

- Eine Baseline hilft, typische Aktivität einzuordnen.
- Abweichungen können harmlose Ursachen haben, zum Beispiel Updates oder Lastspitzen.
- Alarme mit Kontext untersuchen; nicht jede Abweichung ist ein bestätigter Vorfall.

073 CONTAINERSICHERHEIT

1. DOCKER-DAEMON SCHÜTZEN

Der Daemon verwaltet Container und Images und kann weitreichende Host-Rechte besitzen.



- Zugriff auf den Docker-Socket nur vertrauenswürdigen Konten erlauben.
- Socket nicht ungeschützt über TCP veröffentlichen.
- Daemon und Host aktuell halten; unnötige Funktionen deaktivieren.

2. ROOTLESS-CONTAINER

Rootless Docker lässt Daemon und Container als unprivilegierter Benutzer laufen; User Namespaces ordnen Container-IDs Host-IDs zu.



- Begrenzt mögliche Folgen eines Container-Ausbruchs, beseitigt das Risiko aber nicht.
- Funktionsumfang und Anforderungen unterscheiden sich je nach Umgebung.
- Nicht mit dem bloßen Verzicht auf den Prozessbenutzer root im Container verwechseln.

3. NAMESPACES UND CGROUPS

Linux-Namespaces trennen Ansichten von Ressourcen; cgroups organisieren und begrenzen Ressourcennutzung.



- Namespaces betreffen etwa Prozesse, Netzwerk, Mounts und Benutzer-IDs.
- cgroups helfen, CPU- und Speicherverbrauch zu begrenzen.
- Beides ergänzt Isolation, ist aber keine eigenständige Sicherheitsgarantie.

4. SECCOMP-PROFILE

Seccomp filtert Systemaufrufe, die Prozesse an den Kernel richten können.



- Ein passendes Profil reduziert erreichbare Kernel-Funktionen.
- Standardprofile sind ein Ausgangspunkt; Ausnahmen nur begründet ergänzen.
- Ungetestete Einschränkungen können Anwendungen stören.

5. APPARMOR UND SELINUX

Linux-Sicherheitsmodule setzen zusätzliche Zugriffsregeln für Prozesse und Ressourcen durch.



- AppArmor-Profile begrenzen unter anderem Datei- und Prozesszugriffe.
- SELinux steuert Zugriffe anhand von Sicherheitskontexten und Regeln.
- Richtlinien gezielt testen und nicht pauschal deaktivieren, wenn Probleme auftreten.

6. MINIMALE BASIS-IMAGES

Kleine, vertrauenswürdige Basis-Images können unnötige Pakete und damit Angriffsfläche verringern.



- Nur erforderliche Laufzeitkomponenten aufnehmen.
- Quelle und Version nachvollziehbar festlegen; Updates einplanen.
- Klein bedeutet nicht automatisch sicher: Konfiguration und Abhängigkeiten zählen ebenfalls.

7. IMAGE-SCHWACHSTELLEN SCANNEN

Scanner vergleichen Paket- und Image-Inventare mit bekannten Schwachstellen-informationen.



- Images vor Bereitstellung und regelmäßig im Betrieb prüfen.
- Ergebnisse nach Schwere, Ausnutzbarkeit und Einsatzkontext priorisieren.
- Ein Scan findet nicht jede Schwachstelle und beweist keine Fehlerfreiheit.

8. SECRETS SICHER HANDHABEN

Passwörter, Tokens und Schlüssel gehören nicht in den Image-Build oder den Quellcode.



- Secrets nicht in Dockerfile-Anweisungen oder Image-Layer kopieren.
- Geeignete Secret-Verwaltung oder zur Laufzeit bereitgestellte Zugangsdaten nutzen.
- Build-Secrets so einsetzen, dass sie nicht im fertigen Image verbleiben; Zugriff begrenzen und rotieren.



Docker-Härtung: Isolation + minimale Rechte



Merke

Container-Isolation ist nicht gleich vollständige virtuelle Maschine: Der Host-Kernel bleibt eine gemeinsame Vertrauensgrenze.

074

KUBERNETES-SICHERHEITSARCHITEKTUR UND RBAC

Kontrollen für API, Berechtigungen, Workloads, Netzwerk und Nodes.

1. KUBE-API-SERVER ABSICHERN

Der API-Server ist der zentrale Zugang zur Kubernetes-Steuerungsebene.



- Zugriff auf notwendige Identitäten und Netzpfade beschränken.
- Authentisierung und Autorisierung aktiv nutzen; RBAC folgt darauf.
- TLS schützt die Verbindung, erstazt aber keine Zugriffskontrolle.

2. ROLE-BASED ACCESS CONTROL (RBAC)

RBAC verknüpft Subjekte mit Rollen und deren Berechtigungen.



- Role und RoleBinding gelten in einem Namespace.
- ClusterRole und ClusterRoleBinding ermöglichen clusterweite Zuweisungen.
- Nur benötigte Verben und Ressourcen erlauben; Bindungen regelmäßig prüfen.

3. ADMISSION CONTROLLERS

Admission-Controller prüfen oder verändern Anfragen, nachdem Authentisierung und Autorisierung erfolgt sind und bevor Objekte gespeichert werden.



- Validierende Richtlinien können unerwünschte Konfigurationen ablehnen.
- OPA/Gatekeeper kann Richtlinien für Kubernetes-Ressourcen durchsetzen.
- Richtlinien zunächst testen und Ausnahmen eng begrenzen.

4. POD SECURITY STANDARDS (PSS)

Pod Security Standards definieren Sicherheitsstufen für Pod-Konfigurationen; die integrierte Pod Security Admission kann diese auf Namespace-Ebene anwenden.



- Baseline schränkt bekannte riskante Einstellungen ein.
- Restricted verlangt stärkere Schutzmaßnahmen für Pods.
- Labels bestimmen den Modus, etwa enforce, audit oder warn.

5. NETWORK POLICIES

NetworkPolicies legen fest, welcher Pod-Verkehr ein- oder ausgehend zugelassen ist.



- Regeln wirken nur, wenn das eingesetzte Netzwerk-Plugin NetworkPolicies unterstützt.
- Selektierte Pods werden gemäß den passenden Richtlinien eingeschränkt.
- Ingress und Egress getrennt betrachten; notwendige Verbindungen gezielt erlauben.

6. SECRETS UND ETCD

Knoten-Isolation begrenzt, welche Workloads, auf welchen Nodes laufen und wie Ressourcentyp erreimatisch verschlüsselt.



- Node-Labels und Taints/Tolerations für Platzierungsregeln Smtüssen, chützen.
- Kritische Workloads bei Bedarf auf getrennte Node-Pools verteilen.
- Zugriff auf Betriebssmages, Versionskontrottz oder Logs ablegen.

7. NODE-ISOLATION

Kubernetes-Secrets enthallen sensible Daten, aind aber nicht allein durch ihren Ressourcentyp automatisch verschlüsselt.



- Verschlüsselung ruhender Daten für etcd konfigurieren und Kchlüssel schützen.
- Zugriff auf Secrets per RBAC auf erforderliche Identitäten begrenzen.
- Geheimnisse nicht in Images, Versionskontrolle oder Logs ablegen.

8. KUBELET-HARDENING

Der Kubelet-Agent verwaltet Pods auf einem Node und benötigt eine kontrollierte Konfiguration.



- Anonyme Authentisierung deaktivieren und API-Zugriff authentisieren.
- Autorisierung nach dem Prinzip der geringsten Rechte konfigurieren.
- Kubelet-Ports und Verwaltungszugänge nicht unnötig exponieren.

075 SERVERLESS SECURITY

Funktionen, Daten, Schnittstellen und Konfiguration schützen.

1. EVENT-INJECTION-ANGRIFFE



Ereignisse und Nutzdaten sind Eingaben und müssen als nicht vertrauenswürdig behandelt werden.

- Prüfe Schema, Typen, Pflichtfelder, Größe und Wertebereiche.
- Vermeide ungeprüfte Eingaben in SQL, Shell-Befehlen, Vorlagen oder dynamischen Abfragen.
- Nutze parametrisierte Abfragen und kontextgerechte Ausgabekodierung.
- Behandle gefälschte oder wiederholte Ereignisse sicher; prüfe Signaturen, Herkunft und Idempotenz, wo passend.

2. LEAST-PRIVILEGE-FUNKTIONSROLLEN



Gib jeder Funktion nur die Identität und Rechte, die sie für ihre konkrete Aufgabe benötigt.

- Begrenze Aktionen und Ressourcen möglichst eng.
- Trenne Rollen zwischen Funktionen und Umgebungen.
- Vermeide langlebige Schlüssel im Code, nutze verwaltete Identitäten oder sichere Geheimnisdienste.
- Überprüfe Berechtigungen und Zugriffsprotokolle regelmäßig.

3. EPHEMERE SPEICHERHYGIENE



Temporärer lokaler Speicher kann zwischen Aufrufen erhalten bieben oder wiederverwendet werden. Verlasse dich nicht darauf, dass er nach jedem Aufruf automatisch geleert ist.

- Lege keine unnötigen Geheimnisse oder sensiblen Daten im temporären Speicher ab.
- Lösche temporäre Dateien und Zwischendaten, sobald sie nicht mehr gebraucht werden.
- Schütze Daten bei Ablage und Übertragung entsprechend ihrer Sensibllität.

Beachte, dass Ausführungsumgebungen je nach Plattform und Konfiguration variieren.

4. RISIKEN DURCH DRITTANBIETER-ABHÄNGIGKEITEN



Bibliohteken und Pakete erweitern die Angriffsfläche und können bekannte Schwachstellen oder unerwünschte Änderungen einführen.

- Nutze nur benötigte Abhängigkeiten und entferne ungenutzte Pakete.
- Halte Versionen nachvollziehbar fest und beziehe Pakete aus vertrauenswürdigen Quellen.
- Prüfe Abhängigkeiten auf behannte Schwachstellen und aktualisiere risikobasiert.
- Erezuge und schütze Build-Artefakte; beschränke Änderungen an der Lieferkette.

5. API-GATEWAY ABSICHERN



Ein API-Gateway kann den Zugang zu Funktionen kontrollieren, eretazt aber keine Autorisierung und Eingabeprüfüng in der Anwendung.

- Aktiviere passende Authentisierung und prüfe Berechtigungen pro Ressource und Aktion.
- Setze geeignete Größen-, Raten- und Zeitlimits; konfiguriere CORS bewusst.
- Nutze HTTPS und vermeide sensible Daten in URLs oder Logs.
- Behandle Fehler sicher und gib keine internen Details an Clients preis.

6. KALTSTART- UND TIMEOUT-DOS



Überlastung kann Kosten, Latenz oder Verfügbarkeit beeinträchtigen. Kaltstarts sind zusätzliche Startzeiten bestimmter Ausführungen; sie sind nicht automatisch ein Angriff.

- Setze angemessene Timeouts und Ressourcenlimits.
- Begrenze Aufrufraten und parallele Verarbetitung passend zum Dienst und Risiko.
- Schütze abhängige Systeme vor Lastspitzen durch Warteschlangen, Rückstaukontrolle oder Begrenzungen.
- Überwache ungewöhnliche Aufrufmuster und Kostenentwicklung.

EINORDNUNG

Konkrete Quoten, Kaltstartverhalten und Schutsfunktionen hängen vom Anbieter, der Laufzeit und der Konfiguration ab.

7. UMGEBUNGSÜBERWACHUNG



Sicherheitsrelevante Ereignisse müssen auffindbar sein, ohne dabei unnötig sensible Daten zu protokollieren.

- Erfasse Aufrufe, Fehler, Berechtigungsänderungen und relevante Konfigurationsänderungen.
- Protokolliere strukturierte, aussagekräftige Ereignisse mit Zeit und Funktionskontext.
- Schwärze oder lasse Passwörter, Tokens und personenbezogene Inhalte weg.
- Definiere Aufbewahrung, Zugriffsschutz und Alarmierung passend zum Risiko.

8. SERVERLESS-OBSERVABILITY



Observability verbindet Logs, Metriken und verteilte Traces, um Verhalten und Fehler über Funktions- und Dienstgrenzen hinweg zu verstehen.

- Verknüpfe Aufrufe mit Korrelations- oder Trace-IDs, ohne Geheimnisse darin abzulegen.
- Beobachte Fehlerquoten, Latenz, Auslastung und Abhängigkeiten.
- Nutze Alarme mit klaren Schwellen und einem definierten Reaktionsweg.
- Beschränke Zugriff auf Telemetriedaten und prüfe sie auf vertrauliche Inhalte.

MINDMAP 076: CLOUD-SPEICHER- UND OBJEKTSICHERHEIT

Cloud-Objektspeicher wie Amazon S3, Azure Blob Storage und Google Cloud Storage halten Dateien als Objekte in Containern (Buckets oder vergleichbaren Ressourcen). Schütze Vertraulichkeit, Integrität und Verfügbarkeit durch abgestimmte Zugriffsregeln, Verschlüsselung, Wiederherstellung und Überwachung.

1 · PUBLIC ACCESS BLOCKS



Begrenze unbeabsichtigte öffentliche Freigaben auf Konto- und Speicher-Ebene.

- Öffentlichen Zugriff standardmäßig blockieren; Ausnahmen eng begründen.
- Einstellungen für Konto und Bucket prüfen: zusätzliche Schutzebenen können unterschiedlich wirken.
- Regelmäßig testen, ob anonyme Anfragen tatsächlich abgewiesen werden.

2 · BUCKET-POLICIES & ACLs



Policies und Zugriffskontrollisten (ACLs) regeln, wer welche Aktionen ausführen darf.

- Identitäten und Aktionen auf das notwendige Minimum beschränken.
- Explizite Verbote und übergeordnete Organisationsregeln beachten.
- ACLs nur verwenden, wenn erforderlich; die unterstützten Funktionen unterscheiden sich je Dienst und Konfiguration.
- Freigaben auf Konto-, Container- und Objektebene nachvollziehen.

3 · SERVER-SIDE ENCRYPTION (SSE-KMS)



Bei serverseitiger Verschlüsselung verschlüsselt der Cloud-Dienst gespeicherte Objekte; bei SSE-KMS wird dafür ein Schlüsselverwaltungsdienst genutzt.

- Schlüsselzugriff getrennt von Speicherzugriff kontrollieren.
- Schlüsselrichtlinien, Berechtigungen, Rotation und Wiederherstellbarkeit berücksichtigen.
- Verschlüsselung im Rubezustand ersetzt weder sichere Zugriffsregeln noch TLS bei der Übertragung.

4 · CLIENT-SIDE ENCRYPTION



Bei clientseitiger Verschlüsselung werden Daten vor dem Upload auf der kontrollierten Client-Seite verschlüsselt.

- Schlüssel sicher erzeugen, speichern, verteilen und sichern.
- Schlüsselverlust kann Daten unlesbar machen; unbefugter Schlüsselzugriff kann den Schutz aufheben.
- Metadaten und Anwendungsabläufe können weiterhin Informationen offenlegen.

5 · OBJECT-LOCKING & VERSIONIERUNG



Versionierung bewahrt mehrere Objektstände; Object Lock kann Versionen gegen Änderung oder Löschung schützen.

- Aufbewahrungsfristen und Löschschutz passend zum Schutzbedarf festlegen.
- Governance- und Compliance-Modi unterscheiden sich; stärkere Sperren können auch Administratoren einschränken.
- Versionierung allein verhindert nicht jede Löschung und ist kein vollständiges Backup.

Wichtig
Objektsperren und Versionierung dienstspezifisch konfigurieren und die Wiederherstellung praktisch testen.

6 · ZUGRIFFSPROTOKOLLIERUNG



Protokolle machen Zugriffe und Änderungen untersuchbar.

- Datenzugriffsereignisse aktivieren, soweit der Dienst sie separat konfiguriert.
- Protokolle gegen Änderung und unbefugte Einsicht schützen.
- Aufbewahrung, Alarmierung und Zuständigkeiten festlegen; Protokollierung kann Kosten verursachen.

7 · RANSOMWARE-SCHUTZ



Erschwere Verschlüsselung, Manipulation oder Löschung durch kompromittierte Konten.

- Getrennte, möglichst unveränderbare Wiederherstellungskopien vorsehen.
- Schreib-, Lösch- und Schlüsselrechte minimieren und getrennt vergeben.
- Versionierung und Object Lock ergänzen Backups, ersetzen aber keinen getesteten Wiederherstellungsplan.

8 · EXFILTRATION PREVENTION



Erkenne und begrenze unautorisierte Datenabflüsse aus sensiblen Speicherbereichen.

- Daten klassifizieren und Zugriff auf erforderliche Identitäten und Netzwerke beschränken.
- Ungewöhnliche Downloads, öffentliche Freigaben und Richtlinienänderungen überwachen.
- Datenminimierung, Warnmeldungen und ein Reaktionsverfahren kombinieren.

077 CLOUD KEY MANAGEMENT SERVICES

Schlüssel schützen, Nutzung kontrollieren, Zugriffe nachvollziehen.

ZENTRALE IDEE



Cloud-Dienste für Schlüsselverwaltung schützen kryptografische Schlüssel und unterstützen deren kontrollierte Nutzung. Schlüssel sind nicht dasselbe wie gespeicherte Passwörter oder API-Tokens: Für diese Zugangsdaten sind Secrets Manager vorgesehen. Die konkreten Funktionen unterschieden sich je nach Anbieter und Konfiguration.

- Ziel: Vertraulichkeit, kontrollierter Zugriff und nachvollziehbare Schlüsselverwendung.

1. HARDWARE SECURITY MODULES (CLOUD HSM)



Ein HSM ist eine speziell geschützte Hardware zur Erzeugung und Verabeitung kryptografischer Schlüssel. Cloud-HSM-Angebote stellen solche Funktionen als verwalteten Dienst bereit.

- Schlüsseloperationen können im HSM stattfinden, ohne den Schlüssel im Klartext an die Anwendung herauszugeben.
- Zugriff, Verfügbarkeit und Betriebsmodell hängen vom jeweiligen Dienst ab.

2. KUNDENVERWALTETE SCHLÜSSEL (CMK)



Kundenverwaltete Schlüssel geben einer Organisation mehr Kontrolle über Schlüsselrichtlinien und Lebenszyklus als rein dienstverwaltete Schlüssel. Die genaue Kontrolle variiert nach Anbieter und Dienst.

- Konfiguration und Berechtigungen sorgfältig begrenzen.
- Schlüssel deaktivieren oder löschen kann verschlüsselte Daten unzugänglich machen.

3. ENVELOPE ENCRYPTION



Bei der Envelope-Verschlüsselung verschlüsselt ein Datenschlüssel die Daten. Ein übergeordneter Schlüssel schützt anschließend den Datenschlüssel.

- Datenverschlüsselung: Datenschlüssel verschlüsselt Inhalt.
- Schlüsselschutz: KMS schützt oder verwaltet den Datenschlüssel.
- Anwendung und Dienst müssen den Ablauf sicher koordinieren.

4. AUTOMATISIERTE SCHLÜSSELROTATION



Rotation ersetzt einen Schlüssel planmäßig durch eine neue Schlüsselversion. Wie Rotation umgesetzt wird und welche Schlüsselarten sie unterstützt, ist dienstabhängig.

- Rotation ist kein Ersatz für Zugriffskontrollen oder sichere Speicherung.
- Anwendungen müssen mit Schlüsselversionen und bestehenden Daten kompatibel sein.
- Zeitplan und Auswirkungen vor Aktivierung prüfen.

5. BRING YOUR OWN KEY (BYOK)



BYOK bezeichnet typischerweise das Einbringen eines extern erzeugten Schlüssels in einen Cloud-Schlüsseldienst. Danach gelten die Möglichkeiten und Grenzen des jeweiligen Anbieters.

- Ein Import bedeutet nicht automatisch, dass der Schlüssel dauerhaft außerhalb der Cloud bleibt.
- Importverfahren, Exportierbarkeit und Schutzmechanismen unterscheiden sich.

6. GRANULARE RICHTLINIEN FÜR SCHLÜSSELZUGRIFF



Schlüsselrichtlinien und Identitäts- sowie Zugriffsregeln legen fest, wer Schlüsseloperationen ausführen darf. Rechte sollten nach dem Prinzip der geringsten Berechtigung vergeben werden.

- Rollen und Anwendungen nur notwendige Aktionen erlauben.
- Schlüsselverwaltung und Schlüsselverwendung möglichst getrennt betrachten.
- Änderungen an Richtlinien kontrolliert verwalten.
- Änderungen automatisiert überwachen.

7. AUDITIERUNG VON ENTSCHLÜSSELUNGSAUFRUFEN



Auditprotokolle können festhalten, wann und durch welche Identität eine Schlüsseloperation angefordert wurde. Protokollumfang und Verfügbarkeit hängen von Dienst und Einrichtung ab.

- Protokolle zentral schützen und regelmäßig auswerten.
- Fehlgeschlagene und ungewöhnliche Aufrufe beachten.
- Ein Protokolleintrag belegt nicht allein, dass die zugrunde liegende Nutzung legitim war.

8. SECRETS-MANAGER-INTEGRATION



Secrets Manager speichern und verteilen Zugangsdaten wie Passwörter, Tokens oder Datenbank-Anmeldedaten. Ein KMS kann je nach Dienst und Konfiguration Schlüssel zum Schutz gespeicherter Secrets bereitstellen.

- Geheimnisse nicht in Quellcode oder öffentlichen Konfigurationsdateien ablegen.
- Zugriffe begrenzen und erforderliche Erneuerung sicher planen.
- KMS-Schlüsselverwaltung und Secret-Speicherung sind verwandte, aber unterschiedliche Aufgaben.

078 CLOUD THREAT DETECTION

Cloud-Telemetrie auswerten, verdächtige Aktivitäten korrelieren und angemessen reagieren. GuardDuty und Microsoft Sentinel sind unterschiedliche Sicherheitsdienste; ihre Funktionen hängen von Konfiguration, Datenquellen und Berechtigungen ab.

- Erkennung ist kein Beweis für einen erfolgreichen Angriff.
- Protokolle, Kontext und Folgeereignisse gemeinsam betrachten.

1. CLOUDTRAIL- UND SIGN-IN-ANALYSE

Kontroll- und Anmeldeereignisse auf ungewöhnliche Muster prüfen.



- Akteur, Zeit, Quell-IP, Region und Aktion vergleichen.
- Fehlgeschlagene und erfolgreiche Anmeldungen in Folge beachten.
- Änderungen an Rollen, Richtlinien und Protokollierung einordnen.

2. VERHALTENSBASIERTES THREAT HUNTING

Hypothesen mit Zeitreihen und Kontextdaten untersuchen, statt nur bekannte Signaturen zu suchen.



- Abweichungen von üblichen Zeiten, Orten oder Ressourcen ermitelen.
- Zusammengehörige Ereignisse über Identität und Ressource verbinden.
- Normale Änderungen und geplante Automatisierung als Alternativen prüfen.

3. KOMPROMITTIERTE CLOUD-CREDENTIALS

Hinweise auf missbrauchte Zugangsdaten anhand mehrerer Signale bewerten.



- Ungewohnte Quell-IP, Region, Geräte- oder API-Muster prüfen.
- Neue Schlüssel, Rollenübernahmen oder ungewöhnliche Berechtigungen untersuchen.
- Bei begründetem Verdacht Zugang widerrufen oder rotieren und Sitzungen beenden.

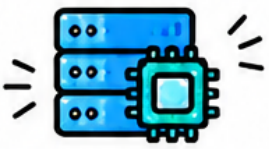
SICHER HANDELN

Untersuchungen und Gegenmaßnahmen nur innerhalb der eigenen Zuständigkeit und der geltenden Reaktionsverfahren durchführen.

4. KRYPTO-MINING-INDIKATOREN IN INSTANZEN

Mögliche Ressourcennutzung zum Mining anhand von Host- und Cloud-Signalen untersuchen.

- Anhaltend hohe CPU- oder GPU-Auslastung mit Prozess- und Netzwerkdaten abgleichen.
- Unbekannte Prozesse, ausgehende Verbindungen und unerwartete Instanzstarts prüfen.
- Hohe Auslastung allein ist kein Beleg; legitime Lastspitzen ausschließen.



5. VPC-FLOW-LOG-AUSWERTUNG

Flow Logs liefern Metadaten zu IP-Verkehr, keine vollständigen Paket-Inhalte.

- Quelle, Ziel, Port, Protokoll, Zeitfenster und Ergebnis betrachten.
- Unerwartete ausgehende Ziele oder Verbindungsvolumen priorisieren.
- Regeln, Netzwerkarchitektur und bekannte Dienste als Kontext heranziehen.



GRENZE DER DATEN

Flow-Logs seigen Verkehrs-Metadaten; sie erklären nicht allein, welche Anwendungsdaten übertragen wurden oder ob eine Verbindung bösartig war.

6. BEDROHUNGSKORRELATION VIA SENTINEL

Verbundenen Datenquellen gemeinsame Zeit-, Identitäts- und Ressourcenbezüge entnehmen.

- Cloud- und Identitätsereignisse mit Alarmen und Asset-Kontext verknüpfen.
- Datenquellen, Zeitstempel und mögliche Verzögerungen berücksichtigen.
- Doppelte oder abhängige Alarme bündeln; Rohereignisse für die Prüfung bewahren.



7. ANOMALIEN BEI DATENEXPORTEN

Ungewöhnliche Datenbewegungen anhand von Umfang, Ziel und Berechtigung bewerten.

- Volumen, Zeitpunkt, Zielkonto und verwendeten Dienst vergleichen.
- Neue Exportwege oder öffentliche Freigaben untersuchen.
- Backups, reguläre Berichte und genehmigte Migrationen ausschließen.



8. ALARMIERUNGS-PLAYBOOKS

Ein Playbook unterstützt eine nachvollziehbare, abgestufte Reaktion auf bestätigte oder wahrscheinliche Vorfälle.



- Alarm triagieren: Schweregrad, Umfang und betroffene Ressourcen erfassen.
- Belege sichern und zuständige Personen nach Verfahren benachrichtigen.
- Genehmigte Eindämmung umsetzen, etwa Schlüssel widerrufen oder Ressource isolieren.
- Maßnahmen, Ergebnisse und offene Fragen dokumentieren.

MERKSATZ

Erkennen → Kontext prüfen → korrelieren → abgestimmt reagieren. Ein Alarm ist ein Untersuchungsanlass, nicht automatisch ein bestätigter Vorfall.

BEGRIFFE

CloudTrail protokolliert Aktivitäten in AWS-Konten. AWS GuardDuty analysiert ausgewählte AWS-Telemetrie auf Bedrohungen. Microsoft Sentinel ist eine cloudnative SIEM-Plattform, die Daten aus verbundenen Quellen analysiert und korreliert.



079 · OT- UND SCADA-SICHERHEIT

Operative Technologie (OT) überwacht oder steuert physische Prozesse. SCADA-Systeme erfassen und überwachen Prozesse oft über räumlich verteilte Anlagen, Sicherheit muss Cyberrisiken und mögliche Auswirkungen auf Menschen, Umwelt und Betrieb gemeinsam berücksichtigen.

- OT: Steuerungs- und Automatisierungstechnik in der physischen Welt.
- SCADA: Systeme zur Überwachung und Steuerung verteilter Prozesse.
- Ziel: sichere, zuverlässige und kontrollierte Abläufe.

1 · PURDUE-MODELL

Das Purdue-Modell ordnet industrielle Systeme in Ebenen ein. Es ist ein Orientierungsmodell, keine universelle Bauvorschrift; reale Anlagen können davon abweichen.



- Ebene 0: physischer Prozess, Sensoren und Aktoren.
- Ebene 1: Steuerungen, etwa SPS/PLC.
- Ebene 2: Überwachung und Bedienung, etwa HMI.
- Ebene 3: Produktionsbetrieb und standortnahe Systeme.
- Ebene 4/5: Unternehmens-IT und externe Netze.
- Zonen und kontrollierte Übergänge helfen, Risiken zu begrenzen.

2 · SCADA UND SPS/PLC

Industrielle Steuerungssysteme bestehen aus Geräten und Software, die Prozesse messen, anzeigen und steuern. Komponenten und Rollen unterscheiden sich je nach Anlage.



- Sensoren messen Werte wie Druck oder Temperatur.
- Aktoren verändern den Prozess, etwa über Ventile oder Motoren.
- SPS/PLC führt Steuerungslogik nahe am Prozess aus.
- HMI zeigt Zustände an und ermöglicht Bedienung.
- SCADA sammelt Daten und unterstützt Überwachung und Steuerung.
- Engineering-Stationen konfigurieren oder programmieren Steuerungen.

3 · INDUSTRIEPROTOKOLLE

Protokolle ermöglichen den Austausch zwischen Geräten und Systemen. Ältere oder ungeschützt eingesetzte Varianten bieten oft keine ausreichende Authentisierung oder Verschlüsselung; Schutz hängt von konkreter Version, Konfiguration und Umgebung ab.



- Modbus: verbreitetes Industrieprotokoll; Modbus/TCP allein garantiert keine Authentisierung oder Verschlüsselung.
- DNP3: für Versorgungs- und Steuerungsumgebungen; Sicherheitsfunktionen sind nicht in jeder Installation aktiv.
- PROFINET: industrielles Ethernet für Automatisierung. Schutzbedarf und Funktionen unterscheiden sich nach Einsatz.
- Unnötige Dienste und Verbindungen abschalten oder beschränken.
- Kommunikation segmentieren und erlaubte Verbindungen überwachen.

Wichtig
Ein Protokollname allein sagt nicht aus, ob eine konkrete Anlage sicher konfiguriert ist.

4 · KONVERGENZ VON IT UND OT

IT und OT sind zunehmend verbunden, etwa für Fernwartung, Auswertung oder zentrale Verwaltung. Verbindungen können Effizienz schaffen, aber auch Wege zwischen Netzen eröffnen.

- IT priorisiert häufig Vertraulichkeit und Datenverarbeitung.
- OT muss zusätzlich Prozesskontinuität und physische Folgen beachten.
- Übergänge inventarisieren und auf notwendige Kommunikation beschränken.
- Fernzugriff nur kontrolliert, zeitlich begrenzt und nachvollziehbar ermöglichen.
- Änderungen mit Betrieb und Sicherheitsverantwortlichen abstimmen.



5 · AIR-GAP-MYTHEN

Ein Air Gap bezeichnet die beabsichtigte physische Trennung eines Netzes. Diese Trennung senkt bestimmte Risiken, ist aber kein Beweis für vollständige Sicherheit.



- Wechselmedien können Daten und Schadsoftware übertragen.
- Wartung, temporäre Verbindungen oder Fehlkonfigurationen können Trennungen umgehen.
- Zugänge und Medien kontrollieren: Änderungen dokumentieren.
- Auch isolierte Systeme bentöigen Inventarisierung, Schutz und Wiederherstellungsplanung.
- Sicherheitsannahmen regelmäßig anhand der tatsächlichen Verbindungen prüfen.

Merksatz
„Isoliert" bedeutet nicht automatisch „unangreifbar".

6 · PHYSISCHE SICHERHEITSRISIKEN

Cyber- und physische Sicherheit hängen in OT-Umgebungen zusammen. Zugang zu Geräten oder Leitungen kann Sicherheitsmaßnahmen umgehen oder Betriebsabläufe stören.

- Schaltschränke, Leitstände und Serverräume angemessen zugangbeschränken.
- Unbeaufsichtigte Anschlüsse und Wartungsports berücksichtigen.
- Zutritte und Wartungsarbeiten nachvollziehbar dokumentieren.
- Geräte vor Manipulation, Beschädigung und Umwelteinflüssen schützen.
- Physische Schutzmaßnahmen mit Betriebs- und Notfallplänen abstimmen.



7 · SAFETY UND SECURITY

Safety schützt Menschen und Umwelt vor Gefahren durch Prozesse. Security schützt Systeme und Informationen vor unbefugten oder schädlichen Einwirkungen. Beide Ziele müssen gemeinsam betrachtet werden.

- Eine Security-Maßnahme darf nicht unbedacht eine Schutzfunktion beeinträchtigen.
- Verfügbarkeit kann in OT unmittelbar sicherheitsrelevant sein.
- Risiken anhand möglicher Folgen für Prozess, Menschen und Umwelt bewerten.
- Bei Vorfällen abgestimmte Sicherheits- und Betriebsverfahren nutzen.
- Änderungen kontrolliert testen und freigeben.



Abwägung
Sicherheit und Betrieb nicht gegeneinander ausspielen: Maßnahmen müssen zum Prozessrisiko passen.

8 · LEGACY UND PATCHING

Industrielle Anlagen enthalten mitunter ältere Systeme, die nicht ohne Weiteres aktualisiert werden können. Patchen kann Kompatibilltät, Verfügbarkeit oder Zulassung beeinflussen und erfordert deshalb kontrollierte Planung.

- Systeme, Versionen, Abhängigkeiten und Verantwortliche erfassen.
- Updates anhand von Risiko, Herstellerhinweisen und Betriebsanforderungen priorisieren.
- Vor Änderungen Test, Wartungsfenster und Rückfallplan vorsehen.
- Wenn ein Patch nicht zeitnah möglich ist: Exposition senken, Zugriffe begrenzen und überwachen.
- Entscheidung, Begründung und nächste Überprüfung dokumentieren.



080 INDUSTRIELLE NETZWERKSICHERHEIT

Industrielle Netzwerksicherheit schützt Steuerungs- und Produktionssysteme. Zonierung trennt Systeme nach Funktion und Schutzbedarf; kontrollierte Verbindungen begrenzen, was zwischen ihnen kommunizieren darf.

- Ziel: Risiken und Auswirkungen von Vorfällen begrenzen.
- Sicherheit muss Verfügbarkeit und Prozesssicherheit berücksichtigen.
- Maßnahmen passend zur Anlage planen und im Betrieb überwachen.

1. ZONEN- UND CONDUIT-KONZEPT

Eine Zone fasst Systeme mit ähnlichen Sicherheitsanforderungen zusammen. Ein Conduit ist der kontrollierte Kommunikationsweg zwischen Zonen.

- Zonen nach Funktion, Risiko und Schutzbedarf abgrenzen.
- Kommunikation zwischen Zonen auf notwendige Flüsse beschränken.
- Regeln dokumentieren und Änderungen kontrolliert prüßen.

2. INDUSTRIELLE DMZ (iDMZ)

Eine Induistrielle demilitarisierte Zone vermittelt zwischen Unternehmens-IT und Produktionsnetz. Sie bietet einen kontrollierten Ort für ausgewählte Dienste und Datenübergaben.

- Verbindungen aus IT und OT nicht pauschal durchleiten.
- Benötigte Dienste gezielt freigeben und protokollieren.
- iDMZ-Systeme selbst absichern und aktuell halten.

3. DEEP PACKET INSPECTION FÜR OT-PROTOKOLLE

Deep Packet Inspection (DPI) untersucht neben Netzwerkadressen auch Protokollinhalte. Bei OT-Protokollen kann dies helfen, unpassende oder unerwartete Befehle zu erkennen.

- Erkennung hängt von Protokollkenntnis und Konfiguration ab.
- Verschlüsselung kann die Inhaltsprüfung einschränken.
- Prüfregeln zunächst risikoarm testen: Fehlalarme oder Blockaden können den Prozess beeinträchtigen.

4. UNIDIREKTIONALE GATEWAYS (DATENDIODEN)

Eine Datendiode ist auf einen gerichteten Informationsfluss ausgelegt. Sie kann Rückkommunikation über diesen Pfad verhindern, erstetzt aber nicht den Schutz der verbundenen Systeme.

- Richtung und erlaubte Datenflüsse passend zum Prozess festlegen.
- Datenübergabe und mögliche Verzögerungen berücksichtigen.
- Einwegtechnik ist keine universelle Lösung für Kommunikationsbedarf in beide Richtungen.

5. FERNWARTUNGSABSICHERUNG

Fernwartung ermöglicht autorisierten Zugriff auf Anlagen aus der Ferne. Der Zugriff sollte begrenzt, nachvollziehbar und nur bei Bedarf verfügbar sein.

- Identitäten individuell vergeben; Mehr-Faktor-Authentisieurng erwägen.
- Zugriff zeitlich und auf erforderliche Systeme begrenzen.
- Sitzungen überwachen und Aktivitäten protokollieren.
- Freigabe- und Notfallverfahren vorab festlegen.

SICHERHEITSPRINZIP

Fernzugriff nicht direkt und unkontrolliert aus dem Internet auf Steuerungssysteme freigeben.

6. OT-ASSET-DISCOVERY

Asset-Discovery erfasst Geräte, Software und Kommunikationsbeziehungen im OT-Netz. Ein aktuelles Inventar unterstützt Risikoanalyse und Änderungsplanung.

- Geräte, Rollen, Standorte und Verbindungen dokumentieren.
- Aktive Erkennung kann Systeme belasten; Methode an Anlage und Herstellerhinweise anpassen.
- Ergebnisse mit Anlagenverantwortlichen abgleichen und Änderungen nachverfolgen.

7. INTEGRITÄT VON SPS-FIRMWARE

Firmware ist Software, die ein Gerät grundlegend steuert. Integritätssicherung soll unautorisierte Änderungen erkennbar machen und vertrauenswürdige Wiederherstellung ermöglichen.

- Freigegebene Versionen und Änderungen nachvollziehbar dokumentieren.
- Signatur- oder Prüfsummenverfahren nutzen, sofern Gerät und Prozess sie unterstützen.
- Gesicherte Konfigurationen und Wiederherstellung praktisch planen.

8. ANOMALIEERKENNUNG IM ANLAGENNETZ

Anomalieerkennung sucht nach Abweichungen vom erwarteten Kommunikations- oder Prozessverhalten. Ein auffälliger Messwert ist ein Hinweis zur Untersuchung, nicht automatisch ein Angriff.

- Basisverhalten und relevante Alarmgrenzen erfassen.
- Alarme mit Anlagenkontext und Änderungen abgleichen.
- Erkennungsregeln regelmäßig anpassen; Fehlalarme und übersehene Ereignisse sind möglich.
- Reaktion so planen, dass Sicherheit und Verfügbarkeit des Prozesses berücksichtigt werden.

BEGRIFFE

OT (Operational Technology) umfasst Systeme zur Überwachung oder Steuerung physischer Prozesse. SPS bedeutet Speicherprogrammierbare Steuerung.

081
ICS/SCADA-PROTOKOLLE UND SCHWACHSTELLEN

Industrielle Kommunikation verstehen und absichern.

ICS/SCADA-PROTOKOLLE UND SCHWACHSTELLEN



Industrielle Steuerungs- und Überwachungssysteme verbinden Leitstellen, Steuerungen und Feldgeräte. Ihre Protokolle transportieren Messwerte und Befehle; Sicherheit hängt auch von Netzsegmentierung, Gerätekonfiguration und Betriebsprozessen ab.

- ICS: Sammelbegriff für industrielle Steuerungssysteme; SCADA bezeichnet Systeme zur übergeordneten Überwachung und Steuerung.
- Protokollmerkmale sind nicht mit der Sicherheit der gesamten Anlage gleichzusetzen.

SICHERHEITSGRENZE

Nur in autorisierten Laboren oder ausdrücklich freigegebenen Umgebungen analysieren. Produktionsanlagen können sicherheitskritische Prozesse steuern.

1. MODBUS TCP/RTU



Modbus ist ein Anfrage-Antwort-Protokoll für industrielle Kommunikation. Die jeweilige Übertragung und Umgebung bestimmen, welche Schutzmechanismen verfügbar sind.

- Modbus TCP kapselt Modbus-Nachrichten in TCP/IP; üblich ist TCP-Port 502.
- Modbus RTU überträgt Nachrichten typischerweise seriell; Geräteadresse und Prüfsumme unterstützen die Übertragung, sind aber keine Authentisierung oder Verschlüsselung.
- Klassische Modbus-Varianten bieten keinen eingebauten Schutz der Nutzdaten vor Mitlesen oder Manipulation.
- Risiko: Unbefugte Befehle oder ausgelesene Prozessdaten, wenn Zugriff und Netzwerk nicht ausreichend geschützt sind.

2. DNP3



DNP3 wird unter anderem in Versorgungs- und Fernwirkanwendungen eingesetzt. Seine Sicherheitslage hängt von Protokollvariante, Implementierung und Konfiguration ab.

- DNP3 unterstützt strukturierte Kommunikation zwischen Stationen und Leitstellen, einschließlich Ereignis- und Messwertübertragung.
- Die Basisspezifikation stellt nicht automatisch eine starke Authentisierung für jede Kommunikation bereit.
- Secure Authentication ist eine ergänzende Sicherheitsfunktion; Unterstützung und Aktivierung müssen für die beteiligten Geräte geprüft werden.
- Risiko: Unzureichend geschützte Verbindungen können unbefugte Abfragen oder Steuerbefehle begünstigen.

3. OPC UA



OPC UA ist eine Plattform für industrielle Interoperabilität mit definierten Sicherheitsmechanismen. Ihre Wirksamkeit setzt passende Konfiguration und vertrauenswürdige Identitäten voraus.

- Sicherheitsrichtlinien und Sicherheitsmodi bestimmen, welche Schutzfunktionen eine Verbindung verwandet.
- Zertifikate können Anwendungen identifizieren und zur Vertrauensprüfung dienen; Gültigkeit, Aussteller und Vertrauenskette sind relevant.
- Verschlüsselung schützt Vertraulichkeit; Signaturen hellen, Integrität und Herkunft zu prüfen.
- Risiko: Abgelaufene, falsch zugeordnete oder ungeprüft akzeptierte Zertifikate sowie unsichere Richtlinien können Schutz untergraben.

4. BACnet



BACnet dient der Kommunikation in Gebäudeautomationssystemen, etwa zwischen Heizungs-, Lüftungs-, Klima- und Management-Komponenten.

- BACnet/IP transportiert BACnet über IP-Netze; BACnet kann auch andere Link-Techniken nutzen.
- Nicht jede BACnet-Installation untzt BACnet/SC oder vergleichbare Schutzmaßnahmen.
- Risiken entstehen durch erreichbare Dienste, Standard- oder schwache Zugangsdaten und unzureichend getrennte Gebäudenetze.
- Auswirkungen können Verfügbarkeit und Steuerung von Gebäudefunktionen betreffen.

5. EtherNet/IP und CIP



EtherNet/IP ist ein industrielles Kommunikationsnetz, das das Common Industrial Protocol (CIP) verwendet. Ethernet im Namen bebeutet nicht automatisch Internetzgagung.

- CIP definiert unter anderem Objekte und Dienste für industrielle Gerätekommunikation.
- Explizite Nachrichten dienen typischerweise Konfiguration und Abfragen; implizite Echtzeit-I/O-Daten folgen anderen Kommunikationsmustern.
- Schutzfunktionen und Verhalten variieren je nach Gerät, CIP-Erweiterung und Netzdesign.
- Risiko: Unkontrollierter Zugriff auf Dienste oder Geräte kann Konfiguration und Prozesskommunikation beeinträchtigen.

6. IEC 60870-5-104



IEC 60870-5-104 überträgt Fernwirkkommuni[illegible] dass [illegible] über TCP/IP und wird unter anderem in Energieversorgungsnetzen eingesetzt.

- Das Protokoll bildet Anwendungsfunktionen der Fernwirktechnik auf eine TCP/IP-Kommunikation ab.
- Die klassische Protokollvariante allein gaarntiet keine moderne Ende-zu-Ende-Verschlüsselung oder starke Authentisierung.
- Ergänzende Sicherheitsmechanismen und Netzschutz müssen passend zur eingestetten Lösung umgesetzt werden.
- Risiken: Unbefugter Zugriff, manipulierte Kommunikation oder Störungen der Erreichbarkeit.

7. MITM UND INJECTION



Ein Man-in-the-Middle-Angriff setzt voraus, dass ein Angreifer Kommunikation abfangen oder beeinflussen kann. Injection bezeichnet das Einschleusen unerwünschter Protokollinhalte oder Befehle.

- Mögliche Voraussetzungen: Zugang zum Kommunikationspfad, kompromittierter Netzwerkknoten oder schwache Vertrauens- und Zugriffskontrollen.
- Unverschlüsselte und nicht authentisierte Kommunikation ist besonders anfällig für Mitlesen und Manipulation.
- Mögliche Folgen reichen von falschen Messwerten bis zu unerwarteten Steuerbefehlen; Auswirkungen hängen vom Prozess und den Schutzmaßnahmen ab.
- Überwachung, Segmentierung und authentisierte, integritätsgeschützte Kommunikation erschweren Angriffe.

MERKSATZ

Ein Protokoll mit Sicherheitsfunktionen ist nicht automatisch sicher konfiguriert; ein älteres Protokoll ist nicht zwangsläufig ungeschützt, wenn zusätzliche Kontrollen wirksam greifen.

082 · IoT-SICHERHEITSARCHITEKTUR

IoT-Geräte erfassen, verarbeiten und übertragen Daten. Ihre Sicherheit hängt vom Zusammenspiel aus Gerät, Firmware, Verbindung, Updateweg und Cloud-Dienst ab.

- Sicherheitsziel: Vertraulichkeit, Integrität und Verfügbarkeit schützen.
- Prüfungen nur an eigenen Geräten oder mit ausdrücklicher Erlaubnis durchführen.

1. IoT-ARCHITEKTURSCHICHTEN

Daten wandern häufig vom physischen Gerät über lokale Verarbeitung und Netzwerkvermittlung zu einem entfernten Dienst.

- Sensor/Aktor: misst Zustände oder führt Befehle aus.
- Edge: verarbeitet Daten nahe am Gerät.
- Gateway: vermittelt zwischen Geräten und Netzwerken.
- Cloud: speichert oder analysiert Daten; nicht jedes IoT-System benötigt alle Schichten.

2. ZUGANGSDATEN UND SCHNITTSTELLEN

Werkseitige oder wiederverwendete Zugangsdaten und unnötig erreichbare Dienste erhöhen das Risiko unbefugter Zugriffe.

- Einzigartige, starke Zugangsdaten setzen und Standardpasswörter ändern.
- Nicht benötigte Dienste und Schnittstellen deaktivieren.
- Zugriff auf Verwaltungsoberflächen auf vertrauenswürdige Netze beschränken.
- Keine Zugangsdaten in Quellcode, Protokollen oder öffentlich erreichbaren Speicherorten hinterlegen.

3. FIRMWARE-ANALYSE UND EXTRAKTION

Firmware ist die Gerätesoftware. Eine Analyse kann Aufbau, Komponenten und mögliche Fehlkonfigurationen sichtbar machen.

- Binwalk kann Firmware-Dateien auf eingebettete Dateisysteme und bekannte Signaturen untersuchen.
- Ghidra unterstützt die statische Analyse von Binärdateien; Ergebnisse erfordern fachliche Interpretation.
- Nur rechtmäßig erhaltene Firmware untersuchen und sensilbe Inhalte wie Schlüssel nicht weitergeben.
- Ein Fund ist zunächst ein Hinweis, nicht automatisch der Nachweis einer ausnutzbaren Schwachstelle.

4. HARDWARE-SCHNITTSTELLEN

UART, JTAG und SWD dienen häufig Entwicklung, Diagnose oder Debugging. Zugängliche Debug-Punkte können Informationen preisgeben oder Änderungen ermöglichen.

- UART ist oft eine serielle Kommunikationsschnittstelle.
- JTAG und SWD ermöglichen je nach Chip Debug- und Testfunktionen.
- Debug-Zugänge im fertigen Produkt deaktivieren oder wirksam absichern.
- Untersuchungen an Hardware nur mit Erlaubnis und geeigneter elektrischer Vorsicht durchführen.

5. DRAHTLOSPROTOKOLLE

IoT nutzt unterschiedliche Funkverfahren. Reichweite, Geräteeigenschaften und Sicherheitsfunktionen variieren je nach Protokoll und Implementierung.

- Bluetooth Low Energy (BLE) ist für energiearme Kurzstreckenkommunikation ausgelegt.
- Zigbee und Z-Wave werden häufig in Heimautomatisierung eingesetzt; konkrete Sicherheitsmerkmale hängen von Version, Konfiguration und Gerät ab.
- Kopplung und Aufnahme ins Netz kontrollieren; unbekannte Geräte nicht ungeprüft zulassen.
- Verschlüsselung schützt Datenübertragung, ersetzt aber keine sichere Geräteverwaltung.

6. OTA-UPDATES

Over-the-Air-Updates verteilen Firmware über eine Netzwerkverbindung. Ein sicherer Updateprozess schützt vor manipulierten oder fehlerhaften Installationen.

- Updatepakete kryptografisch signieren und die Signatur vor Installation prüfen.
- Übertragung schützen; Transportverschlüsselung allein beweist nicht die Echtheit des Pakets.
- Versionen und Rücksetzungen kontrollieren, um Downgrade-Risiken zu begrenzen.
- Fehlerbehandlung und Wiederherstellung einplanen, damit ein Update das Gerät nicht unbrauchbar macht.

7. ZERO TRUST FÜR IoT

Zero Trust bedeutet, Zugriffe nicht allein unfgrund eines Netzwerkstandorts als vertrauenswürdig anzunehmen. Jede Anfrage wird anhand von Identität, Kontext und Berechtigung peprüft.

- Geräte eindeutig identifizieren und authentisieren.
- Nur notwendige Berechtigungen vergeben und Zugriffe begrenzen.
- Netze segmentieren, damit kompromittierte Geräte nicht ungehindert weitere Systeme erreichen.
- Gerätezustand und Zugriffe überwachen; Vertrauen bei Änderungen erneut bewerten.

8. BOTNETZE UND DDoS-ABWEHR

Ein Botnet ist eine Gruppe kompromittierter Geräte, die ferngesteuert gemeinsam handeln kann. Mirai ist bekannt dafür, IoT-Geräte unter anderem durch schwache oder voreingestellte Zugangsdaten zu kompromittieren und für DDoS-Angriffe zu missbrauchen.

- DDoS überlastet Dienste oder Netze durch eine große Menge verteilter Anfragen oder Daten.
- Standardzugänge ändern, Firmware aktuell halten und unnötige externe Erreichbarkeit reduzieren.
- Netzwerkverkefr beobachten und ungewöhnliche Verbindungen oder Lastspitzen untersuchen.
- Abwehr kann Filterung, Ratenbegrenzung, Segmentierung und Unterstützung durch den Netzbetreiber umfassen.

EINORDNUNG

Diese Lernmindmap erklärt Sicherheitskonzepte. Sie ist keine Anleitung zum Zugriff auf fremde Geräte.

AUTOMOTIVE CYBERSECURITY

CAN-Bus-Sicherheit

ZENTRALER ÜBERBLICK



Fahrzeug-Cybersicherheit schützt elektronische Steuergeräte, ihre Kommunikation und vernetzte Funktionen vor Manipulation, Missbrauch und Ausfall. Sicherheit entsteht durch mehrere aufeinander abgestimmte Maßnahmen über den gesemten Entwicklungs- und Betriebslebenszyklus.

- Bedrohungen verstehen und Risiken bewerten
- Kommunikation und Zugänge absichern
- Angriffe erkennen und angemessen reagieren

1. CAN-Bus-Topologie



Der CAN-Bus verbindet Steuergeräte (ECUs) über einen gemeinsamen seriellen Kommunikationsbus. Nachrichten werden anhand einer Kennung priorisiert und verteilt; die Kennung ist kein Absendernachweis.

- Busleitung mit zwei Enden; typischerweise Abschlusswiderstände an beiden Enden
- Nachrichten enthalten Kennung, Steuerinformationen und Datenfeld
- Alle angeschlossenen Knoten können Busnachrichten empfangen
- Physischer Zugang kann je nach Fahrzeugarchitektur mehrere Steuergeräte betreffen

2. FEHLENDE AUTHENTIFIZIERUNG IM KLASSISCHEN CAN



Das klassische CAN-Protokoll bietet selbst keine kryptografische Absenderauthentifizierung oder Verschlüsselung. Ein empfangendes Steuergerät kann deher nicht allein anhand des CAN-Frames nachweisen, welches Steuergerät ihn gesendet hat.

- Kennung beschreibt den Nachrichtentyp und die Priorität, nicht zuverlässig den Urheber
- Manipulierte oder gefälschte Frames sind bei Zugriff auf den Bus ein mögliches Risiko
- Schutz erfordert zusätzliche Maßnahmen auf System- oder Anwendungsebene
- Segmentierung und kontrollierte Gateways begrenzen erreichbare Busbereiche

Einordnung
Das Risiko hängt von Architektur, Zugangsmöglichkeiten und zusäzilichen Schutzmäßnahmen des konkreten Fahrzeugs ab.

3. AUTOMOTIVE ETHERNET UND BROADR-REACH



Automotive Ethernet ermöglicht leistungsfähige Fahrzeugnetzwerke über Ethernet-Technik. BroadR-Reach, heute häufig als 100BASE-T1 bezeichnet, nutzt ein Adernpaar für eine Vollduplex-Ethernet-Verbindung und ist für Fahrzeugverkabelung ausgelegt.

- Unterstützt höhere Datenraten als klassische CAN-Netze
- Switches verbinden Kommunikationssegmente und Steuergeräte
- Ethernet bringt bekannte Netzwerkfragen mit: Segmentierung, Zugriffskontrolle und sichere Konfiguration
- Ein Gateway zwischen Ethernet und CAN muss Protokolle und zulässige Nachrichten kontrollieren

4. CAN-Fuzzing und Frame-Injection



Fuzzing sendet systematisch variierte oder unerwartete Eingaben, um Robustheits- und Sicherheitsfehler in einer kontrollierten Testumgebung zu finden. Frame-Injection bezeichnet das Einspeisen von CAN-Nachrichten in einen Bus, unautorisierte Nutzung kann gefährlich sein.

- Tests nur an eigenen Prüfständen oder mit ausdrücklicher Erlaubnis durchführen
- Mit Simulation oder isoliertem Prüfstand begeimen, nicht mit einem fahrenden Fahrzeug
- Beobachten, ob Steuergeräte fehlerhaft reagieren, hängen bleiben oder unerwartete Zustände annehmen
- Ergebnisse dokumentieren und gefundene Fehler verantwortungsvoll melden

Sicherheitsgrenze
Keine Frames in fremde Fahrzeuge oder produktive Fahrzeugsysteme einspeisen. Fahrzeugsteuerungen können sicherheitskritisch sein.

5. TELEMATIK UND CONNECTED-CAR-ANGRIFFSFLÄCHEN



Telematik-Einheiten verbinden Fahrzeugfunktionen mit Mobilfunk, Satellitennavigation, WLAN, Bluetooth, Apps oder Backend-Diensten, Jede Schnittstelle und Vertrauensgrenze kann eine Angriffsfläche bilden.

- Funkmodule und externe Diagnoseanschlüsse absichern
- Apps, Identitäten und Backend-APIs in die Risikoanalyse einbeziehen
- Nachrichten zwischen Telematik, Gateway und Steuergeräten begrenzen und prüfen
- Softwareupdates authentisieren und Integrität vor Installation prüfen

6. AUTOSAR SecOC



Secure Onboard Communication (SecOC) ist ein AUTOSAR-Mechanismus zum Schutz ausgewählter Nachrichten zwischen Steuergeräten. Er verwendet unter anderem einen Message Authentication Code (MAC) und einen Freshness-Wert, um Integrität, Authentizität und Aktualität zu prüfen.

- MAC-Prüfung hilft, Manipulation und nicht authentisierte Nachrichten zu erkennen
- Freshness-Werte helfen, wiederholte alte Nachrichten (Replay) zurückzuweisen
- SecOC verschlüsselt Nutzdaten nicht automatisch
- Wirksamkeit hängt von Kchlüsselverwaltung, Konfiguration, und korrekter Integration ab.

Merke
Authentizität und Integrität sind nicht dasselbe wie Vertraulichkeit: SecOC ist kein Enstaz für Verschlüsselung, wenn Daten geheim bleiben müssen.

7. ISO/SAE 21434



ISO/SAE 21434 beschreibt Anforderungen und Prozesse für Cybersecurity Engineering bei Straßenfahrzeugen. Der Standard behandelt Aktivitäten über den Lebenszyklus, darunter Risikoanalyse, Entwicklung, Produktion, Betrieb und Außerbetriebnahme.

- Bedrohungsanalyse und Risikobewertung (TARA) unterstützen die Ableitung von Maßnahmen
- Cybersecurity-Anforderungen in Entwicklung und Lieferkette berücksichtigen
- Risiken und Entscheidungen nachvollziehbar dokumentieren
- Der Standard ist ein Engineering-Rahmen, keine Garantie, dass ein Fahrzeug unangreifbar ist.

8. VEHICLE IDS



Ein Vehicle IDS überwacht Fahrzeugkommunikation oder Systemereignisse, um verdächtige Muster zu erkennen, Es ergänzt vorbeugende Schutz-maßnahmen, erstezt sie aber nicht.

- Regelbasierte Erkennung kann bekannte oder unzulässige Nachrichtenmuster melden
- Anomalieerkennung sucht Abweichungen vom erwarteten Verhalten
- Fehlalarme und übersehene Angriffe sind möglich; Erkennungsregeln müssen peprüft werden
- Alarme sicher protokollieren, bewerten und in einen Reaktionsprozess einordnen

MOBILE APPLICATION SECURITY

Eine Lernlandkarte zu acht häufigen Risikobereichen mobiler Apps. Die Bezeichnungen M1–M8 folgen dem vorgegebenen Themenrahmen und sind keine vollständige oder aktuelle OWASP-Rangliste. Risiken hängen von App, Plattform und Einsatz ab.

- Ziel: Daten, Konten und Kommunikation einer mobilen App schützen.
- Merke: Sicherheit entsteht durch mehrere abgestimmte Schutzmaßnahmen, nicht durch eine einzelne Technik.



EINORDNUNG

Die Mindmap ist Lernmaterial, keine Diagnose oder konkrete Handlungsempfehlung für ein bestimmtes Produkt.

MOBILE APP SECURITY



8. M8: UNSICHERE DATENHALTUNG

Daten auf dem Gerät können durch Gerätezugriff, Backups, Protokolle oder Fehlkonfiguration offengelegen jede Kompromittierung.



- Nur notwendige Daten speichern und Aufbewahrungszeiten begrenzen.
- Sensible Werte nicht in Klartextdateien, Logs oder gemeinsam genutztem Speicher ablegen.
- Daten zwischen sicheren App-Komponenten klar trennen.
- für geeignete Geheimnisse nutzen.
- Geräteschutt und App-Schutz ergänzen sich; keiner ersetzt den anderen.

2. M2: UNZUREICHENDE KRYPTOGRAFIE

Kryptografie schützt Daten nur bei passender Auswall und korrekter Anwendung. Sie behebt keine unsichere Speicherung oder fehlerhafte Zugriffskontrolle von selbst.



- Für vertrauliche Daten etablierte, von Plattform oder Bibliotkek bereitgestellte Verfahren verwenden.
- Schlüssel sicher verwalten und nicht fest in den App-Code einbauen.
- Veraltete oder selbst entworfene Kryptoverfahren vermeiden.
- Verschlüsselung schützt nicht automatisch vor Schadsoftware auf einem entsperrten Gerät.

7. M7: CODE-TAMPERING UND REVERSE ENGINEERING

Angreifende können App-Dateien analysieren oder verändern. Verschleierung kann die Analyse erschweren, aber kenien verlässlichen Schutz für Geheimnisse im Client bieten.



- Reine privaten Shlüssel oder deuarheft geheimen Zugangsdaten in der App ausliefern.
- Signierte Builds und kontrollierte Release-Prozesse einsetzen.
- Integritätsprüfungen als zusätzliche Hürde betrachten, nicht als allenige Vertrauensgrundlage.
- Kritische Eetschindungen und Berechtigungen auf vertrauenswürdigen Diensten durchsetzen.

3. M3: UNSICHERE AUTHENTIFIZIERUNG UND AUTORISZIERUNG

Authentifizierung prüft, wer ein Konto nutzt; Autorisierung entscheidet, welche Aktionen dieses Konto ausführen darf.



- Sitzungen sicher verwalten und bei Abmeldung oder Ablauf ungültig machen.
- Berechtigungen serverseitig für jede geschützte Aktion prüfen.
- Zugriff auf ein Objekt nicht allein aus einer vom Client gelieferten Kennung ableiten.
- Zusätzliche Gerätesperren sind kein Ersatz für die Prüfung geschützer Serverakperrten Gerät.

8. M8: PRIVILEGIEN-MISSBRAUCH UND ISOLATION

Apps sollten nur die Rechte erhalten, analysieren oder verändern. Verschleierung kann die Analyse erschweren, aber keinen verlässlichen Schutz für Geheimnisse im Client bieten.



- Keine privaten Schlüssel oder dauerhaft gehelnten Zugangsdaten in der App auslefern.
- Signierte Builds und kontrollierte Release-Prozesse
- Integritätsprüfungen als zusätzliche Hürde betrachten, nicht als alleinige Vertrauensgrundlage betrachten.
- Unnötige Exporte, Debug-Schnittstellen und privilegierte Berechtigungen vermeiden.

4. M4: UNZUREICHENDE EINGABVALIDIERUNG

Eingaben aus Formularen, Deep Links, Zwischenablage und Netzwerk sind grundsätzlich nicht vertrauenswürdig.



- Eingaben nach erwarteter Länge, Format und Wertebereich prüfen.
- Daten kontextgerecht kodieren oder parametrisiert verarbeiten.
- Validierung sowohl in der App als auch an der zuständigen Vertrauensgrenze durchführen.
- Fehler kontrolliert behandeln, ohne Geheimnisse oder interne Details preiszugeben.

6. M6: CODE-QUALITÄT UND SPEICHERFEHLER

Fehlerhafte Verarbeitung kann Abstürze, Datenlecks oder ausnutzbare Schwachstellen verursachen. Speicherfehler hängen auch von Sprache, Laufzeit und nativen Komponenten ab.



- Sichere APIs und gepflegte Abhängigkeiten bevorzugen.
- Eingaben und Fehlerfälle systematisch testen.
- Native Komponenten besonders sorgfältig prüfen: Speicherfehler wie Buffer Overflows sind dort möglich.
- Code-Reviews und automatisierte Tests ergänzen einander.

5. M5: UNSICHERE NETZWERKKOMMUNIKATION

Unverschlüsselte Übertragung kann Mitlesen oder Manipulation erleichtern. TLS schützt die Verbindung nur bei korrekter Konfiguration und Zertifikatsprüfung.



- HTTPS mit akuellen Plattform- und Bibliotheksstandards verwenden.
- Zertifikatsprüfungen nicht deaktivieren oder durch beliebige Zertifikate ersetzen.
- Geheimnisse nicht über Klartextverbindungen übertragen.
- Ein kompromittiertes Gerät oder ein kompromittierter Endpunkt bleibt ein Risiko, auch bei TLS.

SICHERHEITSPRINZIPIEN IM ZUSAMMENSPIEL



Die acht Themen greifer ineinander: Eine sichere Verbindung macht unsichere lokale Speicherung nicht sicher; gute Verschlüsselung ersetzt keine Autorisierung.

- Datenfluss betrachten: Eingabe → Verarbeitung → Speicherung → Übertragung.
- Schutzmaßnahmen an den jeweiligen Vertrauensgrenzen prüfen.
- Bedrohungen und Gegenmaßnahmen regelmäßig neu bewerten, wenn App oder Plattform sich ändern.

085 · MOBILE SICHERHEITSARCHITEKTUREN

Hardware, Isolation, Berechtigungen und Verwaltung schützen gemeinsam.

1. HARDWARE-VERTRAUENSANKER

Specialisierte Hardware kann sensible Schlüsseloperationen von der normalen Betriebssystemumgebung trennen.



- Apple Secure Enclave: isolierter Sicherheitsprozessor für bestimmte Schlüssel- und Schutzfunktionen.
- Android TEE: vertrauenswürdige Ausführungsumgebung, die vom Hauptbetriebssystem getrennt ist.
- TrustZone ist eine ARM-Technologie, auf der eine TEE auf geeigneter Hardware beruhen kann.
- Umsetzung und Schutzumfang variieren nach Gerät und Konfiguration.

2. SANDBOX UND DATEITRENNUNG

Apps werden in getrennten Sicherheitsbereichen ausgeführt, damit eine App nicht automatisch auf die Daten einer anderen zugreifen kann.



- iOS nutzt Sandbox-Regeln und kontrollierte Schnittstellen für App-Zugriffe.
- Android weist Apps eigene Identitäten zu und setzt Isolation unter anderem mit Betriebssystemmechanismen durch.
- Gemeinsam genutzte Daten erfordern vorgesehene APIs oder ausdrücklich erteilte Freigaben.
- Sandboxing begrenzt Folgen, verhindert aber nicht jeden Missbrauch innerhalb erlaubter Zugriffe.

3. BERECHTIGUNGEN UND APP-FÄHIGKEITEN

Berechtigungen steuern den Zugriff auf geschützte Ressourcen wie Standort, Kamera oder Kontakte.



- Android fragt bei vielen sensiblen Berechtigungen zur Laufzeit nach Zustimmung; Details hängen von Version und Berechtigung ab.
- iOS verwendet Berechtigungen und Entitlements, also deklarierte App-Fähigkeiten für bestimmte Dienste.
- Nutzerzustimmung ist kein Beleg dafür, dass ein Zugriff für jeden Zweck angemessen ist.
- Nur notwendige Berechtigungen erteilen und regelmäßig überprüfen.

4. APP-SIGNIERUNG

Digitale Signaturen helfen, Herkunft und Integrität eines App-Pakets zu prüfen und Updates einer App zuzuordnen.



- Android-APK-Pakete sind signiert; App Bundles (AAB) dienen der Veröffentlichung und werden für die Installation in passende Pakete aufgeteilt.
- Bei Google Play kann Play App Signing die Signaturverantwortung zwischen Upload-Schlüssel und Signierschlüssel aufteilen.
- iOS-Code wird signiert und durch Plattformregeln geprüft; Verteilung und Signaturanforderungen hängen vom Vertriebsweg ab.
- Signierung beweist nicht, dass eine App fehlerfrei oder harmlos ist.

5. ROOTING UND JAILBREAKING

Rooting (Android) und Jailbreaking (iOS) verändern oder umgehen vorgesehene Plattformbeschränkungen. Das kann Sicherheitsannahmen und Geräteintegrität beeinträchtigen.



- Apps können Integritätsindikatoren prüfen, etwa bekannte Änderungen oder ungewöhnliche Umgebungswerkmale.
- Solche Erkennung ist fehlbar und kann umgangen werden; ein positives oder negatives Signal ist kein sicherer Beweis.
- Sensible Funktionen serverseitig absichern, statt allein auf eine Geräteprüfung zu vertrauen.
- Keine Umgehung fremder Kontrollen: Prüfungen nur auf eigenen oder ausdrücklich freigegebenen Geräten untersuchen.

6. KEYCHAIN UND KEYSTORE

Betriebssystem-Dienste verwalten App-Geheimnisse und Schlüsselmaterial mit kontrolliertem Zugriff.



- Apple Keychain speichert geschützte Informationen wie Zugangsdaten und Schlüssel; Zugriffsregeln und Synchronisierung hängen vom Eintrag ab.
- Android Keystore ermöglicht die Verwaltung kryptografischer Schlüssel; unterstützte Hardware-Isolation ist geräteabhängig.
- Apps sollten Schlüssel nicht unnötig exportieren oder in Klartextdateien ablegen.
- Sicherer Schlüsselspeicher schützt nicht automatisch Daten, die eine App nach dem Entschlüsseln verarbeitet.

7. MDM UND ENROLLMENT

Mobile Device Management (MDM) ermöglicht Organisationen, verwaltete Geräte oder Arbeitsbereiche nach Richtlinien einzurichten und zu verwalten.



- Enrollment bezeichnet die registrierte Aufnahme eines Geräts oder Profils in eine Verwaltung.
- Verwaltungsumfang und sichtbare Kontrollen hängen von Plattform, Enrollment-Art und Organisationsrichtlinie ab.
- Profile können Konfigurationen wie WLAN, Zertifikate oder Einschränkungen bereitstellen.
- Vor der Registrierung klären, welche Daten und Geräteeinstellungen verwaltet werden.

8. ENTERPRISE MOBILITY UND TRENNUNG

Unternehmensmobilität verbindet Gerätschutz, Zugriffsregeln und organisatorische Verwaltung. Android Work Profile und verwaltete IOS-Konfigurationen können Arbeitsdaten von persönlichen Bereichen abgrenzen.



- Trennung reduziert Vermischung, ist aber abhängig von Plattform und Verwaltungskonfiguration.
- Zugriff nach Bedarf vergeben und Geräte sowie Apps aktuell halten.
- Verlust oder Diebstahl über die vorgesehenen Organisationsprozesse melden.
- Sicherheitsarchitiektur ist Zusammenspiel aus Hardware, Betriebssystem, Apps und Richtlinien.

MINDMAP 086: RANSOMWARE-ÖKOSYSTEM UND RaaS

Ransomware-Erpressung ist ein mehrstufiger Angriff, bei dem Täter Daten verschlüsseln, stehlen oder beides tun können. Bei Ransomware-as-a-Service (RaaS) stellen Betreiber Infrastruktur oder Schadsoftware bereit; Affiliates führen Angriffe aus. Nicht jeder Vorfall folgt demselben Ablauf.

- **Angriffskette verstehen – nicht als starres Schema.**
- **Schutz, Eindämmung und Wiederherstellung zusammendenken.**



1. RaaS-GESCHÄFTSMODELL

RaaS beschreibt ein arbeitsteiliges kriminelles Modell: Rollen und Bedingungen variieren je nach Gruppe.



- Operatoren entwickeln oder betreiben Plattformen und Schadsoftware.
- Affiliates nutzen bereitgestellte Mittel für eigene Angriffe.
- Initial Access Brokers verkaufen oder vermitteln kompromittierte Zugänge.
- Einnahmen können nach vereinbarten Anteilen aufgeteilt werden.
- Infrastruktur, Markenname und Vorgehen können sich ändern.

2. INITIAL ACCESS: ZUGANGSWEGE

Angreifer benötigen zunächst einen Zugang oder eine ausführbare Gelegenheit. Häufige Wege sind soziale Manipulation, ausgenutzte Schwachstellen oder kompromittierte Zugangsdaten.



- Phishing kann Empfänger zu schädlichen Anhängen, Links oder Anmeldeseiten verleiten.
- Exploit-Kits versuchee, verwundbare Software auszunutzen; ihr Einsatz ist nicht bei jedem Vorfall belegt.
- VPN-Zugänge können durch gestohlene Zugangsdaten oder ungepatchte Schwachstellen gefährdet sein.
- IMFA, zeitanhe Updates und sichere Anmeldung erschweren den Zugang.
- Protokolle und Warnmeldungen helfen, verdächtige Anmeldungen zu erkennen.

3. VORBEREITUNG UND LATERALE AUSBREITUNG

Nach einem ersten Zugang versuchen Täter mitunter, weitere Systeme und höherwertige Konten zu erreichen. Diese Phase kann der Erkundung, Ausweitung von Berechtigungen und Vorbereitung des Angriffs dienen.



- Ungewöhnliche Anmeldungen und Rechteänderungen untersuchen.
- Verbindungen zwischen Endgeräten, Servern und Identitätssystemen beobachten.
- Segmentierung begrenzt unnötige Wege zwischen Netzbereichen.
- Least Privilege reduiziert den Umfang zulässiger Zugriffe.
- Administrationskonten gesondert schützen und überwachen.

4. DATENEXFILTRATION UND DOUBLE EXTORTION

Bei Double Extortion werden Daten vor oder zusätzlich zur Verschlüsselung entwendet und mit Veröffenllicbung bedroht. Exfiltration ist nicht bei allen Ransomware-Angriffen nachweisbar.

- Mögliche Anzeichen: unegwöhnliche ausgehende Datenmengen oder neue externe Ziele.
- Netzwerk-, Cloud- und Endpoint-Protokolle zeitlich korrelieren.
- Datenzugriffe nach Konto, System und Umfang prüfen.
- Egress-Kontrollen und geeignete Überwachung erschweren unbemerkten Abfluss.
- Bei Verdacht Beweise sichern und den Vorfallprozess aktivieren.



5. SCHATTENKOPIEN UND BACKUPS

Angreifer können versuchen, Wiederherstellungswege zu beeinträchtigen, etwa lokale Schattenkopien oder erreichbare Sicherungen. Das Löschen von Schattenkopien bedeutet nicht automatisch, dass alle Backups zerstört wurden.

- Backups getrennt und gegen unbefugte Änderung oder Löschung geschützt hallen.
- Offline- oder unveränderbare Sicherungen erwägen.
- Wiederherstellung regelmäßig in kontrollierten Tests erproben.
- Backup-Zugänge und Löschvorgänge protokollieren und überwachen.
- Bei einem Vorfall Sicherungen nicht vorschnell mit möglicherweise kompromittierten Systemen verbinden.



6. VERSCHLÜSSELUNG UND SCHLÜSSELVERWALTUNG

Ransomware kann symmetrische Verschlüsselung für Daten und asymmetrische Kryptografie zum Schutz eines Sitzungsschlüssels kombinieren. Die konkrete Implementierung unterscheidet sich. Ohne passenden Schlüssel oder funktionierende Wiederherstellung ist Entschlüsselung nicht garantiert.

- Symmetrische Verfahren verwenden denselben geheimen Schlüssel zum Ver- und Entschlüsseln.
- Asymmetrische Verfahren verwenden ein Schlüsselpaar aus öffentlichem und privatem Schlüssel.
- Ein Angreifer kann Schlüsselmatarial zentral oder pro Opfer verwalten; das ist nicht einheitlich.
- Kryptografisch sichere Schlüsselverwaltung schützt legitime Systeme, ist aber keine Wiederherstellungsgarantie.
- Unabhängige Backups bleliben entscheidend.



7. VERHANDLUNG UND ZAHLUNG

Täter können Kommuniksationskanäle und Fristen zur Druckausübung einsetzen. Eine Zahlung garantiert weder Datenlöschung noch funktionierende Entschlüsselung und kann weitere Risiken mit sich bringen.

- Entscheidungen nicht unter Zeitdruck allein treffen.
- Incident Response, Rechtsberatung und zuständige Stellen einbeziehen.
- Beweise, Nachrichten und Zeitpunkte nachvollziehbar sichern.
- Zahlungs- und Sanktionsrisiken hängen von Umständen und geltenden Regeln ab.
- Wiederherstellung und Schutzmaßnahmen unabhängig von Verhandlungen planen.



8. PROAKTIVER SCHUTZ UND ISOLATION

Vorbereitung reduiziert Eintrittswahrscheinlichkeit und mögliche Auswirkungen. Bei einem begründeten Verdacht zählt eine koordinierte, schnelle Eindämmung.

- MFA und sichere, getrennte Administrationskonten einsetzen.
- Systeme, VPN-Komponenten und Sicherheitssoftware aktuell halten.
- Netze segmentieren und unnötige Berechtigungen abbauen.
- Backups isolieren, schützen und Wiederherstellung testen.
- Betroffene Geräte nach Incident-Response-Plan vom Netz trennen; Beweise erhalten und zuständige Teams informieren.
- Nach Eindämmung Zugangsdaten prüfen, Persistenz ausschließen und kontrolliert wiederherstellen.



Bei akutem Verdacht
Keine eigenmächtigen Bereinigungsversuche, die Spuren vernichten könnten. Den internen Notfallprozess starten und betroffene Systeme nach dessen Vorgaben isolieren.

SOCIAL ENGINEERING

Social Engineering nutzt soziale Situationen und psychologische Einflussnahme, um Menschen zu Handlungen oder zur Preisgabe von Informationen zu bewegen.

Erkennen, innehalten und über einen unabhängigen Kanal prüfen.

- Angriffe können digital, telefonisch oder vor Ort erfolgen.
- Eine glaubwürdig klingende Geschichte ist kein Identitätsnachweis.

1. CIALDINIS PRINZIPIEN

Einflussprinzipien können in Betrugsversuchen ausgenutzt werden; sie beweisen für sich allein keine Täuschung.

- Autorität: Eine Person beruft sich auf Rang oder Zuständigkeit.
- Verknappung: Angeblich bleibt nur wenig Zeit oder Gelegenheit.
- Reziprozität: Eine Gefälligkeit soll zu einer Gegenleistung bewegen.
- Druck ist ein Anlass zum Prüfen, nicht zum Überspringen von Regeln.

2. SPEAR-PHISHING UND WHALING

Spear-Phishing richtet sich gezielt an bestimmte Personen oder Teams. Whaling bezeichnet gezielte Angriffe auf hochrangige Führungskräfte.

- Beispiel: Eine vermeintliche Führungskraft fordert eine dringende Überweisung.
- Personalisierte Details können aus öffentlich verfügbaren Informationen stammen.
- Absenderadresse, Anfrage und Zahlungsdaten getrennt prüfen.
- Ungewöhnliche Vorgänge über einen bekannten internen Kanal bestätigen.

3. VISHING UND TELEFONBETRUG

Vishing ist Phishing über Sprachanrufe. Dabei können Anrufende eine Organisation oder Person vortäuschen.

- Warnzeichen: Zeitdruck, Geheimhaltungsforderungen oder Code-Abfragen.
- Angezeigte Rufnummern sind kein verlässlicher Identitätsnachweis.
- Gespräch beenden und selbst eine bekannte offizielle Nummer wählen.
- Einmalcodes und Passwörter nicht am Telefon weitergeben.

4. SMISHING UND MESSENGER-KÖDER

Smishing ist Phishing per SMS. Ähnliche Köder können über Messenger-Dienste verbreitet werden.

- Vorsicht bei unerwarteten Paket-, Konto- oder Sicherheitswarnungen.
- Links und QR-Codes nicht ungeprüft öffnen: Adresse unabhängig aufrufen.
- Anfragen nach Geld oder Codes über einen zweiten Kanal bestätigen.
- Verdächtige Nachrichten nach den Regeln der Organisation melden.

5. TAILGATING UND PIGGYBACKING

Tailgating und Pigggbacking bezeichnen das unbefugte Mitgelangen in einen gesicherten Bereich hinter einer berechtigten Person. Die Begriffe werden teils unterschiedlich verwendet.

- Keine fremde Person durch eine Zugangssperre mitnehmen lassen.
- Zutrittsausweis und Besuchsverfahren konsequent nutzen.
- Freundlich auf den Empfang oder die Anmeldung verweisen.
- Verdächtige Situationen nach dem Standortverfahren melden.

6. BAITING UND USB-MEDIEN

Baiting lockt mit einem vermeintlichen Vorteil, etwa einem gefundenen USB-Stick, um eine riskante Handlung auszulösen.

- Unbekannte Speichermedien nicht an Arbeitsgeräte anschließen.
- Fundstücke gemäß Organisationsverfahren abgeben.
- Auch bekannte Geräte nur entsprechend der Sicherheitsregeln verwenden.
- Bei verseentlichen [illegible]haltungs- odort die zuständige IT-Stelle informieren.

7. PRETEXTING UND IDENTITÄTSTÄUSCHUNG

Beim Pretexting erfindet oder nutzt eine angreifende Person einen Vorwand, um Vertrauen zu gewinnen oder Informationen zu erhalten.

- Beispiel: Eine angebliche Supportkraft verlangt Zugangsdaten zur Fehlerbehebung.
- Identität und Berechtigung über bekannte interne Wege prüfen.
- Nur erforderliche Informationen nach geltenden Regeln teilen.
- Ungewöhnliche Geheimhaltungs- oder Umgehungsforderungen hinterfragen.

8. HUMAN RISK MANAGEMENT UND AWARENESS

Wirksame Awareness unterstützt sicheres Verhalten durch verständliche Regeln, regelmäßige Übung und eine einfache Möglichkeit, Vorfälle zu melden.

- Übungen sollen Lerngelegenheiten schaffen, nicht beschämen.
- Klare Meldewege und zeitanhe Rückmeldungen fördern frühe Meldungen.
- Regeln für Zahlungen, Zugangsdaten und Besucher praktisch einüben.
- Erkenntnisse nutzen, um Prozesse und Schutzmaßnahmen zu verbessern.

Merksatz

Stopp – prüfen – unabhängig bestätigen – melden.

Sicherheitsregel

Keine Passwörter, Einmalcodes oder vertraulichen Daten auf Druck herausgeben. Zweifel über einen bekannten, unabhängigen Kontaktweg klären.

088 · PHYSISCHE IT-SICHERHEIT

Physische Sicherheit schützt Gebäude, Geräte und Infrastruktur vor unbefugtem Zugriff, Schäden und Ausfällen.



1. PERIMETERSCHUTZ

Äußere Barrieren erschweren unbefugtes Betreten des Geländes.



- Zaun: Geländegrenze sichtbar machen und Übersteigen erschweren
- Schranke: Zufahrt kontrollieren; Berechtigungen prüfen
- Beleuchtung und klare Sichtlinien unterstützen die Erkennung
- Zonen und Zufahrtswege risikogerecht planen

2. ZUTRITTSKONTROLLE

Zutritt wird anhand von Identität, Berechtigung und Bereich gesteuert.



- Karten: Besitz eines zugeordneten Ausweismittels prüfen
- Biometrie: körperliches Merkmal als Faktor; Datenschutz beachten
- Mehr-Faktor-Verfahren kombinieren unabhängige Nachweise
- Schleusen können jeweils nur eine Person passieren lassen
- Zugänge protokollieren und Berechtigungen regelmäßig entziehen

3. VIDEOÜBERWACHUNG (CCTV)

Kameras unterstützen Beobachtung und nachträgliche Untersuchung; sie ersetzen keine Zutrittskontrolle.



- Kamerablickfelder und tote Winkel bewusst planen
- Aufzeichnungen nur für festgelegte Zwecke und berechtigte Personen
- Speicherfristen nach geltendem Recht und dokumentiertem Zweck festlegen
- Zugriffe auf Aufnahmen protokollieren und Speicher sicher löschen
- Hinweise und Datenschutzanforderungen berücksichtigen

Wichtig
Es gibt keine universelle Aufzeichnungsfrist. Sie muss für den konkreten Einsatz rechtmäßig und zweckgebunden festgelegt werden.

4. BRAND- UND GASLÖSCHANLAGEN

Früherkennung und abgestimmte Löschung begrenzen Schäden an Räumen und Geräten.



- Brandmelder erkennen Rauch, Wärme oder andere definierte Signale
- Alarmierung, Evakuierung und Löschung aufeinander abstimmen
- Gaslöschung nur in dafür ausgelegten, geschützten Bereichen
- Novec und Inergen sind Beispiele für Löschmittel bzw. -ysteme
- Personenschutz, Wartung und Herstellerhinweise sind zentral

Sicherheit zuerst
Gaslöschanlagen können für Menschen gefährlich sein. Planung, Betrieb und Räumungsverfahren gehören in fachkundige Hände.

5. STROMVERSORGUNG

Redundante Versorgung hilft, Ausfälle kontrolliert zu überbrücken.



- USV überbrückt kurze Unterbrechungen und ermöglicht geordnetes Herunterfahren
- Batterien regelmäßig prüfen und nach Vorgaben warten
- Notstromdiesel kann längere Ausfälle abdecken, wenn verfügbar und betriebsbereit
- Umschaltung, Treibstoffversorgung und Last regelmäßig testen
- Notfallpläne und Wartungsprotokolle aktuell halten

6. KLIMATISIERUNG

Kontrollierte Luftführung und Temperatur schützen den zuverlässigen Betrieb der IT.



- Kaltgänge führen gekühlte Luft zu den Geräteeinlässen
- Warmgänge sammeln erwärmte Abluft
- Hot-/Cold-Aisle-Containment trennt Luftströme und vermindert Vermischung
- Sensoren überwachen Temperatur und Luftfeuchte an geeigneten Stellen
- Alarme, Wartung und Ausweichkapazität für Störungen planen

7. SERVER-RACKS UND SABOTAGESCHUTZ

Racks und Verkabelung gegen unbefugten Zugriff, Manipulation und versehentliche Schäden sichern.



- Racks verschließen; Schlüssel und Zugriffsrechte kontrollieren
- Kabel ordentlich führen und kritische Verbindungen kennzeichnen
- Manipulationsspuren und unerwartete Änderungen untersuchen
- Besuche begleiten und Arbeiten dokumentieren
- Schutzmaßnahmen regelmäßig prüfen, nicht nur nach Vorfällen

8. SICHERE HARDWARE-ENTSORGUNG

Datenträger und Geräte erst aussondern, wenn sensilde Daten angemessen behandelt wurden.



- Geräte inventarisieren und Datenklassifizierung berücksichtigen
- Datenträger nach geeignetem, dokumentiertem Verfahren löschen oder zerstören
- Degaussing kann für geeignete magnetische Medien infrage kommen; es ist nicht für alle Speichertypen geeignet
- SSD und Flash benötigen passande Lösch- oder Zerstörungsverfahren
- Übergabe, Transport und Entsorgung nachvollziehbar dokumentieren

Merke
Degaussing wirkt nicht zuverlässig auf nichtmagnetische Speichermedien wie SSDs oder USB-Flashspeicher.

089 · KRYPTOGRAFISCHE PROTOKOLLE

SSH, IPsec und WireGuard sind drei bewährte Protokolle für sichere Verbindungen. Sie bieten Vertraulichkeit und Integrität und unterscheiden sich in Architektur und Einsatz.

1. SSH-ARCHITEKTUR UND HOST-KEY-VALIDIERUNG



SSH baut eine verschlüsselte Verbindung zwischen Client und Server auf. Der Schlüsselaustausch vereinbart Sitzungsschlüssel; der Server weist seine Identität mit einem Hostschlüssel nach.

- Der Client prüft den Hostschlüssel gegen einen vertrauenswürdigen Eintrag, etwa in known_hosts.
- Eine unerwartete Änderung kann auf Neuinstallation, Schlüsselwechsel oder einen Man-in-the-Middle-Angriff hindeuten.
- Benutzeranmeldung erfolgt separat, beispielsweise mit öffentlichem Schlüssel oder Passwort.

2. SSH-HARDENING



Die Sicherheit hängt auch von Serverkonfiguration und Zugangskontrollen ab. Algorithmen werden zwischen Client und Server ausgehandelt.

- KexAlgorithms steuert zulässige Schlüsselaustauschverfahren; Ciphers steuert Verschlüsselungsverfahren.
- Veraltete oder unsichere Algorithmen deaktivieren und unterstützte aktuelle Verfahren einsetzen.
- Port-Forwarding nur bei Bedarf erlauben und Zugriffsrechte eng begrenzen.
- Ein anderer Port als 22 reduziert höchstens automatisierte Scan-Treffer; er ersetzt keine Authentifizierung oder Aktualisierung.

3. IPsec: AH UND ESP



IPsec ist eine Protokollfamilie für Schutz auf IP-Ebene. AH und ESP erfüllen unterschiedliche Aufgaben.

- AH bietet Integritäts- und Herkunftsschutz, aber keine Vertraulichkeit durch Verschlüsselung.
- ESP kann Vertraulichkeit sowie Integritäts- und Authentizitätsschutz bereitstellen, abhängig von Konfiguration und Algorithmen.
- In der Praxis ist ESP der übliche Baustein für verschlüsselte IPsec-VPNs.

4. IKEv1 UND IKEv2



IKE handelt Parameter und Schlüsselmaterial für IPsec-Sicherheitsassoziationen aus und authentifiziert die beteiligten Endpunkte. IKEv2 ist eine überarbeitete, eigenständige Version mit vereinfachtem Austausch gegenüber IKEv1.

- Sicherheitsassoziationen (SA) legen unter anderem Algorithmen und Schlüssel für den Schutz fest.
- IKEv1 hat unterschiedliche Austauschmodi; IKEv2 organisiert den Austausch in weniger Nachrichtenaustauschen.
- Die konkrete Sicherheit hängt von Authentifizierung, Parametern, Implementierung und Betrieb ab.

5. IPsec: TUNNEL- UND TRANSPORTMODUS



Der Modus bestimmt, welcher Teil des IP-Pakets geschützt beziehunggense gekapselt wird.

- Transportmodus schützt typischerweise Nutzlast des ursprünglichen IP-Pakets; der ursprüngliche IP-Header bleibt erhalten.
- Tunnelmodus kapselt ein vollständiges inneres IP-Paket in ein neues äußeres IP-Paket.
- Tunnelmodus eignet sich häufig für VPN-Verbindungen zwischen Gateways oder Netzen; Transportmodus häufig für Host-zu-Host-Kommunikation.

6. WIREGUARD UND NOISE



WireGuard ist ein VPN-Protokoll mit festgelegter Auswahl moderner kryptografischer Bausteine. Es verwendet das Noise-Protokoll-Framework für den authentifizierten Schlüsselaustausch.

- Peers werden über öffentliche Schlüssel identifiziert; Konfiguration ordnet ihnen erlaubte IP-Bereiche zu.
- Der Datentransport nutzt verschlüsselte UDP-Pakete; bei Bedarf helfen Keepalives, NAT-Zuordnungen offen zu halten.
- Kleine Implementierung und reduzierte Konfiguratisonsoptionen bedeuten nicht, dass Schlüsselverwaltung und Netzwerkregeln entfallen.

7. LEISTUNG UND ANGRIFFSFLÄCHEN



Ein pauschaler Sieger bei Leistung oder Sicherheit lässt sich nicht bestimmen: Hardware, Betriebssystem, Netzwerkpfad, Konfiguration und Einsatzszenario beeinflussen das Ergebnis.

- SSH eignet sich für interakitve Fernverwaltung und gezielte Weiterleitungen, nicht als allgemeiner Ersatz für ein standortweites VPN.
- IPsec ist breit integriert und flexibel, benötigt aber abgestimmte Parameter und sorgfältige Schlüsselverwaltung.
- WireGuard ist bewusst kompakt: Erreichbarkeit, Peer-Konfiguration und Schlüsselvertielung blieben wichtige Betriebsaufgaben.
- Bei allen giit: kompromittierte Endpunkte eder gestohlene Zugangsdaten können den Schutz umgehen.

8. PERFECT FORWARD SECRECY (PFS)



PFS bedeutet, dass die spätere Offenlegung eines langfristigen Authentifizierungsschlüssels nicht automatisch frühere aufgezeichnete Sitzungen entschlüsseln lässt. Dafür müssen Sitzungen mit frischem, ephemerem Schlüsselmaterial ausgehandelt werden.

- Bei IPsec ist die Aushandlung frischen Schlüsselmaterials für Daten-SAs relevant; IKE- und Phase-2-Parameter müssen zusammenpassen.
- Bei SSH und WireGuard gehören frische Sitzungsschlüssel zum jeweiligen Schlüsselaustausch; langfristige Schlüssel dienen der Identität beziehunggeense Peer-Zuordnung.
- PFS schützt nicht gegen kompromittierte Endgeräte während einer Sitzung und nicht gegen spätere Aufzeichnung plus schwache oder fehlerhafte Implementierungen.

Kurz zusammengefasst



SSH: sichere Fernverwaltung.
IPsec: Schutz auf IP-Ebene mit AH/ESP und IKE.
WireGuard: modernes, kompaktes VPN.

Sichere Konfiguration, Schlüsselverwaltung und vertrauenswürdige Endpunkte sind bei allen entscheidend.

090 POST-QUANTEN-KRYPTOGRAFIE

PQC schützt kryptografische Verfahren vor künftigen Quantenangriffen; Migration braucht Planung.

1. SHORS ALGORITHMUS: RSA UND ECC



- Shors Algorithmus kann auf einem ausreichend leistungsfähigen fehlertoleranten Quantencomputer die Faktorisierung großer Zahlen und diskrete Logarithmen effizient lösen.
- Betroffen: RSA-Schlüssel und viele ECC-basierte Schlüsselvereinbarungs- und Signaturverfahren.
- Ein solcher Angriff ist mit heutigen klassischen Rechnern nicht gleichzusetzen.
- Vertraulichkeit und Authentizität können durch unterschiedliche Verfahren betroffen sein.

2. GROVERS ALGORITHMUS: SYMMETRISCHE SCHLÜSSEL



- Grovers Algorithmus bietet für unstrukturierte Suche eine quadratische Beschleunigung.
- Auf die Schlüsselsuche übertragen, senkt er die ideale Sucharbeit grob von 2^n auf $2^{n/2}$ Schirtte: praktiscke Kosten und Voraussetzungen sind komplexer.
- Das ist keine vollständige Brechung symmetrischer Kryptografie.
- Größere Schlüssel können den theoretischen Sicherheitsverlust teilweise ausgleichen.
- Hashfunktionen können ebenfalls von quantenbeschleunigter Suche betroffen sein; die konkrete Auswirkung hängt vom Einsatz ab.

3. „STORE NOW, DECRYPT LATER" (SNDL)



- Bei SNDL sammelt ein Angreifer heute verschlüsselte Kommunikation, um sie möglicherweise später mit leistungsfähigeren Mitteln zu entschlüsseln.
- Besonders relevant sind Daten mit langer Geheimhaltungsdauer.
- Auch heute abgefangener Schlüsselaustausch kann künftig ein Risiko sein, wenn er auf angreifbaren Verfahren beruht.
- Datenminimierung und frühzeitige Planung verringern die Exposition.



RISIKO ABSCHÄTZEN

Frage: Wie lange müssen die Daten vertraulich bleiben, und wie lange dauert die Umstellung der betroffenen Systeme?

4. GITTERBASIERTE KRYPTOGRAFIE



- Gitterbasierte Verfahren beruhen auf schwierigen Problemen in hochdimensionalen Gittern. Sie gehören zu den wichtigsten Ansätzen für Post-Quanten-Schlüsselvereinbarung und digitale Signaturen.
- ML-KEM dient der Schlüsselvereinbarung beziehungsweise Kapselung eines gemeinsamen Geheimnisses.
- ML-DSA ist ein gitterbasiertes Signaturverfahren.
- Sicherheit hängt von der konkreten Konstruktion, Parametern und korrekter Implementierung ab.

5. NIST-PQC-STANDARDS: ML-KEM UND ML-DSA



- NIST veröffentlichte Standards für mehrere Post-Quanten-Verfahren. ML-KEM basiert auf CRYSTALS-Kyber; ML-DSA basiert auf CRYSTALS-Dilithium.
- ML-KEM: Kapselung eines gemeinsamen Schlüssels, nicht direkte Verschlüsselung beliebig großer Datenmengen.
- ML-DSA: digitale Signaturen zur Prüfung von Herkunft und Unveränderheit.
- Algorithmen korrekt zu implementieren und in Protokolle einzubetten bleibt entscheidend.



BEGRIFFE

Kyber und Dilithium sind die früheren Projektnamen; ML-KEM und ML-DSA sind die Namen der standardisierten Verfahren.

6. STATEFUL HASH-BASIERTE SIGNATUREN: LMS UND XMSS



- LMS und XMSS sind hashbasierte Signaturverfahren mit Zustand. Ihre Sicherheit stützt sich auf Hashfunktionen, doch die Verwaltung des Signaturzustands ist unerlässlich.
- Ein Einmalschlüssel darf nicht wiederverwendet werden.
- Der Zustand muss zuverlässig aktualisiert, gesichert und synchronisiert werden.
- Fehler bei Sicherung oder paralleler Nutzung können die Sicherheit gefährden.



BETRIEB BEACHTEN

Vor Einsatz prüfen, ob die zustandsbehaftete Schlüsselverwaltung zur Betriebsumgebung passt.

7. HYBRIDE ÜBERGANGSMODELLE



- Ein hybrides Modell kombiniert in einem festgelegten Protokoll klassische und Post-Quanten-Verfahren, zum Beispiel beim Schlüsselaustausch. Ziel ist, von den Sicherheitseigenschaften beider Komponenten zu profitieren.
- Die Kombination muss im Protokoll eindeutig und sicher spezifiziert sein.
- Fehlerhafte Verkettung oder Auswahl kann neue Schwachstellen schaffen.
- Kompatibilität und Interoperabilität sind in der Übergangsphase wichtige praktische Aspekte.



KEINE AUTOMATISCHE GARANTIE

„Hybrid" allein beweist keine Sicherheit. Entscheidend sind Konstruktion, Implementierung und korrekte Protokollnutzung.

8. KRYPTO-AGILITÄT UND MIGRATION



- Bestand erfassen: Algorithmen, Zertifikate, Protokolle, Schlüssel und Abhängigkeiten.
- Priorisieren: Datenlebensdauer, Exposition und Aufwand der Umstellung bewerten.
- Testen: Interoperabilität, Leistung, Fehlerbehandlung und Wiederherstellung prüfen.
- Einführen: Versionierung, Zuständigkeiten und Rückfallpläne dokumentieren.

HARDWARE SECURITY

Hardwarebasierte Vertrauensanker können Schlüssel schützen, Startvorgänge messbar machen und die Integrität eines Geräts unterstützen. Sie beseitigen jedoch nicht alle Risiken: Implementierung, Konfiguration und physischer Zugriff bleiben wichtig.

1. TRUSTED PLATFORM MODULE (TPM 2.0)



Ein TPM ist ein standardisiertes Sicherheitsmodul für kryptografische Operationen und Plattformzustand. Es kann Schlüssel schützen, ohne dass geheime Schlüsselmaterialien regulär an Betriebssystem-anwendungen ausgegeben werden müssen.

- TPM 2.0: Spezifikation für Funktionen und Schnittstellen
- PCRs (Platform Configuration Registers): halten erweiterbare Messwerte
- Messungen können Firmware- und Startkomponenten abbilden
- Sealing bindet Schlüsselzugriff an festgelegte PCR-Zustände
- TPM schützt nicht automatisch vor kompromittierter Software

2. HARDWARE SECURITY MODULE (HSM)



Ein HSM ist ein dediziertes System für geschützte Schlüsselverwaltung und kryptografische Verarbeitung. Es kann als eigenständiges Gerät, Netzwerkdienst oder integrierte Plattformlösung auftreten.

- Typen: Netzwerk-HSM, PCIe-Karte, Cloud-HSM-Dienst
- Schlüsselgenerierung und -nutzung innerhalb geschützter Grenzen
- Zugriffssteuerung, Rollen und Auditprotokolle ergänzen Schutz
- Backups und Wiederherstellung müssen sicher geplant sein
- HSM ersetzt weder sichere Anwendungen noch Schlüsselrichtlinien

3. UEFI SECURE BOOT



Secure Boot prüft beim Start digitale Signaturen von Boot-Komponenten anhand vertrauenswürdiger Schlüssel und Richtlinien. So kann nicht autorisierter Startcode blockiert werden; die konkrete Vertrauenskette hängt von Firmware und Konfiguration ab.

- UEFI-Firmware verwaltet Vertrauensdaten und Prüfregeln
- Signierte Bootloader können vor dem Start verifiziert werden
- Schlüsselbanken und Sperrlisten beeinflussen die Zulassung
- Secure Boot ist nicht dasselle wie vollständige Laufzeitprüfung
- Fehlkonfiguration oder verwundbare signierte Komponenten bleiben Risiken

4. MEASURED BOOT & REMOTE ATTESTATION



Measured Boot erfasst Messwerte von Startkomponenten und erweitert damit PCRs. Eine Remote Attestation kann solche Werte einem prüfenden Dienst zur Bewertung bereitstellen; sie beweist nicht automatisch, dass das gesamte System frei von Angriffen ist.

- Messkette: Komponenten messen nachfolgende Startstufen (Chain of nachfolgende)
- PCR-Erweiterung bindet Messwerte in eine Reihenfolge ein
- Attestation liefert signierte Zustandsinformationen
- Prüfer verglaiichen Werte mit kekannten Richtlinien
- Vertrauen hängt auch von Aktualität und Referenzwerten ab

5. HARDWARE ROOT OF TRUST & SILIZIUM-FINGERPRINTS



Eine Hardware Root of Trust (RoT) ist eine besonders geschützte Ausgangsbasis, auf der weitere Prüfungen aufbauen. Physically Unclonable Functions (PUFs) nutzen individuelle physikalische Eigenschaften eines Chips, sind aber kein Erstaz für sorgfältiges Schlüsselmanagement.

- RoT kann Startcode, Schlüssel oder Messungen verankern
- Vertrauensketten müssen jede Übergabe nachvollziehbar prüfen
- PUFs erzeugen gerätebezogene Antworten aus Hardwaremerkmalen
- Umgebung und Alterung können PUF-Messungen beeinflussen
- Sichere Entwürfe berücksichtigen Fehlertoleranz und Manipulation

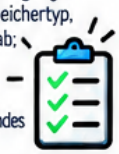
6. SEITENKANALANGRIFFE



Seitenkanäle verraten Informationen über indirekte Effekte einer Berechnung, etwa Laufzeit, Cacheverhalten oder Stromverbrauch. Spectre und Meltdown nutzten spekulative Ausführung auf bestimmte Weise; Rowhammer nutzt Störeinflüsse wiederholter Speicherzugriffe und ist nicht einfach ein klassischer Seitenkanal.

- Spectre: Ausnutrung spekulativer Ausführung zur Informationsgewinnung
- Meltdown: bestimmte Prozessoren ermöglichten unzulässige Datenoffenlegung
- Rowhammer: wiederholte Zugriffe können benachbarte DRAM-Zellen beeinflussen
- Gegenmaßnahmen umfassen Mikrocode-, Firmware- und Software-Updates
- Risiko und Schutz hängen von Hardwaregeneration und Einsatz ab

7. COLD-BOOT & RAM-FORENSIK



Cold-Boot-Angriffe zielen darauf, flüchtige Speicherinhalte kurz nach einer Unterbrechung der Stromversorgung auszulesen. Die Datenremanenz hängt von Speichertyp, Temperatur, Zeit und weiteren Bedingungen ab; moderne Geräte unterscheiden sich stark.

- Angriffsfenster: physischer Zugriff auf ein laufendes oder kürzlich ausgeschaltetes Gerät
- Verschlüsselung im Ruhezustand schützt nicht zwingend bereits entschlüsselte RAM-Daten
- Maßnahmen: Gerät sperren, vollständig herunterfahren, Zugriff kontrollieren
- Speicherforensik muss Beweismittel und Chain of Custody sichern
- Gegenmaßhahmen reduzieren Risiken, garantieren aber keine vollständige Abwehr

8. SICHERHEITSCHIPS IN ENDGERÄTEN



Integrierte Sicherheitskomponenten können Schlüssel und sicherheitskritische Abläufe vom Hauptbetriebssystem abgrenzen. Apple T2 und Titan M sind Beispiele für herstellerspezifische Komponenten; ihre Funktionen und Grenzen unterscheiden sich nach Produktgeneration.

- Apple T2: Sicherheitsfunktionen in ausgewählten Mac-Modellen
- Titan M: Sicherheitschip in ausgewählten Pixel-Geräten
- Sichere Boot- und Schlüsselprozesse können unterstützt werden
- Herstellerdokumentation klärt Modell- und Versionsumfang
- Ein Sicherheitschip macht ein Endgerät nicht automatisch unangreifbar

EINORDNUNG

Die Mindmap vermittelt Grundlagen. Sie ist keine Diagnose eines konkreten Geräts und keine Empfehlung für eine bestimmte Sicherheitskonfiguration.

DNS-SICHERHEIT

DNS übersetzt Domainnamen in Informationen wie IP-Adressen. Diese Mindmap zeigt, wie Antworten authentifiziert, Abfragen geschützt und Missbrauch erkannt werden können.

- Ziel: Vertraulichkeit, Integrität und sichere Richtlinien bei der Namensauflösung

1. DNSSEC-VERTRAUENSKETTE



DNSSEC ergänzt DNS um digitale Signaturen, damit Resolver die Herkunft und Integrität signierter DNS-Daten prüfen können.

- DNSKEY enthält öffentliche Schlüssel einer Zone; der private Schlüssel signiert Daten.
- RRSIG enthält die Signatur für einen Resource-Record-Satz (RRset) und Angaben zur Prüfung.
- Ein DS-Record in der übergeordneten Zone verweist per Hash auf einen DNSKEY der untergeordneten Zone.
- Die Vertrauenskette führt von einem Vertrauensanker über delegierte Zonen bis zu den signierten Daten.

2. VALIDIERUNG UND CACHE-POISONING



Ein validierender Resolver prüft DNSSEC-Signaturen entlang der Vertrauenskette. Das erschwert gefälschte Antworten, ersetzt aber keine allgemeine Absicherung des DNS-Betriebs.

- Prüft Signaturen, Schlüsselbezüge und Gültigkeitszeiträume.
- Eine ungültige Signatur oder unterbrochene Vertrauenskette kann zur Ablehnung der Antwort führen.
- Cache-Poisoning versucht, falsche DNS-Daten im Resolver-Cache zu platzieren.
- DNSSEC schützt nicht vor jeder Form von Ausfall, kompromittierten Endpunkten oder falscher Konfiguration.

3. DoH und DoT



DNS over HTTPS (DoH) und DNS over TLS (DoT) verschlüsseln den Transport zwischen Client und entsprechendem Resolver. Die Verschlüsselung authentifiziert nicht automatisch die DNS-Daten selbst.

- DoH überträgt DNS-Nachrichten über HTTPS; der Verkehr teilt den HTTPS-Transport.
- DoT schützt DNS-Verkehr mit TLS auf einem dafür vorgesehenen Transportkanal.
- Beide erschwerer Beobachtung oder Manipulation auf dem Transportweg zwischen Client und Resolver.
- DNSSEC kann zusätzlich die Authentizität signierter DNS-Daten prüfen.

4. VERSCHLÜSSELTES DNS IM UNTERNEHMEN



Verschlüsseltes DNS kann die Sichtbarkeit herkömmlicher Netzwerküberwachung reduzieren. Geplante Resolver und Endpunkt-Richtlinien helfen, Schutz und Monitoring zusammenzubringen.

- Vorteil: DNS-Abfragen sind auf dem Transportweg zum Resolver besser gegen Mitlesen geschützt.
- Herausforderung: Direkte DoH- oder DoT-Verbindungen können zentrale DNS-Protokollierung umgehen.
- Betrieb: Unternehmensgeräte auf freigegebene Resolver ausrichten und unerwartete Resolver-Verbindungen erkennen.
- Abwägung: Datenschutz, Richtlinien, Störungsanalyse und Aufbewahrung von Protokollen berücksichtigen.

5. DNS-TUNNELING UND EXFILTRATION



DNS-Tunneling kodiert Daten in DNS-Abfragen oder -Antworten, um einen Kommunikationskanal aufzubauen. Es kann für legitime Zwecke oder missbräuchlich eingesetzt werden.

- Auffällig können ungewöhnlich lange oder häufige Subdomain-Abfragen sein.
- Weitere Signale: hohe Entropie, ungewöhnliche Abfragemuster oder viele NXDOMAIN-Antworten.
- Iodine und DNScat2 sind bekannte Werkzeuge, die DNS-basierte Tunnel ermöglichen.
- Erkennung über Baselines, Resolver-Protokolle und Korrelation mit Endpunkt- und Netzwerkdaten.

SICHERHEITSGRENZE

Analysiere Tunnelindikatoren nur in autorisierten Umgebungen; ein einzelnes Merkmal beweist keinen Angriff.

6. DNS-SINKHOLING UND REPUTATIONSFILTER



Ein Sinkhole lenkt Anfragen für ausgewählte Domains auf kontrollierte Ziele um. Reputationsfilter können bekannte schädliche Domains blockieren oder protokollieren.

- Nutzen: Zugriff auf bekannte Schad- oder Phishing-Domains begrenzen und Sichtbarkeit schaffen.
- Risiko: Fehlklassifizierungen können legitime Dienste beeinträchtigen.
- Regeln und Feeds prüfen, Ausnahmen nachvollziehbar verwalten und Änderungen protokollieren.
- Ein Sinkhole stoppt nicht automatisch Schadsoftware oder bereits bestehende Verbindungen.

7. RESPONSE POLICY ZONES (RPZ)



RPZ ermöglicht DNS-Resolvern, Richtlinien auf passende DNS-Antworten anzuwenden. So lassen sich Daten aus Bedrohungsfeeds in kontrollierte DNS-Aktionen umsetzen.

- Mögliche Aktionen sind Antworten umschreiben, blockieren oder auf ein Sinkhole umlenken.
- Richtlinien können nach Domänen oder anderen DNS-Merkmalen greifen.
- Feed-Qualität, Aktualität und Ausnahmen beeinflussen die Trefferqualität.
- Änderungen testen und Auswirkungen auf Nutzer und Anwendungen beobachten.

8. VERDÄCHTIGE REGISTRIERUNGEN UND DGA



Domain Generation Algorithms (DGA) erzeugen viele mögliche Domainnamen, von denen Angreifer einige registrieren oder verwenden können. Die Analyse von Registrierungen und DNS-Verhalten liefert Hinweise, aber keinen sicheren Beweis für Missbrauch.

- Signale: ungewöhnliche Zeichenfolgen, viele neu auftretende Domains oder ähnliche Namensmuster.
- DNS-Daten mit Domain-Alter, Reputation und Endpunkt-Telemetrie korrelieren.
- Registrierungsdaten können unvollständig oder verzögert sein; Kontext und Baselines sind wichtig.
- Verdachtsfälle priorisieren, prüfen und dokumentierte Eskalationswege nutzen.

093 BGP-SICHERHEIT

Routing zwischen AS: Herkunft prüfen, Verbreitung begrenzen, Anomalien erkennen.

1. BGP-ARCHITEKTUR UND VERTRAUENSMODELL



BGP verbindet Routing-Domänen; es ist kein vollständiger Vertrauensnachweis für jede Route.

- Ein autonomes System (AS) wird durch eine ASN identifiziert.
- eBGP tauscht Routen zwischen AS aus; iBGP verteilt sie innerhalb eines AS.
- BGP-Nachrichten tragen unter anderem Präfixe und AS-Pfad-Informationen.
- Die Annahme einer Route ist Ergebnis lokaler Richtlinien und Vertrauensbeziehungen.

2. HIJACKING UND ROUTE LEAKS



Fehlerhafte oder böswillige Ankündigungen können den Verkehr fehlleiten.

- Beim Prefix-Hijacking kündigt ein AS ein fremdes Präfix an oder macht eine Route attraktiver.
- Ein Route Leak verbreitet Routen entgegen den beabsichtigten Routing-Richtlinien weiter.
- Folgen reichen von Umwegen und Ausfällen bis zu möglichem Abfangen von Verkehr.
- Nicht jede ungewöhnliche Route ist ein Angriff: Fehlkonfigurationen sind häufige Ursachen.

3. RPKI UND ROA



RPKI stellt kryptografisch signierte Zuordnungen bereit, die eine Autorisierung für Präfixankündigungen ausdrücken.

- Ein ROA benennt ein Präfix, die autorisierte Origin-ASN und optional eine maximale Präfixlänge.
- ROAs werden validiert; daraus entstehen Route-Origin-Validation-Ergebnisse.
- RPKI prüft damit die autorisierte Herkunft eines Präfixes, nicht den gesamten AS-Pfad.
- Ungültige oder fehlende Daten haben unterschiedliche Bedeutung: „invalid" ist nicht gleich „not found".

INerke

Ein gültiges ROA ist eine Aussage über die Präfix-Origin-Autorisierung, keine Garantie für eine sichere Route.

4. ROUTE ORIGIN VALIDATION (ROV)



ROV vergleicht eine BGP-Ankündigung mit den RPKI-Validierungsdaten.

- Valid: Origin-AS und Präfixlänge sind durch mndeendes ROA autorisiert.
- Invalid: Ein passendes ROA widerspricht Origin-AS oder erlaubter Präfixlänge.
- NotFound: Es gibt kein passendes ROA; die Herkunft ist dadurch nicht bestätigt, aber auch nicht widerlegt.
- Netzbetreiber legen fest, wie ihre Router oder Routing-Netzsme dierz Zustände behanlnle.
- Netzbetreitezt legen fest, wie ihre Router oder Routing-Systeme diese Zustände behandeln.

5. BGPsec: ERWEITERUNG UND GRENZEN



BGPsec ergänzt BGP um kryptografische Absicherung von AS-Pfad-Informationen.

- Signaturen sollen die Integrität der Pfadangaben entlang unterstützter BGPsec-Updates prüfbar machen.
- BGPsec ist von RPKI/ROV zu unterscheiden: ROV prüft die Origin-Autorisierung eines Präfixes.
- Einführung erfordert unterstützende Router, Schlüsselverwaltung und koordinierte Bereitstellung.
- BGPsec erstezt keine Routing-Richtlinien, Filter oder Betriebsüberwachung.

6. MANRS



MANRS steht für Mutually Agreed Norms for Routing Security und fördert bewährte Maßnahmen für sichereres Routing.

- Die Initiative richtet sich an Netzbetreiber, Internet Exchange Points, Cloud-Anbieter und andere Akteure.
- Zu den verbreiteten Mäßnahmen zählen Filterung, Anti-Spoofing, Koordination und globale Validierung.
- MANRS ist ein freiwilliger Handlungsrahmen, kein BGP-Protokoll und keine Garantie gegen Vorfälle.
- Die konkrete Umsetzung hängt von Rolle und Netzarchitektur ab.

7. PEERING-FILTER UND PREFIX-LIMITS



Filter begrenzen, welche Routen ein Nachbar ankündigen darf und welche akzeptiert werden.

- Import- und Exportfilter sollten zu Peering-Vertrag und Routing-Richtlinie passen.
- Prefix-Listen und AS-Pfad-Regeln reduzieren unbeabsichtigte oder unerlaubte Ankündigungen.
- Prefix-Limits begrenzen die Zahl akzeptierter Präfixe, je Sizzung; ein zu niedriges Limit kann legitime Routen abschneiden.
- Regelmäßige Pflege ist wichtig, wenn Präfixe oder Geschäftsbeziehungen sich ändern.

Betriebshinweis

Grenzwerte und Filter müssen zur erwarteten Route-Menge passen und kontrolliert angepasst werden.

8. MONITORING GLOBALER BGP-ANOMALIEN



Monitoring hilft, unerwartete Routing-Änderungen zeitnah zu erkennen und einzuordnen.

- Beobacthe neue oder zurückgezogene Präfixe, Änderungen des AS-Pfads und Routenfluktuationen.
- Vergleiche Ereignisse mit Wartungsfenstern, Peering-Änderungen und bekannten Störungen.
- Nutze mehrere Datenquellen; ein einzelner Messpunkt zeigt nicht zwingend die globale Sicht.
- Bei Verdacht: betroffene Ankündigung prüfen, Nachbarn kontaktieren und Eskalationswege nutzen.

094 · NAC & 802.1X

ZENTRUM · NAC & 802.1X

Network Access Control steuert, welche Geräte und Nutzer Zugang zu einem Netzwerk erhalten. IEEE 802.1X bietet dafür eine portbasierte Zugangskontrolle; Richtlinien und Netzwerktechnik ergänzen die Authentifizierung.

- Ziel: Zugang anhand festgelegter Identitäts- und Sicherheitsregeln erlauben, begrenzen oder verweigern.

Begriffe
Ein 802.1X-Port ist ein logischer Zugangspunkt, etwa an einem Switch-Port oder einer WLAN-Verbindung; nicht zwingend eine physische Buchse.

1 · IEEE 802.1X-ARCHITEKTUR

Drei Rollen arbeiten zusammen:

- **Supplicant:** Gerät oder Software, die Zugang anfordert.
- **Authenticator:** Switch oder WLAN-Access-Point; kontrolliert den Zugang und vermittelt Authentifizierungsverkehr.
- **Authentifizierungsserver:** prüft die Zugangsdaten, häufig über RADIUS. Der Server entscheidet nach der konfigurierten Richtlinie.

2 · EAP-METHODEN

EAP (Extensible Authentication Protocol) ist ein Rahmen für Authentifizierungsmethoden; es ist selbst keine einzelse Methode.

- EAP-TLS: gegenseitige Authentifizierung mit Zertifikaten; schützt die Verbindung bei korrekter Zertifikatsprüfung.
- PEAP: baut einen TLS-geschützten Tunnel auf, die innere Methode hängt von der Konfiguration ab.
- EAP-TTLS: TLS-Tunnel, in dem eine innere Authentifizierungsmethode verwendet werden kann.
- Zertifikatsprüfung, vertrauenswürdige Zertifizierungsstellen und sichere Gerätekonfiguration sind entscheidend.

Sicherheitsmerkpunkt
Ein verschlüsselter Tunnel allein genügt nicht: Wird das Serverzertifikat nicht geprüft, kann ein Gerät einem falschen Authentifizierungsserver vertrauen.

3 · RADIUS UND TACACS+

Beide Protokolle unterstützen zentrale Authentifizierungsdienste, werden aber oft in unterschiedlichen Einsatzbereichen verwendet.

- RADIUS: häufig für Netzwerkzugang; kombiniert Authentifizierung und Autorisierung in typischen Zugangsabläufen und kann Abrechnungsdaten übertragen.
- TACACS+: häufig für die zentrale Verwaltung administrativer Zugriffe auf Netzwerkgeräte; trennt Authentifizierung, Autorisierung und Abrechnung stärker.
- Protokollwahl richtet sich nach unterstützten Funktionen, Einsatzfall und vorhandener Infrastruktur.
- Beide benötigen sichere Konfiguration und geschützte Kommunikationswege.

4 · POSTURE ASSESSMENT

Eine Compliance-Prüfung bewertet, ob ein Endgerät festgelegte Sicherheitsanforderungen erfüllt.

- Mögliche Signale: verwalteter Gerätestatus, Betriebssystemstand, Verschlüsselung oder aktiver Schutzagent.
- Prüfergebnisse können Zugang erlauben, einschränken oder eine Behebung verlangen.
- Erfasste Merkmale und Entscheidungen sollten transparent, zweckgebunden und datensparsam behandelt werden.
- Ein Prüfsignal ist kein Beweis, dass ein Gerät frei von Schadsoftware oder Risiken ist.

5 · DYNAMISCHE VLANS & QUARANTÄNE

Nach erfolgreicher Authentifizierung kann eine Richtlinie den Netzwerkzugang passend zur Identität oder Gerätegruppe zuweisen.

- Dynamische VLAN-Zuweisung ordnet ein Gerät einem konfigurierten logischen Netzwerk zu.
- Ein Quarantäne- oder Remediation-Netz begrenzt den Zugang und kann nur nötige Dienste zur Fehlerbehebung bereitstellen.
- VLANs trennen Verkehr logisch, ersetzen aber keine Firewall-Regeln oder weitere Sicherheitskontrollen.
- Fehlerhafte Richtlinien können Geräte aussperren oder unerwünschten Zugang erlauben.

6 · MAC AUTHENTICATION BYPASS (MAB)

MAB ist ein Ausweichverfahren, bei dem die MAC-Adresse eines Geräts als Kennung für eine Zugangskontrolle verwendet wird.

- MAC-Adressen lassen sich fälschen; MAB bietet daher keine starke Geräteauthentifizierung.
- Nur dort einsetzen, wo 802.1X nicht praktikabel ist, und das Risiko ausdrücklich berücksichtigen.
- MAB-Zugänge möglichst restriktiv zuweisen, überwachen und auf benötigte Dienste begrenzen.
- MAB nicht mit einem kryptografischen Identitätsnachweis verwechseln.

Risiko
Eine bekannte oder beobachtete MAC-Adresse kann nachgeahmt werden. Eine Freigabe allein anhand dieser Adresse ist deshalb kein verlässlicher Identitätsbeweis.

7 · GASTZUGANG & PORT-SICHERHEIT

Gastzugänge sollten vom internen Netz getrennt und nachvollziehbar verwaltet werden.

- Eigene Gastnetz-Richtlinien und angemessene Netzwerksegmentierung verwenden.
- Nur erforderliche Ziele und Dienste freigeben; Verwaltungsoberflächen abschirmen.
- Switch-Ports und WLAN-Zugänge nach Bedarf absichern; ungenutzte Ports deaktivieren oder begrenzen.
- Zugangsdauer, Verantwortlichkeit und Überwachung passend zum Zweck festlegen.

8 · IDENTITÄTSPROVIDER & EDR

NAC kann Identitäts- und Gerätesignale aus weiteren Systemen in Zugangsentscheidungen einbeziehen.

- Identity Provider (IdP): liefert Identitäts- oder Gruppeninformationen über unterstützte Integrationen.
- EDR: kann Status- oder Warnsignale eines Endgeräts bereitstellen.
- Eine Richtlinie sollte festlegen, welche Signale welche Zugangsstufe auslösen.
- Verfügbarkeit, Aktualität und Fehlalarme der Signale berücksichtigen; Zugriff nicht blind von einem einzelnen Status abhängig machen.

095 WEB-SICHERHEIT: CLIENT-SIDE-ANGRIFFE

Browser, Webseiten und Herkunftsprüfungen schützen.

1. DOM-basiertes Cross-Site Scripting



Angreifende Daten gelangen in den DOM und werden dort als ausführbarer Inhalt interpretiert.

- Quelle: URL, Fragment, Nachricht oder Eingabefeld
- Gefährliche Sinks: innerHTML, document.write, eval
- Sicherer: textContent und sichere DOM-APIs
- Kontextgerecht validieren und kodieren

2. Clickjacking und UI Redressing



Eine täuschend eingebettete Seite bringt Nutzende dazu, unbeabsichtigt auf Bedienelemente einer anderen Seite zu klicken.

- Unsichtbare oder überlagerte Frames täuschen die Oberfläche vor
- Mögliche Folge: unerwünschte Aktionen mit angemeldeter Sitzung
- Frame-Busting per JavaScript ist kein verlässlicher Alleinschutz
- Schutz: CSP frame-ancestors oder X-Frame-Options

8. Browser-Sicherheitsfeatures



Browsermechanismen können bestimmte Klassen clientseitiger Fehler erschweren, wenn sie konsequent eingesetzt werden.

- Subresource Integrity (SRI) prüft eingebundene Dateien anhand eines Hashes
- Trusted Types können gefährliche DOM-Sinks auf vertrauenswürdige Werte begrenzen
- Sichere DOM-APIs und kontextgerechte Behandlung bleiben wichtig
- Features gezielt testen und als zusätzliche Schutzschicht einsetzen

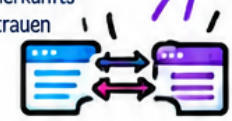
3. CSRF: gefälschte Anfragen



Cross-Site Request Forgery versucht, den Browser zu einer unerwünschten Anfrage an eine Website zu bewegen, bei der die Person angemeldet ist.

- Cookies können bei passenden Anfragen automatisch mitgesendet werden
- Schutz: serverseitig geprüfte, unvorhersagbare CSRF-Tokens
- SameSite-Cookies erschweren bestimmte Cross-Site-Anfragen
- Zusätzlich Herhunft prüfen; GET nicht für zustandsändernde Aktionen nutzen

7. PostMessage und Event-Listener



postMessage ermöglicht Nachrichten zwischen Fenstern oder Frames. Unsichere Herkunfts- oder Inhaltsprüfungen können Vertrauen missbrauchen.

- Beim Senden eine konkrete Ziel-Origin angeben
- Beim Empfang event.origin exakt prüfen
- Nachrichtenformat und erlaubte Aktionen validieren
- Keine vertraulichen Aktionen allein aufgrund einer Nachricht ausführen

4. CORS-Fehlkonfigurationen



Cross-Origin Resource Sharing (CORS) steuert, ob Browser JavaScript den Zugriff auf Antworten anderer Origins erlauben.

- CORS ist keine allgemeine Zugriffskontrolle für den Server
- Erlaubte Origins gezielt und exakt festlegen
- Credentials nicht mit beliebigen Origins kombinieren
- Preflight-Anfragen und erlaubte Methoden sorgfältig konfigurieren

6. Cross-Site Script Inclusion und Datenlecks



XSSI bezeichnet Risiken, bei denen eine fremde Website eingebundene Skripte oder Antworten ausnutzt, um sensible Daten zugänglich zu machen.

- Besonders riskant: sensible Daten als ausführbares JavaScript ausliefern
- Daten nicht als aktiven Skriptinhalt bereitstellen
- Antworten angemessen mit Content-Type und Schutzmaßnahmen ausliefern
- CORS und Authentisierung passend zum Datenzugriff konfigurieren

5. Content Security Policy



CSP ist eine Browserrichtlinie, die zulässige Quellen und Verhaltensweisen für Inhalte einer Webseite einschränkt.

- default-src setzt eine allgemeine Quellenvorgabe
- script-src begrenzt Skriptquellen; Nonces oder Hashes können erlaubte Skripte markieren
- unsafe-inline und unsafe-eval schwächen den Skriptschtuz
- CSP ergänzt sichere Programmierung, ersetzt sie aber nicht

SERVER-SIDE ATTACKS



1. SERVER-SIDE REQUEST FORGERY (SSRF)



Bei SSRF wird ein Server dazu gebracht, eine vom Angreifer beeinflusste Adresse abzurufen.

- Risiko: Zugriff auf interne Dienste oder Cloud-Metadaten.
- Blocklisten allein sind oft lückenhaft; erlaubte Ziele möglichst eng festlegen.
- Weiterleitungen, DNS-Auflösung und IPv4/IPv6 berücksichtigen.
- Ausgehenden Netzwerkverkehr begrenzen und sensible Metadaten schützen.

2. XML EXTERNAL ENTITY (XXE)



XXE entsteht, wenn ein XML-Parser externe Entitäten verarbeitet und diese Funktion nicht sicher konfiguriert ist.

- Eine DTD kann Entitäten definieren; externe Entitäten können auf Dateien oder Netzwerkressourcen verweisen.
- Mögliche Folgen: Offenlegung lokaler Daten, SSRF oder übermäßiger Ressourcenverbrauch.
- Externe Entitäten und DTDs deaktivieren, sofern sie nicht benötigt werden.
- Parser aktuell halten und Eingaben sowie Verarbeitungsgrenzen kontrollieren.

3. UNSICHERE DESERIALISIERUNG



Deserialisierung wandelt gespeicherte oder übertragene Daten zurück in Programmobjekte. Nicht vertrauenswürdige Eingaben können dabei gefährlich werden.

- Java: Gadget-Ketten in verfügbaren Klassen können bei unsicherer Objektverarbeitung problematisch sein.
- Python: pickle ist nicht für nicht vertrauenswürdige Daten geeignet und kann beim Laden Code ausführen.
- PHP: unsichere Verarbeitung serialisierter Objekte kann magische Methoden auslösen.
- Bevorzugt einfache Datenformate, prüfe Schema und Typen, und vertraue keiner Nutzereingabe.

4. SERVER-SIDE TEMPLATE INJECTION (SSTI)



SSTI liegt vor, wenn nicht vertrauenswürdige Eingaben als Template-Code statt als Daten interpretiert werden.

- Risiko reicht von unerwarteter Ausgabe bis zu serverseitiger Codeausführung.
- Nutzereingaben niemals in Template-Ausdrücke oder Template-Quelltext einsetzen.
- Daten getrennt an Templates übergeben; Kontext-Escaping nutzen.
- Template-Engine mit minimalen Rechten und sicherer Konfiguration betreiben.

5. RCE ÜBER FILE UPLOADS



Remote Code Execution (RCE) kann entstehen, wenn hochgeladene Dateien vom Server als ausführbarer Inhalt behandelt werden.

- Dateiendungen und vom Client gelieferte MIME-Typen sind kein verlässlicher Sicherheitsnachweis.
- Uploads außerhalb des Webroots speichern und Ausführung dort verhindern.
- Dateityp anhand des Inhalts prüfen; Größen- und Mengenlimits setzen.
- Dateien mit minimalen Rechten verarbeiten; bei Bedarf isoliert scannen oder konvertieren.

6. PATH TRAVERSAL UND LOCAL FILE INCLUSION



Path Traversal manipuliert Pfade, um vorgesehene Verzeichnisse zu verlassen. LFI bindet lokal gespeicherte Dateien ein.

- Relative Pfadbestandteile wie ../ können eine unzureichende Pfadprüfung umgehen.
- Dateien über feste IDs oder eine Allowlist auswählen statt über freie Pfade.
- Kanonische Pfade bilden und sicherstellen, dass sie innerhalb des erlaubten Verzeichnisses liegen.
- Dateizugriffsrechte minimieren; Nutzerinput nicht direkt als Include-Ziel verwenden.

7. HTTP REQUEST SMUGGLING



Request Smuggling kann auftreten, wenn vorgeschaltete und nachgelagerte HTTP-Komponenten eine Anfrage unterschiedlich abgrenzen.

- CL.TE: Frontend und Backend priorisieren Content-Length bezeichnungsgemäß Transfer-Encoding unterschiedlich.
- TE.CL: Die gegensätzliche Priorisierung kann ebenfalls zu abweichender Nachrichtenabgrenzung führen.
- Mögliche Folgen betreffen nachfolgende Anfragen und gemeinsam genutzte Verbindungen.
- Fehlerdeutige Anfragen an der Vertrauensgrenze ablehnen; Proxy- und Backend-Konfiguration abstimmen.

8. ABWEHR UND SICHERE EINGABEMUSTER



Robuste Abwehr setzt auf sichere APIs, klare Vertrauensgrenzen und minimale Rechte - nicht nur auf das Bereinigen einzelner Zeichen.

- Eingaben nach Schema und Zweck validieren; unerwartete Werte ablehnen.
- Für SSRF ausgehende Ziele begrenzen; für Parser gefährliche Funktionen abschalten.
- Templates mit Daten statt zusammengesetztem Code füllen.
- Dateien isoliert speichern, Pfade begrenzen und HTTP-Nachrichten eindeutig behandeln.
- Fehler sicher protokollieren, Komponenten aktualisieren und Schutzmaßnahmen testen.

SICHERHEITSKONTEXT

Die Beispiele dienen dem Lernen und der defensiven Prüfung eigener Systeme. Nur mit ausdrücklicher Berechtigung testen.

API-SICHERHEIT

Risiken verstehen



APIs verbinden Anwendungen und Daten. Diese Mindmap erklärt acht häufige Risikobereiche aus den OWASP API Security Top 10 (2023) und zeigt grundlegende Schutzideen. Die Nummerierung folgt dieser Ausgabe; sie ist keine Rangfolge für jedes System.

- Authentifizierung prüft, wer eine Anfrage stellt; Autorisierung prüft, was diese Identität tun darf.
- Ein API-Risiko wird im Kontext bewertet: Daten, Funktionen, Geschäftsabläufe und Betriebsumgebung zählen.

1. BOLA/IDOR

Objektberechtigung



- Beispiele: Zugriff auf fremde Objekte (z. B. IDs, Dateien, Ressourcen).
- Schutz: Objektzugriff prüfen, kontextbezogene Berechtigungen, keine vorhersehbaren IDs.

2. Broken Authentication

Anmeldung und Token



- Beispiele: Gestohlene Tokens, unsichere Sitzungen, fehlerhafte Anmeldung.
- Schutz: Starke Authentifizierung, kurzlebige Tokens, sichere Sitzungsverwaltung.

3. Property Level Authorization

Offenlegung und Änderung von Feldern



- Beispiele: Ungewollte Sichtbarkeit oder Änderung sensibler Felder.
- Schutz: Feldbasierte Berechtigungen, selektive Datenfreigabe.

4. Resource Consumption

Limits und Verfügbarkeit



- Beispiele: Überlastung durch viele Anfragen, hohe Ressourcen-Nutzung.
- Schutz: Rate-Limiting, Quoten, Monitoring und Alerting.

5. Function Level Authorization

Rollen und Aktionen



- Beispiele: Unbefugte Nutzung von Funktionen, falsche Rollen.
- Schutz: Rollenbasiertes Zugriffsmanagement, feine Berechtigungen, Prüfung von Aktionen.

6. Sensitive Business Flows

Missbrauch von Abläufen



- Beispiele: Manipulation von Zahlungen, Bestellungen oder Onboarding-Prozessen.
- Schutz: Abläufe validieren, Transaktionen prüfen, Anomalien erkennen.

7. SSRF

Serverseitige Abrufe



- Beispiele: Zugriff auf interne Netzwerke, Cloud-Metadaten, externe Systeme.
- Schutz: Nur erlaubte Ziele, URL-Validierung, Netzwerk-Segmentierung.

8. Security Misconfiguration

Sichere Konfiguration und Schatten-APIs



- Beispiele: Standardpasswörter, zu viele Rechte, unentdeckte Schatten-APIs.
- Schutz: Sichere Defaults, regelmäßige Prüfung, API-Discovery.

BEGRIFFE



BOLA steht für Broken Object Level Authorization; IDOR (Insecure Direct Object Reference) bezeichnet ein verwandtes Zugriffskontrollproblem. Die Begriffe überschneiden sich, sind aber nicht in jedem Kontext austauschbar.

API1: BOLA

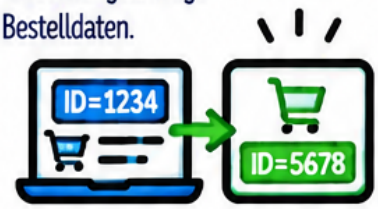
1. API1: BROKEN OBJECT LEVEL AUTHORIZATION

- Eine API nimmt eine Objektkennung entgegen, prüft aber nicht zuverlässig, ob die anfragende Identität genau dieses Objekt verwenden darf.
- Eine schwer erratbare ID ersetzt keine Berechtigungsprüfung.



1. API1: BROKEN OBJECT LEVEL AUTHORIZATION

- Beispiel: Ein angemeldeter Nutzer ändert die ID einer Bestellungsanfrage und erhält fremde Bestelldaten.



2. BEDEUTUNG

- Berechtigungen müssen bei jedem Objektzugriff geprüft werden, nicht nur bei der Anmeldung.



3. SCHUTZ

- Schutz: Berechtigung bei jedem Objektzugriff serverseitig anhand von Identität, Aktion und Objekt prüfen.



4. RISIKEN

- Unbefugter Zugriff kann zu Datenlecks, Manipulationen und finanziellen Schäden führen.



5. PRAXISBEISPIEL

- Ein Nutzer ändert die Bestellungs-ID in der API und erhält die Daten eines anderen Kunden.



6. MERKE

- Objektzugriff standardmäßig verweigern und nur nach einer passenden, aktuellen Prüfung erlauben.



7. TESTIDEE

- Mit getrennten Testkonten IDs austauschen und Lese-, Änderungs- sowie Löschzugriffe prüfen.



8. KURZE ZUSAMMENFASSUNG

- Eine API ohne zuverlässige Objektprüfung kann zu unbefugtem Zugriff auf fremde Objekte führen.



2. API2: AUTHENTIFIZIERUNG

1. RISIKEN

Fehler in Anmeldung, Sitzungsverwaltung oder Token-Verarbeitung können es Angreifenden ermöglichen, sich als andere Identität auszugeben oder Zugangsdaten zu missbrauchen.



2. SICHERE TOKEN-VERWALTUNG

- Zugangsdaten und langlebige Token nicht in URLs, Logs oder öffentlich erreichbarem Client-Code ablegen.
- Token sicher übertragen, Ablauf und Widerruf berücksichtigen und Signatur sowie erwartete Ansprüche serverseitig prüfen.



3. MEHRFAKTOR-AUTHENTIFIZIERUNG

- Mehrfaktor-Authentifizierung kann besonders schützenswerte Konten zusätzlich absichern.



4. RATE-LIMITS

- Anmeldeversuche begrenzen und auffällige Nutzung überwachen; Rate-Limits allein ersetzen keine sichere Anmeldung.



5. TOKEN-VALIDIERUNG



6. AUTORISIERUNG



7. ZUSÄTZLICHE SICHERHEIT

- Mehrfaktor-Authentifizierung kann besonders schützenswerte Konten zusätzlich absichern.



8. KURZE ZUSAMMENFASSUNG

- Die wichtigsten Maßnahmen für sichere Authentifizierung auf einen Blick.



ABGRENZUNG

Ein gültig signiertes Token beweist nicht automatisch, dass die angefragte Aktion erlaubt ist. Autorisierung bleibt eine eigene Prüfung.

3. API3: EIGENSCHAFTEN

1. WAS IST ES

- Diese Kategorie verbindet Risiken durch übermäßige Offenlegung von Eigenschaften und durch unzureichende Prüfung beschreibbarer Eigenschaften.
- Eine API sollte pro Vorgang nur die nötigen Datenfelder freigeben.



2. RISIKEN

- Übermäßige Offenlegung von Eigenschaften.
- Unzureichende Prüfung beschreibbarer Eigenschaften.



3. BEISPIELE

- Beispiel Offenlegung:
  Eine Antwort enthält interne Felder oder persönliche Daten, die der Client nicht benötigt.
- Beispiel Änderung:
  Ein Client kann ein geschütztes Feld wie „Rolle“ oder „Kontostatus“ unerlaubt setzen.



4. SCHUTZREGELN

- Antwortfelder gezielt auswählen und erlaubte Eingabefelder explizit festlegen.
- Vertrauenswürdige Geschäftsregeln und sensible Felder ausschließlich serverseitig kontrollieren.



5. SICHERHEIT

- Nur notwendige Daten freigeben.
- Eingaben strikt validieren.
- Unberechtigte Feldänderungen unterbinden.



6. BEISPIEL

- Offenlegung:
  Antwort enthält interne Felder wie „isAdmin“ oder „geheimeNotiz“.
- Änderung:
  Client setzt Feld „Rolle“ auf „Admin“.



7. VORGEHEN

- Relevante Felder identifizieren.
- Nur notwendige Felder freigeben.
- Eingabefelder explizit erlauben.
- Serverseitige Validierung nutzen.



8. KURZE ZUSAMMENFASSUNG

- Übermäßige Offenlegung und unzureichende Prüfung sind Hauptursachen.
- Gezielte Feldauswahl und serverseitige Kontrolle minimieren Risiken.



PRAKTISCHE PRÜFUNG



Vergleiche für jede Rolle die tatsächlich zurückgegebenen und änderbaren Felder mit den ausdrücklich benötigten Feldern.

4. API4: RESSOURCEN-VERBRAUCH

4. API4: UNBEGRENZTER RESSOURCENVERBRAUCH

- Unbegrenzte oder schlecht begrenzte Anfragen können Rechenleistung, Speicher, Netzwerkbandbreite oder kostenpflichtige Dienste erschöpfen. Das kann die Verfügbarkeit beeinträchtigen oder unerwartete Kosten verursachen.



4. API4: LIMITS DEFINIEREN

- Begrenze Anfrageraten passend zu Konto, Endpunkt und Geschäftsrisiko; definiere auch sinnvolle Spitzenlastregeln.
- Setze Obergrenzen für Seitengröße, Uploads, Batch-Anfragen und teure Such- oder Rechenoperationen.



4. API4: KOSTENRISIKEN IM BLICK

- Ein Rate-Limit ist eine Schutzschicht, aber kein vollständiger Schutz vor Missbrauch oder Überlastung.



4. API4: ZEIT- UND RESSOURCENBUDGETS

- Nutze Zeitlimits und Ressourcenbudgets für abhängige Dienste; überwache ungewöhnliche Nutzung und Kosten.



4. API4: ÜBERWACHUNG

- Beobachte die Nutzung von Anfrageraten, Speicher, Netzwerk und Kosten. Erkenne ungewöhnliche Muster frühzeitig.



4. API4: ABWÄGUNG

- Zu enge Limits können legitime Nutzung stören. Grenzwerte anhand erwarteter Nutzung testen und nachvollziehbar kommunizieren.



4. API4: ÜBERGREIFEND SCHÜTZEN

- Kombiniere Limits, Überwachung und Budgets für einen wirksamen Schutz.
- Passe Einstellungen an Nutzung und Risiko an.



5. API5: FUNKTIONS-BERECHTIGUNG

1. WAS IST ES

- Eine API prüft nicht ausreichend, ob eine Identität eine Funktion ausführen darf.
- Das Risiko betrifft Aktionen und Rollen, nicht nur den Zugriff auf einzelne Objekte.



2. BEISPIEL

- Ein normales Konto kann einen Verwaltungsendpunkt aufrufen, weil nur die Anmeldung geprüft wird.



3. SCHUTZ

- Für jede Aktion serverseitig Rolle, Berechtigung und Kontext prüfen.



4. NICHT VERTRAUEN

- Nicht auf verborgene Schaltflächen, schwer erratbare URLs oder HTTP-Methode als Sicherheitskontrolle vertrauen.



5. TESTS

- Tests: Funktionen mit Konten unterschiedlicher Berechtigungs-stufen aufrufen, auch direkt statt über die Benutzeroberfläche.



6. BELLES

- Fest auf Aktionen mit bekannt Enden ache, oder jed natürlic hB Benlangoee, URLs oder HTTP-Methode gerüt: als Sicherheitskontrolle vertrfiggrtäten.

7. ROLLEN

- Unterschiedliche Rollen haben unterschiedliche Berechtigungen.
- Rollen müssen auf Funktionsebene geprüft werden.



8. KURZE ZUSAMMENFASSUNG

- Aktionen und Rollen prüfen, nicht nur Objekte.
- Serverseitige Berechtigungen und Kontext verwenden.
- Tests mit unterschiedlichen Konten durchführen.



MERKE



„Angemeldet“ bedeutet nicht „für jede Funktion berechtigt“.

6. API6: UNRESTRICTED ACCESS TO SENSITIVE BUSINESS FLOWS

Geschäftsabläufe können missbraucht werden, obwohl einzelne Anfragen technisch korrekt autorisiert sind. Entscheidend ist, ob die API einen Ablauf in unzulässigem Umfang automatisiert oder ausnutzen lässt.



- Beispiele: massenhafte Reservierungen, automatisierte Kontoerstellung oder wiederholte Nutzung eines begrenzten Angebots.
- Erkenne missbräuchliche Muster anhand von Ablauf, Häufigkeit und Geschäftskontext statt nur einzelner Anfragen.
- Nutze geeignete Begrenzungen, Missbrauchserkennung und zusätzliche Schritte bei riskanten Abläufen.
- Schütze legitime Automatisierung und Barrierefreiheit; Maßnahmen müssen zum Risiko passen.

1. WAS IST ES

- Schutz vor missbräuchlicher Nutzung von Geschäftsabläufen über die API.
- Verhindert die Automatisierung unzulässiger Abläufe.
- Erhöht die Sicherheit sensibler Prozesse und Daten.



2. WOZU DIENT ES

- Missbrauch von Geschäftsabläufen verhindern.
- Sensible Prozesse schützen.
- Automatisierung und legitime Nutzung ermöglichen.



8. KURZE ZUSAMMENFASSUNG

- Missbrauch von Geschäftsabläufen verhindern.
- Ablauf, Häufigkeit und Kontext berücksichtigen.
- Schutz und Flexibilität im richtigen Gleichgewicht.



API6: GESCHÄFTSABLÄUFE

3. WAS ES UMFASST

- Ablauf- und Prozesskontrolle.
- Erkennung auffälliger Muster.
- Monitoring und Risikomanagement.



7. BEST PRACTICES

- Geschäftsabläufe analysieren und typische Missbrauchsmuster erkennen.
- Klare Limits und Schwellenwerte definieren.
- Automatisierte Erkennung und gezielte Gegenmaßnahmen implementieren.



4. HÄUFIGE RISIKEN

- Massenhafte Buchungen.
- Automatisierte Kontoerstellung.
- Ausnutzung zeitlich begrenzter Angebote.



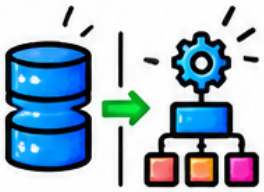
5. KONTROLLEN

- Verhaltensanalyse.
- Ablauf- und Frequenzüberwachung.
- Zusätzliche Prüfungen bei kritischen Abläufen.



6. ABGRENZUNG ZU API4

API4 fokussiert den Verbrauch technischer Ressourcen.
API6 fokussiert den Missbrauch eines sensiblen Geschäftsablaufs.



7. API7: SSRF

7. API7: SERVER-SIDE REQUEST FORGERY

Bei SSRF (Server-Side Request Forgery) lässt sich ein Server durch eine Anfrage dazu bringen, selbst eine vom Angreifer beeinflusste Adresse abzurufen. Dadurch können auch interne oder anderweitig nicht direkt erreichbare Ziele betroffen sein.



2. RISIKO

- Risiko: Eine API lädt eine vom Client angegebene URL ohne ausreichende Prüfung.



8. KURZE ZUSAMMENFASSUNG

- Zieladressen erlaubnisbasiert beschränken; interne, lokale und nicht benötigte Ziele blockieren.



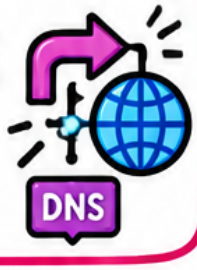
3. PRÜFUNG EINBEZIEHEN

- Weiterleitungen, DNS-Auflösung und alternative IP-Darstellungen in die Prüfung einbeziehen.



7. WEITERLEITUNGEN & DNS

- Weiterleitungen, DNS-Auflösung und alternative IP-Darstellungen in die Prüfung einbeziehen.



4. NETZWERKZUGRIFF

- Ausgehenden Netzwerkzugriff begrenzen und Abrufe in einer isolierten Umgebung ausführen.



WICHTIG

- Eine einfache Prüfung des URL-Textes kann unzureichend sein: Das Ziel kann sich durch Weiterleitungen oder Namensauflösung ändern.



8. API8: SECURITY MISCONFIGURATION

Unsichere oder uneinheitliche Konfigurationen können APIs unnötig exponieren. Dazu gehören beispielsweise zu freizügige Zugriffsregeln, nicht benötigte Funktionen oder unzureichend geschützte Fehlermeldungen.

1. SICHERE DEFAULTS

- Sichere Standardkonfiguration nutzen und nicht benötigte Endpunkte, Methoden und Funktionen deaktivieren.



2. ZUGRIFF

- Zugriff auf Ursprünge, Netzwerk und Verwaltungs-oberflächen bewusst begrenzen.



3. FEHLERMELDUNGEN

- Fehlermeldungen dürfen keine Geheimnisse, internen Details oder unnötigen technischen Informationen offenlegen.



4. ÄNDERUNGEN

- Konfigurationen, Abhängigkeiten und Bereitstellungen regelmäßig prüfen und Änderungen nachvollziehbar verwalten.



8. KURZE ZUSAMMENFASSUNG

- Unsichere Konfigurationen führen zu unnötiger Exposition und erhöhen das Risiko.
- Standardmäßig sicher konfigurieren, Zugriff begrenzen, Fehlermeldungen entschärfen und Änderungen verfolgen.



5. SCHATTEN-APIs

- Unbekannte, vergessene oder nicht verwaltete API-Endpunkte erschweren Schutz und Inventarisierung.
- APIs erfassen und - verantwortliche Teams zuordnen.



DAS WICHTIGSTE IM ÜBERBLICK

- Sichere Standardkonfigurationen nutzen.
- Zugriff bewusst begrenzen.
- Fehlermeldungen entschärfen.
- Änderungen nachvollziehbar verwalten.
- Schatten-APIs identifizieren und zuordnen.



API-SICHERHEIT: ZUSAMMENFASSUNG

1. ZUSAMMENFASSUNG: API-RISIKEN SYSTEMATISCH PRÜFEN

Die acht Knoten decken Autorisierung für Objekte, Eigenschaften und Funktionen, Anmeldung, Ressourcen, Geschäftsabläufe, serverseitige Abrufe sowie Konfiguration ab. Wirksame API-Sicherheit verbindet diese Prüfungen mit Inventarisierung, Tests und laufender Beobachtung.



- Für jeden Endpunkt: Wer darf welche Aktion auf welche Daten ausführen?
- Für Eingaben und Ausgaben: Welche Felder und Ressourcen sind wirklich erforderlich?
- Für Betrieb und Abläufe: Welche Grenzwerte, Abhängigkeiten und unbekannten Endpunkte müssen kontrolliert werden?
- Die OWASP-Kategorien sind ein Lern- und Prüfrahmen, keine Garantie, dass jede Schwachstelle erfasst ist.

2. IDENTITÄT

- Zugriff steuern.
- Authentifizierung prüfen.
- Berechtigungen verwalten.



3. DATEN

- Eingaben validieren.
- Ausgaben prüfen.
- Sensible Daten schützen.



4. GRENZEN

- Rate Limits setzen.
- Abhängigkeiten prüfen.
- Fehler und Ausnahmen behandeln.



5. GESCHÄFTSABLÄUFE

- Prozesse analysieren.
- Rollen und Verantwortlichkeiten prüfen.
- Sicherheitskontrollen integrieren.



6. SERVERSEITIGE ABRUFE

- Externe Dienste prüfen.
- Datenflüsse überwachen.
- Missbrauch verhindern.



7. KONFIGURATION

- Sichere Einstellungen verwenden.
- Fehlkonfigurationen vermeiden.
- Änderungen überwachen.



8. INVENTARISIERUNG

- Alle Endpunkte erfassen.
- Versionen und Abhängigkeiten berücksichtigen.
- Unbekannte Endpunkte identifizieren.



SICHER ÜBEN



Tests nur an eigenen Systemen oder mit ausdrücklicher Erlaubnis durchführen. Diese Mindmap ist Lernmaterial, keine Diagnose eines konkreten Systems.

098 · MODERNE WEB-AUTHENTIFIZIERUNG

OAuth 2.0 regelt delegierte Zugriffsberechtigungen. OpenID Connect ergänzt OAuth 2.0 um eine standardisierte Anmeldung und Aussagen über die Identität. Die Protokolle haben unterschiedliche Aufgaben: Ein Access-Token dient dem Zugriff auf Ressourcen; ein ID-Token übermittelt Informationen zur Authentifizierung.

- OAuth 2.0: Autorisierung; OIDC: Authentifizierung auf OAuth-2-0-Basis.
- Prüfe Tokens und Anfragen nach den Regeln des jeweiligen Protokolls.



MERKSATZ
Ein Access-Token ist nicht automatisch ein Nachweis der Benutzeridentität.

1 · ROLLENVERTEILUNG



Vier Rollen wirken zusammen. Die Bezeichnungen benennen Aufgaben; eine Person oder ein System kann in einer Anwendung unterschiedliche Aufgaben übernehmen.

- **Resource Owner:** kann Zugriff auf geschützte Ressourcen gewähren.
- **Client:** fordert im Namen des Resource Owners Zugriff an.
- **Authorization Server:** authentifiziert den Nutzer und stellt Autorisierungsnachweise oder Tokens aus.
- **Resource Server:** stellt die geschützte API oder Ressource bereit und prüft das Access-Token.

2 · AUTHORIZATION CODE FLOW MIT PKCE



Der Client leitet den Nutzer zum Authorization Server. Nach Anmeldung und Zustimmung erhält der Client über die registrierte Weiterleitungsadresse einen kurzlebigen Authorization Code. Der Client tauscht diesen am Token-Endpunkt gegen Tokens ein. PKCE bindet den Austausch an eine zuvor erzeugte Code-Challenge.

- Client erzeugt Code-Verifier und daraus eine Code-Challenge; die Challenge geht mit der Autorisierungsanfrage mit.
- Beim Token-Austausch sendet der Client den Verifier; der Server prüft ihn gegen die Challenge.
- PKCE erschwert das Einlösen eines abgefangenen Codes durch einen anderen Akteur.
- Auch vertrauliche Clients sollten PKCE verwenden; PKCE ersetzt weder Client-Authentifizierung noch sichere URI-Prüfung.

3 · OIDC: ID-TOKEN UND ACCESS-TOKEN



OpenID Connect verwendet das ID-Token als signierten JWT mit Aussagen über die Authentifizierung des Nutzers für den Client. Das Access-Token berechtigt gegenüber einem Resource Server zu Zugriffen; Format und Inhalt können je nach Implementierung unterschiedlich sein.

- ID-Token: Der Client prüft unter anderem Signatur, Aussteller (iss), Zielgruppe (aud) und Ablaufzeit (exp).
- Nonce im ID-Token muss zur zuvor gesendeten Nonce der Anmeldeanfrage passen.
- Access-Token: an die vorgesehene API und deren Prüfregeln gebunden.
- Ein ID-Token ist kein Ersatz für ein Access-Token; das Access-Token ist nicht automatisch ein ID-Token.

WICHTIG
Ein JWT ist lediglich ein Format. Seine Lesbarkeit macht seine Angaben nicht vertrauenswürdig; prüfe Signatur und relevante Ansprüche.

4 · JWT: AUFBAU UND SIGNATUR



Ein kompakter JWT besteht gewöhnlich aus drei Base64url-kodierten Teilen: Header, Payload und Signatur, getrennt durch Punkte. Die Kodierung ist keine Verschlüsselung. Die Signatur ermöglicht Integritäts- und Herkunftsprüfung, sofern der Verifizierer Algorithmus und Schlüssel korrekt festlegt und prüft.

- Header: enthält typischerweise Metadaten wie den Algorithmus (alg).
- Payload: enthält Claims, also Aussagen; sie sind ohne Verschlüsselung grundsätzlich lesbar.
- Signatur: wird über die vorgesehenen Teile mit einem vereinbarten Verfahren erzeugt und geprüft.
- Claims wie iss, aud und exp müssen gemäß Anwendung und Protokoll validiert werden.

5 · JWT-ANGRIFFE UND ABWEHR



Fehler bei Algorithmuswahl, Schlüsselbehandlung oder Anspruchsprüfung können Token-Validierung unterlaufen. Der Verifizierer muss eine sichere, ausdrücklich erlaubte Konfiguration verwenden und Eingaben als nicht vertrauenswürdig behandeln.

- Algorithmus-Verwechslung oder unsichere alg-Behandlung: akzeptiere keine unerwarteten oder deaktivierten Algorithmen; verwende eine feste Allowlist.
- Signaturfälschung: prüfe die Signatur mit dem vertrauenswürdigen Schlüssel und dem vorgesehenen Verfahren.
- Key Injection: vertraue Schlüsselhinweisen wie jku oder eingebetteten jwk nicht ungeprüft, beziehe Schlüsseel nur aus vertrauenswürdiger Konfiguration bzw, einem validierten Aussteller-Schlüsselsatz.
- Zusätzlich Aussteller, Zielgruppe, Ablaufzeit und weitere erforderliche Claims prüfen.

ABWEHRPRINZIP
Dekodieren ist nicht Validieren. Verarbeite ein Token erst nach erfolgreicher Prüfung als vertrauenswürdig.

6 · REVOKATION UND INTROSPECTION



Ein signiertes JWT kann offline geprüft werden und bleibt bis zum Ablauf verwendbar, sofern die Anwendung keine zusätzliche Widerrufsprüfung vorsieht. Introspection ist ein serverseitiger OAuth-Endpunkt, an dem ein berechtigter Client den Status eines Tokens abfragen kann. Widerruf und Introspection sind verwandte, aber unterschiedliche Mechanismen.

- Kurze Token-Laufzeiten begrenzen das Zeitfenster bei kompromittierten Tokens.
- Revokation meldet einem Authorization Server, dass ein Token widerrufen werden soll; die Wirkung hängt von Implementierung und Prüfmodell ab.
- Introspection kann den aktuellen Status eines Tokens liefern; sie erfordert geschützte Kommunikation und Client-Berechtigung.
- Abwägen: ummittelbare Statusprüfung kann zusätzliche Verfügbarkeit, Latenz und Infrastruktur erfordern.

7 · STATE UND NONCE



State und Nonce sind zufällige, an eine einzelne Anmeldung gebundene Werte mit verschiedenen Aufgaben. Sie müssen gegen die ursprüngliche Anfrage geprüft werden und dürfen nicht als austauschbare Parameter behandelt werden.

- State: Client erzeugt einen nicht erratbaren Wert, speichert ihn sitzungsgebunden und vergleicht ihn bei der Weiterleitung zurück; das hilft, CSRF und Antwortverwechsslungen abzuwehren.
- Consce: OIDC-Client sendet sie in der Authentifizierungs-anfrage und prüft die Übereinstimmung im ID-Token, sie bindet die Antwort an die Anfrage.
- Werte sicher erzeugen, kurzlebig halten und nach erfolgreicher Verwendung verwerfen.
- PKCE ergänzt diese Schutzmaßnahmen, erstezt State oder OIDC-Nonce aber nicht pauschal.

8 · REDIRECT-URIS UND CONSENT



Weiterleitungsziele und Zustimmungsdialoge sind zentrale Sicherheitsgrenzen. Der Authorization Server und die Anwendung sollten Weiterleitungen eng begrenzen und Einwilligungen verständlich sowie zweckgebunden darstellen.

- Redirect-URI exakt gegen vorab registrierte Werte prüfen; keine offenen Weiterleitungen oder unsicheren Wildcards zulassen.
- Nur erforderliche Scopes anfordern und die Berechtigungen klar benennen.
- Consent nicht als Sicherheitsgarantie missverstehen: Nutzer können irreführende Anfragen dennoch bestätigen.
- Tokens, Codes und sensible Daten nicht unnötig in URLs, Logs oder Browser-Verlauf offenlegen.

099 · PASSWORTLOSE AUTHENTIFIZIERUNG

FIDO2 verbindet WebAuthn im Browser mit CTAP zwischen Gerät und Authenticator. Statt ein gemeinsames Kennwort beim Dienst einzugeben, bestätigt ein Authenticator eine Anmeldung mit einem kryptografischen Schlüsselpaar.

- Öffentlicher Schlüssel: beim Dienst registriert.
- Privater Schlüssel: bleibt beim Authenticator oder in dessen verwalteter Sicherung.
- Die acht Knoten zeigen Bausteine, Ablauf und Grenzen.



1 · FIDO-ALLIANZ UND MOTIVATION

Die FIDO Alliance entwickelt Standards für starke, einfachere Anmeldung. FIDO-Verfahren verringern die Abhängigkeit von wiederverwendbaren Passwörtern, die bei Credential Stuffing aus geleakten Zugangsdaten ausprobiert werden.



- Credential Stuffing nutzt geleakte Kombinationen aus Benutzername und Passwort erneut.
- Bei FIDO meldet sich der Authenticator mit einem dienstspezifischen Schlüssel an.
- Weniger Passwortwiederverwendung bedeutet weniger verwertbare Zugangsdaten für diesen Angriff.

2 · WEBAUTHN API UND BROWSER

WebAuthn ist eine Web-API, über die eine Website Anmeldedaten erstellen und verwenden kann. Der Browser vermittelt zwischen Website, Betriebssystem und Authenticator; die Verfügbarkeit hängt von Plattform, Browser und Authenticator ab.



- Registrierung: Website fordert neue Anmeldedaten an.
- Anmeldung: Website sendet eine frische Challenge zur Signatur.
- Der Browser prüft Kontext und Origin und gibt der Website eine Antwort zurück.
- Die Website muss die Antwort serverseitig prüfen.

3 · CTAP2

Das Client to Authenticator Protocol (CTAP) ermöglicht die Kommunikation zwischen Client und externem Authenticator. CTAP2 unterstützt FIDO-Anwendungsfälle wie die Nutzung von Public-Key-Anmeldedaten; die konkreten Funktionen variieren nach Gerät und Implementierung.

- Client: etwa Browser oder Betriebssystem.
- Authenticator: etwa ein Sicherheitsschlüssel, der über USB oder NFC verbunden ist.
- CTAP transportiert Anfragen und Antworten; es ist nicht die WebAuthn-API.
- Ein Authenticator kann eine lokale Nutzerbestätigung verlangen.



4. · ROAMING- UND PLATTFORM-AUTHENTICATOR

Ein Roaming-Authenticator ist separat und kann an verschiedenen kompatiblen Geräten verwendet werden. Ein Plattform-Authenticator ist in ein Gerät oder dessen Betriebssystem integriert.

- Roaming: zum Beispiel ein USB- oder NFC-Sicherheitsschlüssel.
- Plattform: zum Beispiel biometrische oder PIN-gestützte Bestätigung am eigenen Gerät.
- Beide können die WebAuthn-Anmeldung unterstützen.
- Wahl nach Gerätezugang, Richtlinien und Wiederherstellungsbedarf treffen.



5 · PASSKEYS UND SYNCHRONISIERUNG

Passkeys sind passwortlose Anmeldedaten auf Basis von Public-Key-Kryptografie. Ein Passkey kann gerätegebunden oder über einen Anbieter zwischen Geräten synchronisiert sein. Sicherung und Verfügbarkeit richten sich nach dem jeweiligen Ökosystem.

- Bei der Anmeldung signiert der Passkey eine frische Challenge.
- Der Dienst registrierte Passkeys können Gerätewechsel erleichtern.
- Cloud-Synchronisierung ist nicht dasselbe wie ein einzelner physischer Sicherheitsschlüssel.
- Kontoschutz des Synchronisierungsanbieters bleibt wichtig.



6 · PUBLIC-KEY-KRYPTOGRAFIE IM FIDO2-ABLAUF

Bei der Registrierung erzeugt der Authenticator ein Schlüsselpaar. Der Dienst speichert den öffentlichen Schlüssel. Bei einer späteren Anmeldung signiert der Authenticator eine vom Dienst bereitgestellte Challenge mit dem passenden privaten Schlüssel.

- Der private Schlüssel wird dem Dienst nicht übermittelt.
- Der Dienst verifiziert die Signatur mit dem registrierten öffentlichen Schlüssel.
- Die Challenge erschwert das Wiederholen einer alten Antwort.
- Ein erfolgreicher Signaturnachweis erstezt nicht die sichere Prüfung der gesamten Anmeldung.



8 · FALLBACK UND KONTOWIEDERHERSTELLUNG

Passwortlose Anmeldung braucht einen durchdachten Weg für verlorene Geräte, gesperrte Konten und Gerätewechsel. Ein schwacher Fallback kann den Schutz der starken Anmeldung unterlaufen.

- Mehrere geeignete Authenticator oder sichere Wiederherstellungsoptionen vorsehen.
- Wiederherstellung angemessen gegen Kontoübernahme absichern.
- Fallbacks nicht pauschal auf leicht erratbare oder wiederverwendete Passwörter reduzieren.
- Nutzer verständlich über Gerätewechsel und Sicherung informieren.



Begriffe

FIDO2 umfasst WebAuthn und CTAP. Passkeys sind eine nutzerfreundliche Form passwortloser Anmeldedaten; je nach Umsetzung können sie zwischen Geräten synchronisiert werden.



Merksatz

PAM und Bastion Hosts

Privileged Access Management (PAM) schützt und steuert Konten mit erweiterten Rechten. Bastion Hosts sind kontrollierte Zugangspunkte für administrative Verbindungen. Ziel: Privilegien begrenzen, Zugriffe nachvollziehbar machen und Risiken reduzieren.

- PAM verwaltet privilegierte Identitäten, Berechtigungen und Sitzungen.
- Ein Bastion Host vermittelt administrative Verbindungen zu geschützten Systemen.



1. PAM-Kernprinzipien und Abgrenzung zu Standard-IAM



IAM verwaltet digitale Identitäten und deren Zugänge allgemein. PAM richtet besondere Kontrollen auf Konten und Sitzungen mit weitreichenden Rechten aus.

- Least Privilege: nur nötige Rechte, nur so lange wie nötig.
- Administrative Konten von Alltagskonten trennen.
- Zugriff nach Identität, Zweck, System und Richtlinie steuern.
- MFA, Freigaben und Protokollierung risikogerecht einsetzen.
- PAM ergänzt IAM; es ersetzt nicht Identitätslebenszyklus oder Zugriffspolitik.

2. Privileged Account Discovery und Inventarisierung



Ermitteln und pflegen Sie privilegierte Konten, damit unbekannte oder verwaiste Zugänge auffallen.

- Quellen abgleichen: Verzeichnisdienste, Server, Datenbanken, Cloud und Anwendungen.
- Konten, Eigentümer, Systeme, Rechte und Verwendungszweck erfassen.
- Menschliche, Dienst-, Notfall- und lokale Konten unterscheiden.
- Gemeinsam genutzte und nicht interaktive Konten besonders prüfen.
- Änderungen regelmäßig abgleichen; Ausnahmen begründen und nachverfolgen.

3. Enterprise Password Vaults und Credential Injection



Ein Vault speichert Zugangsdaten kontrolliert und kann deren Nutzung an Richtlinien binden. Credenital Injection übermittelt ein Geheimnis an eine Sitzung, ohne es dem Anwender im Klartext zu zeigen; die konkrete Umsetzung hängt vom Produkt und Zielsystem ab.

- Zugriff auf Tresor und Geheimnisse eng begrenzen und protokollieren.
- Einzigartige, starke Geheimnisse verwenden; Rotation passend zu Risiko und Technik planen.
- Injektion reduziert Offenlegung, verhindert aber nicht alle Möglichkieten, Geheimnisse abzugreifen.
- Zielsystem, Broker und Endgerät blelben schützenswerte Komponenten.
- Tresorzugriff, Wiederherstellung und Ausfallverfahren sicher planen.

4. Just-in-Time (JIT) Privilegien und Ephemeral Credentials



JIT gewährt erhöhte Rechte erst bei Bedarf und zeitlich begrenzt. Ephemeral Credenitals sind kurzlebige Zugangsdaten, die nicht dauerhaft wiederverwendet werden sollen.

- Antrag an Identität, Aufgabe, Ziel und begrenzte Dauer binden.
- Freigaben und automatische Richtlinien dem Risiko anpassen.
- Rechte nach Ablauf entziehen; tatsächlichen Entzug überprüfen.
- Kurzlebige Daten sicher ausstellen und über TLS schützen.
- JIT senkt dauerhafte Berechtigungen, beseitigt aber Fehlkonfigurationen nicht.

5. Jump Hosts, Bastion Hosts und Session-Proxying



Ein Jump Host dient als Zwischenstation zu anderen Systemen. Ein Bastion Host ist besonders gehärtet und kontrolliert, weil er an einer Sicherheitsgrenze exponiert sein kann. Ein Session-Proxy vermittelt Verbindungen und kann Richtlinien durchsstzen.

- Zugang nur über definierte Pfade und freigegebene Ziele erlauben.
- Host minimieren, patchen, überwachen und administrativen Zugriff beschränken.
- Direkte Umgehungswege zu Zielsystemen vermeiden.
- Protokolle und Anwendungen passend absichern, etwa SSH oder RDP.
- Ein Bastion Host ist kein Schutz, wenn er selbst kompromittiert oder falsch konfiguriert ist.

6. Session Recording, Keystroke Logging und Audit-Trails



Sitzungsaufzeichnungen und Audit-Trails unterstützen Nachvollziehbarkeit und Untersuchung. Keystroke Logging erfasst Eingaben und kann besonders sensible Daten enthalten.

- Zweck, Umfang, Zugriff und Aufbewahrung vorab festlegen.
- Aufzeichnungen vor unbefugtem Zugriff und Manipulation schützen.
- Datenschutz, Transparenz und geltende Vorgaben berücksichtigen.
- Protokolle mit Zeitstempeln, Identität, Ziel und Ergebnis verknüpfen.
- Aufzeichnung ist kein Ersatz für Alarmierung oder aktive Überwachung.

7. Dual-Control und Vier-Augen-Genehmigungen



Dual-Control teilt kritische Aktionen auf mehrere Personen oder Rollen auf. Eine Vier-Augen-Freigabe verlangt eine zweite Prüfung, bevor eine festgelegte Aktion ausgeführt wird.

- Kontrollierte Aktionen und Schwellenwerte eindeutig definieren.
- Antragsteller und Genehmiger aurseichend trennen.
- Identität, Entscheidung, Zetipunkt und ausgeführte Aktion protokollieren.
- Notfallausnahmen begrenzen und nachträglich prüfen.
- Freigaben dürfen nicht zu automatischem Anhicken ohne Kontext werden.

8. Notfallzugriff: Break-Glass-Accounts



Break-Glass-Verfahren ermöglichen eng begrenzten Zugriff, wenn reguläre Wege ausfallen oder eine dringende Wiederherstellung nötig ist.

- Konten und Auslösekriterien vorab festlegen; nur für echte Notfälle verwenden.
- Geheimnisse geschützt und getrennt von regulären Zugangswegen verwahren.
- Zugriff alarmieren und unmitebler protokollieren.
- Nach Nutzung Geheimnis ändern, Aktion prüfen und Ursache dokumentieren.
- Regelmäßig testen, ohne unnötig produktive Notfallkonten offenzulegen.

Sicherheitsgrenze

Notfallzugriff darf nicht zu eimem duaarheft unkontrollierten Hintereingang werden. Verfügbarkeit, Schutz und Überprüfbarkeit gehören gemeinsam zum Verfahren.

Merke

PAM ist ein Kontrollkonzept und Werkzeugbereich; ein Bastion Host ist ein möglicher technischer Zugangspunkt. Beides ergänzt, ersetzt aber nicht, sichere Konten-, Netzwerk- und Systemkonfiguration.

101 · ZTNA UND SDP

Anwendungszugriff nach Identität, Gerät und Kontext prüfen.

1 · SDP-CONTROLLER

Steuert die Richtlinienentscheidung und koordiniert die Freigabe eines Zugriffswegs. Er kann Identitäts- und Gerätesignale berücksichtigen; die Datenübertragung selbst übernnhmen häufig weitere Komponenten.



- prüft Anfrage gegen Zugriffsregeln
- erteilt oder verweigert Freigabe
- Controller und Datenpfad sind nicht zwingend dieselbe Komponente

2 · ZTNA-GATEWAYS

Vermitteln zugelassene Verbindungen zu geschützten Anwendungen. Sie begrenzen den erreichbaren Dienst, statt automatisch das gesamte interne Netz freizugeben.



- Standort kann lokal, im Rechenzentrum oder in der Cloud sein
- Verfügbarkeit und sichere Konfiguration bleiben wichtig
- Gateway ist eine mögliche, nicht die einzige Architekturform

3 · KONTEXTBASIERTE ZUGRIFFSKONTROLLE

Richtlinien können mehrere Signale zusammenführen. Welche Signale verfügbar sind und wie sie gewichtet werden, hängt von der Implementierung ab.



- Identität und Rolle
- angefragte Anwendung und Aktion
- Zeit, Standort oder Risikosignal – sofern erhoben und zulässig
- Ergebnis: erlauben, verweigern oder zusätzliche Prüfung verlangen

4 · LEAST-PRIVILEGE-TUNNELING

Eine Verbindung wird auf die benötigte Anwendung und Berechtigung begrenzt. Ein Tunnel kann den Transport schützen, ersetzt aber weder Authentisierung noch Autorisierung.



- nur erforderliche Ziele und Dienste erreichbar machen
- Berechtigungen passend zur Aufgabe vergeben
- nicht pauschal das ganze Netz über einen Tunnel öffnen

5 · DEVICE-POSTURE-CHECKS

Gerätesignale können in die Zugriffsentscheidung eingehen, etwa ob ein Gerät verwaltet ist oder Mindestanforderungen erfüllt. Solche Prüfungen liefern Hinweise, aber keine absolute Sicherheitsgarantie.



- mögliche Signale: Sicherheitsupdates, Gerätemanagement, Schutzstatus
- Anforderungen müssen zur Anwendung und zum Risiko passen
- bei fehlendem oder veraltetem Signal: nach Richtlinie reagieren

6 · MIKROSEGMENTIERUNG

Teilt Systeme oder Arbeitslasten in kleinere Sicherheitsbereiche und begrenzt erlaubte Verbindungen zwischen ihnen. Sie ergänzt ZTNA, ist aber nicht dasselbe wie der Zugriff auf eine einzelne Anwendung.



- Kommunikation nur zwischen benötigten Komponenten erlauben
- Regeln sichtbar dokumentieren und regelmäßig überprüfen
- Fehlkonfiguration kann Schutzlücken oder Betriebsstörungen erzeugen

7 · IDENTITÄTSBEWUSSTE PROXYS

Ein Proxy kann Anfragen anhand von Identität und Richtlinien vermitteln. Je nach Ausführung sitzt er vor einer Anwendung oder in einem anderen Teil des Verbindungswegs.



- Anwendung statt allgemeiner Netzstandort im Mittelpunkt
- Identität vor Freigabe prüfen
- Protokollierung und Verschlüsselung passend konfigurieren

8 · KONTINUIERLICHE RE-AUTHENTIFIZIERUNG

Eine Sitzung muss nicht bis zum Ende unverändert als vertrauenswürdig gelten. Richtlinien können bei erneuten Anfragen oder relevanten Änderungen eine erneute Prüfung verlangen.



- Anlass und Häufigkeit risikogerecht festlegen
- Änderungen an Identität, Gerät oder Kontext können Neubewertung auslösen
- Sitzungen beenden oder Rechte reduzieren, wenn Anforderungen nicht mehr erfüllt sind

MERKE

Zero Trust bedeutet nicht „niemandem vertrauen", sondern Zugriffe fortlaufend anhand von Richtlinien und Signalen zu prüfen.

102 · SASE UND SSE

Cloud-Sicherheit + WAN-Zugriff nach Richtlinien

ZENTRALE IDEE



SASE verbindet Netzwerkzugriff und Sicherheitsdienste in einer cloudgestützten Architektur. SSE bezeichnet den Sicherheitsdienste-Teil; SD-WAN ergänzt die Vernetzung. Die konkrete Umsetzung unterscheidet sich je nach Anbieter.

- Ziel: Nutzer, Geräte und Anwendungen nach Richtlinien verbinden und schützen.
- Die acht Bausteine sind kombinierbar, aber nicht automatisch ein einziges Produkt.

BEGRIFFE

SASE steht für Secure Access Service Edge, SSE für Security Service Edge. SSE umfasst typischerweise Cloud-Sicherheitsfunktionen; SASE verbindet diese mit WAN-Funktionen wie SD-WAN.

1. SD-WAN-INTEGRATION

Softwaredefiniertes WAN steuert Verbindungen zwischen Standorten, Nutzern und Cloud-Anwendungen anhand zentraler Richtlinien.

- Wählt geeignete Netzwerkpfade nach Richtlinien und Verbindungsqualität.
- Kann direkten Cloud-Zugriff mit Sicherheitskontrollen kombinieren.
- SD-WAN ist Netzwerkfunktion, kein Ersatz für Sicherheitsprüfung.



2. SECURE WEB GATEWAY (SWG)

Ein SWG kontrolliert Webzugriffe und setzt Sicherheitsrichtlinien für den Internetverkehr um.

- Filtert Webanfragen und kann unerwünschte Inhalte oder Ziele blockieren.
- Prüft Webverkehr auf bekannte Bedrohungen, soweit Funktion und Konfiguration dies unterstützen.
- Verschlüsselte Verbindungen begrenzen die Sichtbarkeit, sofern keine geeignete Prüfung eingerichtet ist.



3. CLOUD ACCESS SECURITY BROKER (CASB)

Ein CASB vermittelt Richtlinien zwischen Nutzern und Cloud-Diensten. Je nach Integration arbeitet er etwa über APIs oder als Kontrollpunkt für den Datenverkehr.

- Erkennt und steuert die Nutzung von Cloud-Anwendungen.
- Kann Datenrichtlinien und Zugriffsbeschränkungen unterstützen.
- Sichtbarkeit und Kontrollen hängen von Dienst, Verbindung und Berechtigungen ab.



4. ZERO TRUST NETWORK ACCESS (ZTNA)

ZTNA gewährt Zugriff auf konkrete Anwendungen nach Prüfung von Identität, Gerätezustand und Richtlinien statt pauschalem Vertrauen in ein Netzwerk.

- Authentisierung und Autorisierung sind getrennte Entscheidungen.
- Zugriff wird auf benötigte Anwendungen und Rechte begrenzt.
- Eine erfolgreiche Anmeldung bedeutet nicht automatisch Zugriff auf alle Ressourcen.



5. FIREWALL AS A SERVICE (FWaaS)

FWaaS stellt Firewall-Funktionen als cloudbasierten Dienst bereit und wendet Netzwerkrichtlinien an verteilten Zugriffspunkten an.

- Regeln können Verbindungen anhand von Merkmalen wie Quelle, Ziel und Dienst erlauben oder blockieren.
- Zentrale Richtlinien erleichtern die konsistente Verwaltung verteilter Nutzer.
- Regelqualität und Ausnahmen müssen weiterhin sorgfältig gepflegt werden.



6. DATENSCHUTZ- UND DATENKONTROLLEN

Datenkontrollen erkennen oder begrenzen den Umgang mit sensiblen Informationen. Häufige Funktionen umfassen Data Loss Prevention (DLP); Produktnamen und Umfang variieren.

- Richtlinien können Daten klassifizieren und unerlaubte Übertragungen verhindern oder melden.
- Kontrollen greifen je nach Datenquelle, Kanal und Integrationsart unterschiedlich.
- Fehlalarme und legitime Arbeitsabläufe bei der Richtliniengestaltung berücksichtigen.



8. ZENTRALE CLOUD-MANAGEMENT-KONSOLE

Die Konsole bündelt Richtlinien, Konfiguration und Betriebsübersicht für angebundene Dienste.

- Verwaltet Funktionen wie URL- und Dateiprüfung, Malware-Erkennung und Eindringungsschutz.
- Meldungen sind keine Eindringungsschutz.
- Erkennung ist keine Garantie, jede neue oder gezielt veränderte Bedrohung abzuwehren.
- Aktualisierte Erkennungsregeln und abgestimmte Richtlinien verbessern den Schutz.



103 — XDR-ARCHITEKTUR

XDR verbindet Sicherheitsdaten aus mehreren Ebenen, um zusammenhängende Bedrohungen besser zu erkennen und Untersuchungen sowie Reaktionen zu unterstützen.

1. CROSS-LAYER-TELEMETRIE

Signale aus verschiedenen Bereichen zusammenführen.



- Endgeräte: Prozesse, Dateien, Anmeldungen
- Netzwerk: Verbindungen, DNS, Datenflüsse
- Cloud und Identitäten: Aktivitäten, Zugriffe
- Zeitstempel und Gerätebezug erleichtern die Zuordnung

2. NATIVE VS. OPEN XDR

Unterschiedliche Wege, Daten und Werkzeuge einzubinden.



- Native XDR: häufig eng integrierte Komponenten eines Anbieters
- Open XDR: stärker auf Drittanbieter-Datenquellen und Schnittstellen ausgerichtet
- Bezeichnungen sind nicht einheitlich definiert
- Prüfen: Abdeckung, Datenzugriff, Export und Abhängigkeiten

3. AUTOMATISIERTE KORRELATION

Zusammengehörige Hinweise als mögliche Angriffskette betrachten.



- Ereignisse nach Zeit, Konto, Gerät und Indikator verknüpfen
- Einzelne schwache Signale können gemeinsam aussagekräftiger sein
- Korrelation ist ein Hinweis, kein Beweis
- Datenqualität und Kontext beeinflussen die Zuverlässigkeit

4. ALERT-PRIORISIERUNG

Meldungen nach Risiko und Handlungsbedarf ordnen.



- Mögliche Auswirkungen und betroffene Systeme berücksichtigen
- Kontext wie privilegierte Konten oder kritische Geräte einbeziehen
- Doppelte oder zusammengehörige Meldungen bündeln
- Priorität regelmäßig durch Analystinnen und Analysten überprüfen

5. INCIDENT TIMELINE RECONSTRUCTION

Ereignisse zeitlich ordnen, um den Vorfallverlauf zu rekonstruieren.



- Zeitstempel und Zeitzonen beachten
- Erkennung, Erstaktivität und Folgeereignisse unterscheiden
- Quellen und Unsicherheiten sichtbar halten
- Lücken nicht als Beleg für fehlende Aktivität deuten

6. ORCHESTRIERTE REMEDIATION

Reaktionen über festgelegte Abläufe koordinieren.



- Mögliche Aktionen: Gerät isolieren, Sitzung widerrufen, Indikator blockieren
- Auslöser, Berechtigungen und Freigaben klar festlegen
- Nebenwirkungen und Wiederherstellung berücksichtigen
- Kritische Aktionen zunächst kontrolliert testen

Wichtig
Automatisierte Reaktionen können legitime Arbeit unterbrechen. Umfang und Freigaberegeln müssen zum Risiko passen.

7. THREAT-HUNTING-SCHNITTSTELLEN

Gezielte Suche nach Mustern, die Alerts allein möglicherweise nicht zeigen.



- Abfragen über Telemetriequellen hinweg formulieren
- Hypothesen anhand beobachtbarer Spuren prüfen
- Ergebnisse mit Kontext und Suchzeitraum dokumentieren
- Schnittstellen können Abdeckung und Abfragesprache begrenzen

8. MACHINE-LEARNING-FEEDBACK-LOOPS

Rückmeldungen können Erkennungsmodelle und Regeln verbessern.



- Analystenurteile als gekennzeichnete Rückmeldungen erfassen
- Fehlalarme und übersehene Fälle getrennt auswerten
- Verzerrte oder unvollständige Labels vermeiden
- Änderungen kontrolliert testen und Auswirkungen beobachten

CYBER THREAT HUNTING

Gezielte, hypothesengeleitete Suche nach bislang unentdeckten Bedrohungen in autorisierten Systemen. Hunting ergänzt automatisierte Erkennung; es ersetzt weder Monitoring noch Incident Response.

GRUNDSATZ

Nur in ausdrücklich autorisierten Umgebungen arbeiten. Eine Anomalie ist ein Prüfhinweis, kein Beweis für einen Angriff.

1. HYPOTHESEN FORMULIEREN



- Eine gute Hypothese verbindet eine konkrete Bedrohung mit beobachtbaren Spuren und einer prüfbaren Frage.
- Beispiel: „Falls ein Konto ungewöhnlich viele Hosts per Fernzugriff anspricht, könnten Anmelde- und Prozessprotokolle eine laterale Bewegung zeigen."
- Notiere erwartete Spuren, relevante Systeme, Zeitraum und mögliche harmlose Erklärungen.
- Formuliere so, dass die Hypothese durch Daten gestützt oder abgeschwächt werden kann.

2. BASELINES UND ANOMALIEN



- Baselines beschreiben übliches Verhalten in einem bestimmten Kontext. Abweichungen helfen beim Priorisieren, sind aber allein kein Angriffsnachweis.
- Vergleiche mit passenden Zeiträumen, Rollen, Geräten und Arbeitsabläufen.
- Prüfe neue Prozesse, ungewöhnliche Anmeldezeiten, seltene Verbindungen oder auffällige Datenmengen.
- Berücksichtige Änderungen, Wartung, Zeitzonen, saisonale Muster und unvollständige Protokollierung.

3. HUNTING-MATURITY-MODELL

- Reifegradmodelle ordnen Fähigkeiten zur Bedrohungssuche. Ein verbreitetes Modell unterscheidet fünf Stufen: die Bezeichnungen können je nach Darstellung variieren.
- Initial: kaum strukturierte Daten oder wiederholbare Jagd.
- Minimal und prozedural: verfügbare Daten werden zunehmend systematisch ausgewertet.
- Innovativ: eigene Analysen und Hypothesen prägen die Suche.
- Lege den Reifegrad als Orientierung fest, nicht als Qualitätsbeweis; verbessere Datenzugang, Wiederholbarkeit und Rückkopplung schrittweise.



4. TAKTIKBASIERTE JAGD · ATT&CK

- MITRE ATT&CK ordnet beobachtete Angreiferverhaltensweisen in Taktiken und Techniken. Es ist eine Wissensbasis, kein vollständiges Angriffsmodell und kein Beweis für einen Vorfall.
- Wähle eine relevante Technik und übersetze sie in konkrete, lokal beobachtbare Spuren.
- Prüfe passende Telemetrie und mögliche Blindstellen; eine ATT&CK-Zuordnung ersetzt keine Analyse.
- Dokumentiere Version und Kontext der verwendeten Zuordnung.



5. WERKZEUGE · YARA, SIGMA UND KQL

- Werkzeuge unterstützen unterschiedliche Aufgaben. Ihre Syntax, Datenmodelle und Einsatzbereiche sind nicht austauschbar.
- YARA: Regeln zur Mustererkennung in Dateien oder Speicherinhalten, Ergebnisse brauchen Kontext und Prüfung.
- Sigma: herstellerübergreifendes Format für Erkennungsregeln auf Protokolldaten; die Regel muss für Zielplattform und Felder angepasst werden.
- KQL: Abfragesprache unter anderem für Azure Data Explorer und Microsoft Sentinel; Tabellen und verfügbare Felder bestimmen die Abfrage.
- Teste Regeln kontrolliert auf Fehlalarme, Leistung und erwartete Treffer.



6. DATENQUELLEN UND SENSOREN

- Die Aussagekraft einer Jagd hängt davon ab, welche Ereignisse erfasst, aufbewahrt und zuverlässig zugeordnet werden können.
- Mögliche Quellen: Identitäts- und Anmeldeereignisse, Endpunkttelemetrie, DNS, Proxy, Firewall, Cloud-Audit und Anwendungsprotokolle.
- Prüfe Zeitstempel, Zeitsynchronisierung, Host- und Konto-IDs, Aufbewahrungsfristen sowie Erfassungslücken.
- Beachte Zugriffsrechte, Datenschutz, Datenminimierung und den genehmigten Untersuchungsumfang.



7. DOKUMENTATION UND TRACKING

- Ein nachvollziehbarer Hunting-Datensatz macht Ergebnisse überprüfbar und wiederholbar.
- Festhalten: Hypothese, Umfang, Zeitraum, Datenquellen, Abfragen oder Regelversionen.
- Ergebnisse trennen: Beobachtung, Interpretation, Unsicherheit und verworfene harmlose Erklärungen.
- Status und nächste Schritte notieren: offen, bestätigt, nicht bestätigt oder wegen Datenlücke unklar.
- Nur notwendige Daten speichern und den vereinbarten Zugriffschutz einhalten.



8. RÜCKFÜHRUNG IN ERKENNUNGSREGELN

- Wiederkehrende, belastbare Erkenntnisse können die Erkennung verbessern. Nicht jede Jagdbeobachtung eignet sich unmittelbar als Regel.
- Übersetze bestätigte Verhaltensmuster in eine präzise, erklärbare Erkennungslogik.
- Validiere mit bekannten Beispielen und repräsentativen harmlosen Fällen; prüfe Fehlalarme und Datenabhängigkeiten.
- Versioniere, dokumentiere und überwache die Regel: passe sie bei Änderungen der Datenquelle an.
- Teile Einschränkungen und verbleibende Blindstellen mit dem zuständigen Team.



105 SOC-METRIKEN & BETRIEBSABLÄUFE

Ein Security Operations Center (SOC) überwacht Sicherheitsereignisse, bewertet Alarme und koordiniert Reaktionen. Kennzahlen helfen, Abläufe zu verstehen – sie brauchen klare Definitionen und Kontext.

- Messen, um zu lernen und gezielt zu verbessern; Zahlen allein erklären keine Ursache.

BEGRIFFE KLAR DEFINIEREN

Start- und Endzeitpunkte sowie einbezogene Fälle festlegen. Vergleiche sind nur sinnvoll, wenn Messsregeln und Zeitraum zusammenpassen.

1. MTTD – ZEIT BIS ZUR ERKENNUNG



Mean Time to Detect bezeichnet die mittlere Zeitspanne von einem festgelegten Beginn eines Sicherheitsvorfalls bis zu seiner Erkennung.

- Startpunkt und Erkennungszeitpunkt eindeutig festlegen.
- Mittelwert kann durch einzelee lange Fälle verzerrt werden; Zeitraum und Fallzahl mit betrachten.
- Niedrigere Werte sind nicht automatisch besser, wenn Erkennungen unzuverlässig oder falsch positiv sind.

2. MTTR – ZEIT BIS ZUR REAKTION



MTTR wird unterschiedlich verwendet, etwa für mittlere Reaktions- oder Wiederherstellungszeit. Im SOC muss daher ausdrücklich feststehen, welches Ereignis die Messung beendet.

- Beispiel für Reaktionszeit: vom bestätigten Vorfall bis zur eingeleiteten Maßnahme.
- Wiederherstellungszeit kann dagegen bis zur Rückkehr des Dienstes zum Betrieb reichen.
- Nicht mit MTTD verwechseln: Erkennung und Reaktion sind verschiedene Phasen.



MESSREGEL DOKUMENTIEREN

Die Abkürzung allein ist nicht eindeutig. Schreibe Definition, Start, Ende und betrachtete Fälle zur Kennzahl.

3. ALARM-ERMÜDUNG



Alarm-Ermüdung entsteht, wenn viele wiederkehrende, irrelevante oder wenig aussagekräftige Warnungen die Aufmerksamkeit beanspruchen. Dadurch können wichtige Hinweise leichter übersehen werden.

- Alarmvolumen und Anteil bestätigter relevanter Alarme gemeinsam beobachten.
- Schwellenwerte, Duplikate und Datenquellen prüfen; Änderungen kontrolliert testen.
- Nicht relevante Alarme nicht pauschal abschalten: Auswirkung und Ausnahmen berücksichtigen.

4. TIER-1 BIS TIER-3 ROLLEN



Tiers beschreiben häufig abgestufte Zuständigkeiten. Die genaue Aufgabenverteilung unterscheidet sich je nach Organisation und SOC-Modell.

- Tier 1: Alarme sichten, priorisieren, erste Prüfungen nach Vorgabe durchführen und eskalieren.
- Tier 2: komplexere Untersuchungen und Zusammenhänge analysieren; Maßnahmen koordinieren.
- Tier 3: spezialisierte Analyse, etwa komplexe Bedrohungs- oder Malware-Untersuchung; Prozesse und Erkennungen weiterentwickeln.
- Übergaben und Eskalationskriterien verbindlich festlegen.

5. SCHICHTÜBERGABEN



Eine gute Übergabe macht offene Fälle, Risiken und nächste Schritte für das folgende Team verständlich und nachvollziehbar.

- Fallkennung, Priorität, bekannte Fakten und Zeitangaben festhalten.
- Bereits durchgeführte Prüfungen und Ergebnisse nennen; Vermutungen als solche markieren.
- Nächste Schritte, zuständige Person und Frist angeben.
- Bei dringenden oder laufenden Vorfällen direkte Rückbestätigung vereinbaren.

6. PLAYBOOK-LEBENSZYKLUS



Ein Playbook beschriebt wiederholbare Schritte für einen definierten Alarm- oder Vorfalltyp. Es unterstützt die Arbeit, ersetzt aber keine situationsbezogene Bewertung.

- Entwurf: Auslöser, Voraussetzungen, Rollen und sichere Schritte beschreiben.
- Prüfung: Schritte mit zuständigen Teams und geeigneten Testfällen durchgehen.
- Betrieb: Version, Freigabe, Zugriff und Änderungen nachvollziehbar verwalten.
- Verbesserung: Erkenntnisse aus Fällen und Tests einarbeiten; veraltete Abläufe zurückziehen.

7. QUALITÄTSKONTROLLE (QA)



QA prüft, ob SOC-Arbeit und Dokumentation konsistent, nachvollziehbar und passend zu den vereinbarten Abläufen sind.

- Stichproben nach transparenten Kriterien auswählen.
- Bewertung auf Belege, Klassifizierung, Eskalation und Dokumentation stützen.
- Fehlerarten und wiederkehrende Ursachen erfassen; konkrete Verbesserungen ableiten.
- Ergebnisse rückmelden und später prüfen, ob die Änderung geholfen hat.

8. SLA-MANAGEMENT



Ein Service Level Agreement (SLA) legt vereinbarte Serviceziele oder Pflichten fest. SOC-Ziele können etwa Reaktionszeiten nach Priorität betreffen; ihre genaue Bedeutung folgt der jeweiligen Vereinbarung.

- Ziel, Messmethode, Prioritätsstufen und Ausnahmen eindeutig dokumentieren.
- Erfüllung im vereinbarten Zeitraum berichten und Abweichungen einordnen.
- Reaktionszeit nicht mit vollständiger Behebung oder Wiederherstellung gleichsetzen.
- Kennzahlen gemeinsam mit Qualität, Fallaufkommen und Auswirkungen betrachten.

THREAT INTELLIGENCE PLATTFORMEN (TIP)

TIP sammeln, ordnen und teilen Informationen über Cyberbedrohungen, damit Teams sie in ihrem Kontext bewerten und für Schutzmaßnahmen nutzen können. Daten sind Hinweise, keine Gewissheit: Herkunft, Aktualität und Kontext zählen.



1. STIX und TAXII



STIX beschreibt Cyberbedrohungsinformationen in einem strukturierten, maschinenlesbaren Modell. TAXII ist ein Protokoll für den Austausch solcher Informationen. Sie ergänzen sich, sind aber nicht dasselbe.

- STIX kann etwa Beobachtungen, Indikatoren und Beziehungen zwischen Objekten abbilden.
- TAXII ermöglicht den Austausch über definierte Schnittstellen und Sammlungen.
- Ein gemeinsames Format erleichtert Verarbeitung, ersetzt aber keine Prüfung des Inhalts.

2. MISP-Infrastruktur



MISP ist eine Plattform zum Sammeln, Teilen und Analysieren von Bedrohungsinformationen. Organisationen können Ereignisse, Attribute und Beziehungen verwalten und den Austausch nach ihren Regeln konfigurieren.

- Ereignisse bündeln zusammengehörige Informationen zu einem Vorfall oder Thema.
- Attribute können beispielsweise IP-Adressen, Domains, Hashwerte oder Textbeschreibungen enthalten.
- Freigaben und Verknüpfungen sollten zum Zweck und zur Vertrauensbeziehung passen.

3. Indikatorenverwaltung: IoC und TTP



Ein Indikator für eine Kompromittierung (IoC) ist ein beobachtbarer Hinweis, der mit einer möglichen Bedrohung verbunden ist. Taktiken, Techniken und Prozeduren (TTP) beschreiben typische Ziele und Vorgehensweisen von Angreifern.

- Beispiele für IoCs: verdächtige Domain, Datei-Hash oder IP-Adresse.
- TTPs beschreiben Verhalten und können länger relevant bleiben als einzelne Infrastruktur-Indikatoren.
- Ein einzelner IoC ist kein Beweis für einen Angriff; Kontext und weitere Signale sind nötig.

4. Vertrauensmodelle und TLP



Vertrauen entsteht nicht automatisch durch den Empfang eines Feeds. Traffic Light Protocol (TLP) kennzeichnet, wie Informationen weitergegeben werden dürfen; es bewertet nicht deren Wahrheit.

- TLP:CLEAR erlaubt Verteilung ohne besondere Einschränkung, sofern keine andere Regel entgegensteht.
- TLP:GREEN ist für eine Gemeinschaft bestimmt; außerhalb dieser Gemeinschaft nicht weitergeben.
- TLP:AMBER beschränkt die Weitergabe auf berechtigte Empfänger nach dem Need-to-know-Prinzip.
- TLP:RED ist nur für die ausdrücklich adressierten Empfänger bestimmt.

Wichtig
TLP regelt die Verbreitung, nicht die Verlässlichkeit oder technische Gefährlichkeit einer Information.

5. Feed-Ingestion und Normalisierung



Bei der Feed-Ingestion werden Datenquellen kontrolliert eingelesen. Normalisierung überführt unterschiedliche Feldnamen und Darstellungen in ein einheitliches internes Modell.

- Dokumentiere Quelle, Abrufzeit, Lizenz- und Nutzungsbedingungen sowie bekannte Einschränkungen.
- Prüfe Schema, Datentypen und fehlerhafte oder unerwartete Einträge.
- Bewahre Herkunft und Originalwerte nach Möglichkeit nachvollziehbar auf.
- Lege Ablaufzeiten fest: ein alter Indikator kann unbrauchbar oder irreführend werden.

6. Threat Feeds und Deduplizierung



Mehrere Feeds können denselben Indikator in unterschiedlicher Schreibweise oder mit abweichendem Kontext enthalten. Deduplizierung reduziert redundante Einträge, darf aber wichtige Unterschiede nicht verschleiern.

- Vergleiche normalisierte Werte und passende Typen, nicht nur ähnliche Textzeichenfolgen.
- Erhalte Quellen, Zeitstempel, Konfidenz und Kontext jedes Beitrags.
- Fasse gleichartige Werte zusammen, ohne unterschiedliche Beobachtungen als identisch auszugeben.
- Behandle fehlende oder widersprüchliche Angaben ausdrücklich.

7. CTI-Scoring



Ein Score hilft beim Priorisieren, ist aber eine kontextabhängige Einschätzung und keine objektive Gewissheit. Ein brauchbares Modell macht Kriterien und Unsicherheit sichtbar.

- Berücksichtige Aktualität, Quellenqualität, beobachtete Häufigkeit und Relevanz für die eigene Umgebung.
- Trenne Konfidenz in die Aussage von der möglichen Auswirkung eines Treffers.
- Dokumentiere Begründung und Zeitpunkt; überprüfe Regeln anhand eigener Beobachtungen.
- Vermeide automatische Blockierung allein aufgrund eines ungeprüften Scores.

8. SIEM- und EDR-Anbindung



Ein SIEM korreliert und analysiert Sicherheitsereignisse aus verschiedenen Quellen. Ein EDR erfasst und untersucht Aktivitäten auf Endgeräten und unterstützt Reaktionen. TIP-Daten können beide Systeme anreichern.

- Überführe Indikatoren in passende Erkennungs- oder Suchregeln und berücksichtige Ablaufzeiten.
- Prüe Treffer mit lokalen Protokollen und weiteren Signalen, bevor du reagierst.
- Begrenze automatisierte Maßnahmen durch Richtlinien, Tests und nachvollziehbare Freigaben.
- Miss Nutzen und Fehlalarme und passe die Anbindung an die Umgebung an.

107 · INCIDENT COMMANDER UND KRISENKOMMUNIKATION

ZENTRUM · FÜHRUNG IM SICHERHEITSVORFALL

Ziel: Lage verstehen, Entscheidungen koordinieren und verlässlich kommunizieren. Der Incident Commander (IC) führt den Prozess; technische Teams untersuchen und beheben den Vorfall. Rollen und Meldewege richten sich **nach dem Krisenplan der Organisation.**



1 · ROLLENPROFIL INCIDENT COMMANDER



Der IC hält den Überblick und sorgt für abgestimmte Entscheidungen. Er muss nicht selbst die technische Ursache analysieren.

- Ziele, Prioritäten und Zuständigkeiten festlegen.
- Entscheidungen, Begründung und offene Fragen protokollieren.
- Eine Person als Stellvertretung benennen; Übergaben ausdrücklich festhalten.

2 · TRIAGE UND LAGEBEURTEILUNG



Zuerst klären, was bekannt ist und was noch geprüft werden muss. Frühe Einschätzungen als vorläufig kennzeichnen.

- Welche Systeme, Konten oder Daten sind betroffen? Seit wann, und welche Auswirkungen sind beobachtet?
- Belege sichern und Zeitpunkte dokumentieren; Eindämmung mit Beweissicherung abstimmen.
- Nächste Lagebesprechung und Verantwortliche für offene Prüfpunkte festlegen.

3.· ESKALATIONSKRITERIEN



Vorab definierte Schwellen helfen, passende Unterstützung rechtzeitig einzubeziehen. Die konkrete Einstufung folgt dem organisationsinternen Plan.

- Mögliche Kriterien: Ausbreitung, Ausfall kritischer Dienste, sensible Daten, privilegierte Konten oder Sicherheitsrisiko für Menschen.
- Bei Überschreiten der Schwelle: zuständige Leitung, IT-Betrieb, Datenschutz und weitere benannte Rollen alarmieren.
- Unsicherheit oder neue Auswirkungen erneut bewerten; Eskalation und Zeitpunkt festhalten.

4.· OUT-OF-BAND-KOMMUNIKATION



Wenn E-Mail oder Chat kompromitiert sein könnten, für die Krisenkoordination einen getrennten, zuvor festgelegten Kommunikationsweg nutzen.

- Kanal und Kontaktliste über einen unabhängigen, bekannten Weg bestätigen.
- Zugriff beschränken; sensible Details nur mit den benötigten Personen teilen.
- **Keine Links oder Kontaktangaben** aus verdächtigen Nachrichten als Vertrauensnachweis verwenden.

5 · STAKEHOLDER-BRIEFING



Kurz, abgestimmt und mit klarer Trennung zwischen Fakten, Annahmen und offenen Fragen berichten.

- Was ist passiert? Was ist bestätigt, was wird noch untersucht?
- Welche Auswirkungen und Maßnahmen sind bekannt? Wer entscheidet und wann folgt das nächste Update?
- Empfänger passend auswählen: Leitung, Technik, Datenschutz, Recht, Kommunikation und betroffene Teams.

6.· GESETZLICHE MELDEPFLICHTEN

Mögliche Melde- und Benachrichtigungspflichten früh mit Datenschutz- und Rechtsverantwortlichen prüfen. Nicht jeder Sicherheitsvorfall löst dieselben Pflichten aus.

- **DSGVO:** Bei einer Verletzung des Schutzes personenbezogener Daten prüft der Verantwortliche die Voraussetzungen und Fristen; die Aufsichtsbehörde ist grundsätzlich binnen 72 Stunden zu benachrichtigen, sofern die Verletzung voraussichtlich ein Risiko für Rechte und Freiheiten mit sich bringt.
- **NIS-2:** Pflichten hängen unter anderem von Anwendungsbereich, Rolle und nationaler Umsetzung ab. Für erfasste Einrichtungen sieht die Richtlinie gestufte Meldungen vor. die konkrete Anwendbarkeit fachkundig prüfen.
- **Zeitpunkt der Kenntnis,** Sachstand, Entscheidungen und erfolgte Meldungen nachvollziehbar dokumentieren.

WICHTIG

Diese Mindmap ist Lernmaterial, keine Rechtsberatung. Zuständigkeit, nationale Umsetzung und aktuelle Vorgaben im konkreten Fall prüfen.

7 · PRESSEARBEIT



Externe Aussagen über einen benannten Kommunikationskanal bündeln. Keine Spekulation und keine ungeprüften Details verbreiten.

- Sprecherrolle und Freigabeweg festlegen; Datenschutz und laufende Ermittlungen berücksichtigen.
- Nur bestätigte Informationen, Auswirkungen und nächste Informationsschritte nennen.
- Anfragen und veröffentlichte Aussagen mit Zeitstempel dokumentieren.

8 · DEBRIEFING



Nach Stabilisierung den Allauf auswerten und Verbesserungen in konkrete Aufgaben übersetzen.

- Zeitlinie, Entscheidungen und Übergaben rekonstruieren; Belege und Wissenslücken berücksichtigen.
- Was funktionierte gut, wo gab es Verzögerungen; oder unklare Zuständigkeiten?
- Maßnahmen mit Verantwortlichen und Termin festlegen; Krisenplan und Kontaktwege aktualisieren.

108. DIGITAL FORENSICS READINESS

Spuren nachvollziehbar sichern und untersuchen.

ZENTRALER ÜBERBLICK

Forensische Einsatzbereitschaft schafft Bedingungen, um digitale Spuren nach einem Sicherheitsvorfall nachvollziehbar zu sichern und auszuwerten. Beweismittelsicherung schützt Herkunft, Integrität und dokumentierten Umgang mit den Daten; sie garantiert allein keine rechtliche Verwertbarkeit.



EINORDNUNG
Diese Mindmap vermittelt Lerninhalte, keine Rechtsberatung. Anforderungen an Beweise und Verfahren hängen vom Rechtsraum und Einzelfall ab.

1. CHAIN OF CUSTODY

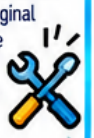
Lückenlose Aufzeichnung, wer ein Beweismittel wann, wo und zu welchem Zweck übernommen, verwahrt, oder weitergegeben hat.



- Eindeutige Kennung und Beschreibung vergeben
- Übergaben mit Zeit, Beteiligten und Zweck protokollieren
- Verwahrung und Zugriff gegen unbefugte Änderung schützen

2. FORENSIC IMAGING: E01 UND RAW

Ein forensisches Abbild erfasst Datenträgerdaten zur Untersuchung, ohne möglichst am Original zu arbeiten. Raw speichert typischerweise die Sektordaten direkt; E01 ist ein verbreitetes Containerformat, das Metadaten und optionale Kompression unterstützen kann.



- Originalzustand und verwendete Methode festhalten
- Abbild mit geeignetem Werkzeug erstellen und Ergebnis prüfen
- Format, Werkzeugversion und etwaige Fehler dokumentieren

3. REIHENFOLGE NACH FLÜCHTIGKEIT

Flüchtige Daten können bei Stromverlust oder Systemänderungen verschwinden. RFC 3227 beschreibt Leitlinien zur Sammlung flüchtiger Informationen; die konkrete Reihenfolge richtet sich nach System, Zugriff und Vorfall.



- Mögliche Reihenfolge: Register und Cache, Arbeitsspeicher, temporäre Dateisysteme, Datenträger
- Netzwerkzustand und laufende Prozesse zeitnah erfassen
- Jeden Eingriff und seine möglichen Auswirkungen dokumentieren

WICHTIG
Die Reihenfolge ist keine starre Universalregel. Eine Maßnahme kann flüchtige Daten verändern; Vorgehen und Risiken abwägen.

4. INTEGRITÄTSPRÜFUNG MIT HASHES

Kryptografische Hashfunktionen bilden Daten auf einen Prüfwert fester Länge ab. Gleiche Eingabedaten ergeben bei gleicher Funktion denselben Hash; schon kleine Änderungen führen typischerweise zu einem anderen Wert.



- Hash des Abbilds nach der Erfassung berechnen
- Bei späterer Prüfung erneut berechnen und Werte vergleichen
- Algorithmus, Zeitpunkt und Werkzeug protokollieren

GRENZE
Ein übereinstimmender Hash stützt die Aussage, dass verglichene Daten gleich sind. Er beweist nicht allein Herkunft, Erfassungsqualität oder rechtliche Zulässigkeit.

6. HARDWARE-WRITE-BLOCKER

Ein Hardware-Write-Blocker soll Schreibzugriffe des Untersuchungsrechners auf das angeschlossene Beweismedium verhindern. Er ist kein Ersatz für kontrolliertes Vorgehen und Prüfung des konkreten Setups.



- Kompatibilität mit Medium und Schnittstelle prüfen
- Funktion vor Einsatz nach dokumentiertem Verfahren testen
- Anschluss, Gerät und Prüfergebnis protokollieren

VORSICHT
Ein Write-Blocker schützt nicht vor jedem Bedienfehler und verhindert nicht automatisch Änderungen durch alle Geräte oder Verfahren.

7. DOKUMENTATIONSSTANDARDS

Eine nachvollziehbare Dokumentation ermöglicht anderen, Untersuchungsschritte und Schlussfolgerungen einzuordnen.



- Datum, Uhrzeit, Zeitzone und ausführende Person festhalten
- System, Zustand, Werkzeuge und Versionen beschreiben
- Schritte, Beobachtungen, Fehler und Abweichungen getrennt notieren
- Originale, Arbeitskopien und Analyseergebnisse klar unterscheiden

8. RECHTLICHE VERWERTBARKEIT

Ob digitale Beweise rechtlich verwertbar sind, hängt unter anderem von geltendem Recht, Zuständigkeit, Erhebungsbefugnis und konkretem Verfahren ab. Technische Sorgfalt allein entscheidet diese Frage nicht.



- Zuständigkeit und erforderliche Befugnisse vor einer Erhebung klären
- Aufbewahrung, Zugriff und Weitergabe nachvollziehbar dokumentieren
- Bei realen Fällen zuständige Rechts- oder Ermittlungsstellen einbeziehen

109 · WINDOWS-EVENT-LOG TIEFENANALYSE

Ereignisse im Kontext lesen

1. PROZESSAUSFÜHRUNG



Neue Prozesse können in der Windows-Sicherheitsüberwachung oder – bei installiertem und konfiguriertem Sysmon – als Prozessereignisse erscheinen.

- Security 4688: Erstellung eines neuen Prozesses; Überwachung muss passend aktiviert sein.
- Sysmon 1: Process Create; enthält typischerweise Prozess- und Elternprozessinformationen.
- Kommandozeilen sind nur enthalten, wenn die entsprechende Richtlinie aktiviert ist.
- Vergleiche Benutzer, Elternprozess, Pfad, Zeit und Kommandozeile mit dem normalen Verhalten.

2. ANMELDEÜBERWACHUNG



Anmeldeereignisse helfen, erfolgreiche und fehlgeschlagene Zugriffe zu untersuchen. Typ und Kontext der Anmeldung sind für die Interpretation wichtig.

- 4624: erfolgreiches Anmelden; Logon Type und Konto beachten.
- 4625: fehlgeschlagenes Anmelden; Status- und Substatusinformationen können Hinweise geben.
- Viele Fehlversuche können auf Fehlkonfiguration oder einen Angriff hindeuten, sind allein aber kein Beweis.
- Zeit, Quellhost, Zielkonto und erwartbare Arbeitsabläufe korrelieren.

3. BESONDERE PRIVILEGIEN



Ereignis 4672 meldet besondere Privilegien, die einer neuen Anmeldung zugewiesen wurden. Das Ereignis ist nicht automatisch verdächtig.

- Prüfe das zugehörige Anmeldeereignis und die betroffene Sicherheitskennung.
- Vergleiche Konto, Host und Zeitpunkt mit administrativen Aufgaben.
- Unerwartete Konten oder ungewöhnliche Systeme können eine vertiefte Prüfung begründen.
- Beachte, dass legitime System- und Administrationsvorgänge solche Einträge erzeugen.

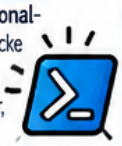
4. ACTIVE-DIRECTORY-ÄNDERUNGEN



Ereignis 5136 protokolliert Änderungen an einem Verzeichnisdienstobjekt, wenn die passende Überwachung und Überwachungsrichtlinie eingerichtet sind.

- Prüfe Objekt, geändertes Attribut, handelndes Konto und Zeitstempel.
- Änderungen an Gruppenmitgliedschaften, Konten oder Richtlinien im Kontext bewerten.
- Änderung und gegebenenfalls zugehörige Werte anhand der verfügbaren Protokolldaten nachvollziehen.
- Die Erfassung setzt geeignete Audit-Einstellungen voraus.

5. POWERSHELL-SKRIPTBLOCKS



Ereignis 4104 gehört zum **PowerShell Operational**-Protokoll und kann den Inhalt verarbeiteter Skriptblöcke erfassen, wenn **Script Block Logging** aktiviert ist.

- Prüfe zusammengehörige Einträge, Zeit, Benutzer, Host und Prozesskontext.
- Lange Skripte können über mehrere Ereignisse verteilt sein.
- Skripttext ist Kontext, nicht automatisch ein Beleg für schädliches Verhalten.
- Achte auf Datenschaitz und sichere Aufbewahrung, da Inhalte sensible Daten enthalten können.

6. AUFGABENPLANUNG

Ereignis 4698 im Security-Protokoll bezieht sich auf das Erstellen einer geplanten Aufgabe, sofern die entsprechende Überwachung aktiv ist.

- Prüfe Aufgabenname, ausführendes Konto und enthaltene Aufgabeninformationen.
- Vergleiche Erstellungszeit und ausführbare Befehle mit genehmigten Änderungen.
- Geplante Aufgaben können legitime Wartung oder unerwartete Persistenz abbilden.
- **Korreliere mit Prozess- und Anmeldeereignissen.**



7. PROTOKOLLLÖSCHUNG



Ereignis 1102 im Security-Protokoll zeigt an, dass das Überwachungsprotokoll gelöscht wurde. Das kann eine administratve Handlung oder ein Hinweis auf Verschleierung sein.

- Prüfe Zeitpunkt, betroffenen Computer und verfügbaren Akteurkontext.
- Suche nach vorherigen auffälligen Aktivitäten und Lücken in der Ereignisfolge.
- Korrelation mit zentral gesammelten Protokollen kann helfen, lokale Lücken einzuordnen.
- Nicht jede Löschung ist böswillig; Änderungen und Wartung berücksichtigen.

8. KERBEROS-EVENTS



Kerberos-Ereignisse geben Einblick in Authentifizierung und Ticketvorgänge in Active-Directory-Umgebungen. Die Bedeutung hängt von Ereignis-ID und Umgebung ab.

- Beziehe die konkrete Ereignis-ID und ihre Felder ein, statt „Kerberos" als Einzelindikator zu bebandeln.
- Konten, Dienstprinzipale, Quellhosts und Zeitverläufe im Zusammenbang prüfen.
- Ungewöhnliche Ticketanforderungen sind Untersuchungshinweise, keine eigenständigen Beweise.
- Mit Anmelde-, Prozess- und Verzeichnisänderungen korrelieren.

EINORDNUNG

Protokollierung und verfügbare Felder hängen von Windows-Version, Richtlinien, Audit-Konfiguration und installierten Komponenten ab.

MINDMAP 110: SYSMON – BEREITSTELLUNG UND KONFIGURATIONSOPTIMIERUNG

1. PROZESSSTARTS & BEFEHLSZEILEN · EID 1



Erfasst Prozesskreationen und liefert Kontext für Prozessketten.

- Abbild, Prozess-GUID, PID und Zeitstempel korrelieren
- Elternprozess und Befehlszeile zur Einerdnung nutzen
- Befehlszeilen können sensible Daten enthalten; Zugriff begrenzen

2. NETZWERKVERBINDUNGEN · EID 3



Protokolliert Netzwerkverbindungen, sofern das Ereignis in der Konfiguration aktiviert ist.

- Quell- und Zieladresse, Ports und Protokoll betrachten
- Prozess-GUID oder PID mit EID 1 verknüpfen
- Verbindungen allein belegen keine schädliche Aktivität

8. MODULARER REGELAUFBAU



Eine Sysmon-Konfiguration strukturiert Ereignistypen und Filterregeln. Community-Konfigurationen, etwa das Schema von SwiftOnSecurity, können als Ausgangspunkt dienen, müssen aber zur eigenen Umgebung passen.

- Version und unterstützte Konfigurationssyntax beachten
- Filter je Ereignistyp nachvollziehbar aufbauen und testen
- Änderungen versionieren; Erfassung und Ausschlüsse kontrolliert überprüfen

KURZ ZUSAMMENGEFASST
Gute Sysmon-Telemetrie verbindet passende Ereignisse mit Prozesskontext, begrenzt Rauschen vorsichtig und bewahrt ausreichend Sichtbarkeit für Untersuchungen.

3. DATEIERSTELLUNG & HASHES · EID 11



Meldet die Erstellung von Dateien im überwachten System.

- Pfad, Prozesz und Zeitstempel prüfen
- Konfigurierte Hashwerte können Dateiidentifikation unterstützen
- Hashwerte sind Indikatoren, kein Beweis für bösartige Absicht

SYSMON IM ERKENNUNGSBETRIEB



Sysmon (System Monitor) protokolliert ausgewählte Systemaktivitäten unter Windows in Ereignissen. Die Daten helfen bei der Analyse und Erkennung; Sysmon selbst ist kein vollständiges Schutz- oder Alarmierungssystem. Aussagekraft und Datenmenge hängen von Version und Konfiguration ab.

- Ereignisse im Protokoll „Microsoft-Windows-Sysmon/Operational"
- Mit Prozess-, Datei-, Netzwerk- und weiteren Kontextdaten untersuchen

EINORDNUNG
Eine Ereignis-ID beschreibt einen Ereignistyp, nicht automatisch einen Angriff. Kontext und weitere Belege sind entscheidend.

4. REGISTRIERUNGSÄNDERUNGEN · EID 12/13



EID 12 erfasst Ereignisse zur Erstellung und Löschung von Registrierungsschlüsseln oder -werten; EID 13 betrifft das Setzen von Registrierungswerten.

- Pfad und auslösenden Prozess einordnen
- Änderung mit erwarteter System- oder Softwareaktivität vergleichen
- EID 14 erfasst Umbenennungen von Registrierungsschlüsseln

7. AUSSCHLÜSSE GEGEN RAUSCHEN



Gezielte Ausschlüsse reduzieren unnötige Ereignisse, können aber auch relevante Telemetrie verbergen.

- Zuerst Ereignisaufkommen und legitime Ursachen boebachten
- Ausschlüsse möglichst eng nach Ereignistyp und Merkmalen formulieren
- Änderungen dokumentieren und Auswirkungen regelmäßig kontrollieren

ABWÄGUNG
Breite Ausschlüsse nach Prozess oder Verzeichnis können Erkennungslücken erzeugen. Nur begründet und so eng wie möglich filtern.

5. REMOTE THREADS · EID 8



Erfasst, wenn ein Prozess einen Thread in einem anderen Prozess erstellt. Dieses Verhalten kann legitim sein oder bei bestimmten Angriffstechniken auftreten.

- Quell- und Zielprozess sowie Startadresse untersuchen
- Prozesskette und zeitlichen Kontext korrelieren
- EID 8 allein beweist keine Injection oder Kompromittierung

6. DNS-ABFRAGEN · EID 22



Protokolliert DNS-Abfragen eines Prozesses, sofern verfügbar und aktiviert.

- Abgefragten Namen mit Prozess und Zeitstempel verbinden
- Antwortinformationen und zugehörige Netzwerkereignisse ergänzen das Bild
- Cache, Systemkontext und Konfiguration können die Sichtbarkeit beeinflussen

111 SIGMA-REGELN

Offenes Format für portable Log-Erkennungslogik

1. YAML-SYNTAX

Regeln werden üblicherweise als YAML formuliert. Die Einrückung strukturiert die Daten; konsistente Leerzeichen und korrekt gesetzte Zeichen sind entscheidend.



- Metadaten: title, id, status, description, author, date, references und tags.
- Erkennung: detection enthält benannte Auswahlblöcke und eine condition.
- Felder und Werte werden als Schlüssel-Wert-Paare geschrieben; Listen können mehrere Werte enthalten.
- Tags wie attack.t1059 können eine ATT&CK-Technik referenzieren.

2. LOGSOURCE

Der Logsource beschreibt, aus welcher Art von Ereignisquelle eine Regel Daten erwartet. Er hilft Werkzeugen, die passende Datenbasis und gegebenenfalls die erforderliche Feldabbildung zu bestimmen.



- Typische Angaben: product, category und service.
- Beispiele: Windows-Sicherheitsereignisse oder Prozess-Erstellung.
- Logsource-Angaben sind keine Garantie, dass ein konkretes System die benötigten Ereignisse tatsächlich sammelt.
- Prüfe Konfiguration, Ereignisabdeckung und Feldnamen der eigenen Umgebung.

3. DETECTION-LOGIK

Unter detection stehen Auswahlblöcke mit Feldbedingungen. Die condition verknüpft diese Blöcke zu einer Erkennungslogik.



- Beispiel: selection: EventID: 4688; condition: selection.
- Kombinationen sind möglich, etwa selection_a and selection_b oder selection_a and not filter_known.
- Modifier verändern den Feldvergleich, zum Beispiel endswith oder contains.
- Ein Modifier wie all kann verlangen, dass mehrere angegebene Werte zutreffen; genaue Semantik anhand der Regel und Werkzeugsdokumentation prüfen.

4. KONVERTIERUNG

Backends übersetzen Sigma-Regeln in Abfragen oder Formate für bestimmte Analyseplattformen. pySigma ist ein erweiterbaree Python-Framework für diese Vararbeitung; Sigmac ist ein älterer, historischer Konvertername.



- Beckend passend zum Zielsystem auswählen und Version sowie unterstüizte Funktionen beachten.
- Feldnamen, Operatoren und Logsource können plattformspezifische Zuordnungen benötigen.
- Konvertierungserfolg bedeatet nicht automatisch, dass die Abfrage fachlich korrekt oder vollständig ist.
- Generierte Abfrage prüfen und mit repräsentativen Ereignisdaten testen.

5. MITRE ATT&CK

ATT&CK-Tags können eine Regel mit Taktiken und Techniken des MITRE-ATT&CK-Wissensbestands verknüpfen. Diese Zuordnung erleichtert Einordnung und Suche, ersetzt aber keine Prüfung der tastächlichen Erkennungslogik.



- Tags folgen häufig einem Namensschema wie attack.execution oder attack.t1059.
- Eine Technik-Zuordnung ist eine kontextuelle Beschreibung, keine Aussage über sichere Angriffserkennung.
- Bei Änderungen von Regel oder ATT&CK-Bezeichnungen Zuordnung erneut prüfen.

6. FALSE-POSITIVE-TUNING

Ein False Positive liegt vor, wenn legitime Aktivität die Regel auslöst. Tuning soll unnötige Alarme verringern, ohne die beabsichtigte Erkennung unbemerkt auszuhöhlen.



- Auslösende Ereignisse untersuchen und legitime Ausnahmen belegen.
- Filter eng auf nachvollziehbare Benutzer, Hosts, Pfade oder Prozesse begrenzen.
- Vorher und nachher mit bekannten Ereignissen testen; Fehlalarme und übersehene Fälle beebachten.
- Ausnahmen dokumentieren und regelmäßig auf Gültigkeit prüfen.

VORSICHT

Breite Ausschlüsse können echte verdächtige Ereignisse verdecken. Jede Ausnahme sollte begründet, überprüfbar und möglicht spezifisch sein.

7. CI/CD-PIPELINE

Eine Pipeline kann Regeländerungen automatisiert auf Syntax, Format, Konvertierbarkeit und Tests prüfen, bevor sie in einen produktiven Erkennungsprozess übernommen werden.

- Änderungen versionieren und nachvollziehbar reviewen.
- YAML und Regelstruktur automatisiert validieren.
- Konvertierung gegen konfigurierte Backends ausführen; Fehler und Warnungen auswerten.
- Tests mit Beispielereignissen sowie kontrollierte Bereitstellung und Rücknahme vorsehen.



8. COMMUNITY-REGELWERKE

Öffentliche Regelwerke bieten Lern- und Wiederverwendungsmöglichkeiten. Regeln können sich in Qualität, Voraussetzungen und Pflegezustand unterscheiden; die Verantwortung für die Nutzung bleibt beim betreibenden Team.



- Quelle, Lizenz, Änderungsdatum und Regelstatus prüfen.
- Logsource, erforderliche Felder und Abhängigketien
- mit der eigenen Umgebung abgleichen.
- Regel isoliert testen und Änderungen vor produktiver Nutzung nachvollziehbar dokumentieren.
- Upstream-Änderungen verfolgen und lokal gepflegte Abweichungen kontrollieren.

112. YARA – REGELBASIERTE SCHADCODE- UND DATEIERKENNUNG

1. REGELSTRUKTUR



Eine Regel besitzt einen Namen und typischerweise die Bereiche *strings* und *condition*. Metadaten sind optional und dienen der Dokumentation.

- Grundform: rule Name { strings: ... condition: ... }
- Strings erhalten Bezeichner wie $a oder $marker.
- Die condition entscheidet, ob die Regel zutrifft.
- Kommentare und klare Namen erleichtern Pflege und Prüfung.

2. TEXT- UND HEXADEZIMAL-STRINGS



Textmuster suchen Bytefolgen in Dateien. Hexadezimalmuster eignen sich, wenn ein Muster als Bytes beschrieben werden soll.

- Beispiel Text: $name = "sample"
- Modifier wie *nocase* verändern die Textsuche; *ascii* und *wide* decken unterschiedliche Byte-Darstellungen ab.
- Hexbeispiel: $sig = { 4D 5A } – typische Anfangsbytes von PE-Dateien, allein kein Schadcode-Beweis.
- Ein spezifisches Muster ist oft trennschärfer als ein häufiges Wort.

3. REGEX & WILDCARDS



YARA unterstützt reguläre Ausdrücke für variierende Textmuster sowie Wildcards in Hexmustern. Flexibilität kann die Zahl der Treffer und den Scanaufwand erhöhen.

- Regex-Strings beginnen üblicherweise mit /.../.
- Hex-Wildcard ?? steht für ein beliebiges Byte; Sprünge können variable Bereiche abbilden.
- Begrenze Muster, wenn möglich, durch feste Präfixe oder Bereiche.
- Teste Varianten gegen bekannte passende und unpassende Dateien.

4. CONDITION-LOGIK UND ZÄHLER



Bedingungen verknüpfen Strings mit boolescher Logik und Mengenregeln. So lässt sich ein einzelner Treffer durch zusätzliche Anforderungen absichern.

- $a and $b verlangt beide Strings; $a or $b mindestens einen.
- any of them bedeutet: mindestens einer der Strings trifft zu.
- 2 of them verlangt mindestens zwei Treffer aus der String-Gruppe.
- Kombiniere Muster nur so, dass die fachliche Bedeutung nachvollziehbar bleibt.

5. PE- & ELF-MODULE



Module stellen formatbezogene Informationen bereit. Das PE-Modul unterstützt Abfragen zu Portable-Executable-Dateien; für ELF-Dateien steht ein eigenes Modul zur Verfügung.

- Modulfelder können Bedingungen zu Headern und Formatmerkmalen ergänzen.
- Prüfe, ob die Eingabe tatsächlich dem erwarteten Format entspricht.
- Modulzugriff ersetzt keine Prüfung des Dateiinhalts.
- Verfügbarkeit und unterstützte Felder richten sich nach YARA-Version und Modul.

FORMAT IST KEIN URTEIL

Ein gültiges PE- oder ELF-Format bedeutet weder „sicher" noch „schädlich". Formatmerkmale sind Kontext für eine Regel.

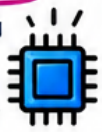
6. PERFORMANCE



Regeln sollten präzise und für den vorgesehenen Bestand testbar sein. Umfangreiche oder breit passende Muster können Scans verlangsamen.

- Wähle ausreichend charakteristische Strings.
- Vermeide unnötig breite Regex- und Wildcard-Muster.
- Nutze passende feste Bedingungen, bevor komplexe Prüfungen nötig sind.
- Miss die Laufzeit auf repräsentativen, autorisierten Testdaten.
- Dokumentiere Regelversion, Datenbasis und bekannte Grenzen.

7. SPEICHERSCANS MIT YARA



YARA kann – abhängig vom eingesetzten Werkzeug und der Integration – auch Speicherbereiche untersuchen. Speicher ist veränderlich und kann unvollständige oder kurzlebige Hinweise enthalten.

- Scans nur mit den nötigen Rechten und im genehmigten Umfang durchführen.
- Treffer mit Prozess-, Zeit- und Erfassungsinformationen dokumentieren.
- Ein fehlender Treffer schließt eine Bedrohung nicht aus.
- Speicherabbilder können sensilde Daten enthalten; Zugriff und Aufbewahrung schützen.

DATENSCHUTZ BEACHTEN

Speicheranalysen können Zugangsdaten, Inhalte oder personenbezogene Daten erfassen. Behandle Ergebnisse entsprechend vertraulich.

8. SANDBOX & EDR



In Analyse- und Abwehrumgebungen kann YARA Regeln als einen Erkennungsschritt ergänzen. Die Einbindung hängt vom jeweiligen Produkt und Ablauf ab.

- Sandbox: Regel gegen bereitgestellte Probe oder Analyseartefakte anwenden.
- EDR: vorhandene Integrationsmöglichkeiten und Berechtigungen prüfen.
- Treffer mit weiteren Signalen korrelieren, statt isoliert zu eskalieren.
- Regeln versionieren, kontrolliert testen und Fehlalarme nachverfolgen.

113 SNORT UND SURICATA

Netzwerkbasierte Intrusion-Detection-Systeme (NIDS) untersuchen beobachtbaren Netzwerkverkehr und melden Muster, die zu Angriffen oder Richtlinienverstößen passen.

Snort und Suricata verwenden Regeln zur Erkennung; ein Treffer ist ein Hinweis zur Prüfung, kein Beweis für einen Angriff.

- NIDS beobachtet Verkehr; ein IDS-Modus alarmiert, ein IPS-Modus kann zusätzlich blockieren.
- Erkennung hängt von Sichtbarkeit, Regelqualität und korrekter Interpretation des Traffics ab.

EINORDNUNG
Verschlüsselte Nutzdaten sind ohne geeignete Entschlüsselung nicht im Klartext prüfbar. Metadaten und unverschlüsselte Teile des Verkehrs können weiterhin sichtbar sein.

1. REGELSYNTAX UND HEADER

Eine Regel legt fest, welcher Verkehr geprüft wird und was bei einem Treffer geschieht.

- Typische Form: Aktion, Protokoll, Quelladresse und -port, Richtung, Zieladresse und -port; danach Optionen in Klammern.
- Header grenzen den Suchbereich ein, etwa TCP-Verkehr von einem Netz zu einem Dienst.
- Aktion und unterstützte Optionen unterscheiden sich je nach Engine und Version.
- Variablen und Adressgruppen können Regeln lesbarer und leichter wartbar machen.

2. CONTENT-MATCHING UND PAYLOADS

Content-Optionen suchen nach Bytefolgen in den für die Regel verfügbaren Paket- oder Anwendungsdaten.

- Ein fester Inhalt kann etwa einen bekannten Protokollmarker oder eine verdächtige Zeichenfolge abgleichen.
- Optionen zur Lage und Ausrichtung begrenzen den Suchbereich und reduzieren zufällige Treffer.
- Mehrere Bedingungen lassen sich kombinieren; die genaue Auswertung richtet sich nach Engine und Regeloptionen.
- Normalisierung und Protokollparser beeinflussen, welche Daten tatsächlich verglichen werden.

GRENZE
Ein einzelner Treffer auf eine Zeichenfolge belegt nicht, dass ein System kompromittiert ist. Kontext und weitere Telemetrie sind wichtig.

3. PCRE IN NIDS-REGELN

PCRE-Optionen ermöglichen reguläre Ausdrücke für komplexere Muster in geeigneten Datenbereichen.

- Reguläre Ausdrücke können variable Schreibweisen oder Musterfolgen beschreiben.
- Sie sind oft aufwendiger als einfache Inhaltsvergleiche und können die Prüfkosten erhöhen.
- Anker, begrenzte Quantifizierer und ein enger Suchbereich helfen, Muster gezielt zu halten.
- Syntax und verfügbare Modifikatoren können zwischen Snort-Versionen und Suricata variieren.

PRAXISREGEL
Zuerst prüfen, ob ein einfacher Inhaltsvergleich genügt. Ausdrükse mit breitem Suchbereich oder starkem Backtracking vermeiden.

4. FLOWBITS UND SESSION-TRACKING

Flowbits können innerhalb eines Verkehrsflusses Zustände markieren, sosass Regeln auf frühere Beobachtungen reagieren können.

- Ein Flow ist typischerweise durch Kommunikcations-endpunkte und Protokollmerkmale abgegrenzt.
- Eine Regel kann einen Zustand setzen; eine spätere Regel kann dessen Vorhandensein voraussetzen oder prüfen.
- Zustandsabhängige Erkennung hilft, mehrstufige Abläufe abzubllden.
- Asymmetrisches Routing, Paketverlust oder abweichende Flusszuordnung können die Beobachtung beeinträchtigen.

5. EVE-JSON-LOGGING IN SURICATA

Suricata kann Ereignisse im EVE-Format als JSON ausgeben. Die Ereignisse lassen sich in Log- und Analysepipelines weiterverarbeiten.

- Je nach Konfiguration enthalten Ereignisse etwa Alarme, Flow-Informationen oder Protokollmetadaten.
- JSON-Felder unterstützen strukturierte Suche und Korrelation mit anderen Datenquellen.
- Protokollierte Ereignistypen, Ausgabepfad und Umfang werden konfiguriert.
- Logs können sensible Netzwerkdetails enthalten; Zugriff und Aufbewahrung sollten angemessen geregelt sein.

AUSWERTUNG
Ein Alarmereignis im Log ist ein Prüfhinweis. Zeitpunkt, Endpunkte, Regel-ID und zugehöriger Kontext helfen bei der Einordnung.

6. HARDWARE-BESCHLEUNIGUNG UND MULTI-THREADING

Durch parallele Verarbeitung und passande Hardware kann die Inspektionsleistung verbassert werden, der Effekt hängt jedoch von Plattform und Arbeitslast ab.

- Multi-Threading verteilt Arbeit auf mehrere CPU-Kerne, soweit Engine, Konfiguration und Traffic dies erlauben.
- Netzwerkkartenfunktionen wie Empfangswarteschlangen können den Paketempfang unterstützen.
- Speicherbandbreite, Paketgröße, Regelkomplexität und Protokollmix beeinflussen den Durchsatz.
- Leistungstests sollten unter repräsentativer Last erfolgen und zugleich Paketverluste und Erkennungsqualität beobachten.

7. INLINE-IPS-MODUS

Im Inline-Betrieb liegt das System im Verkehrsweg und kann abhängig von Engine, Regelaktion und Konfiguration Pakete verwerfen oder anderweitig behandeln.

- Ein NIDS außerhalb des Verkehrswegs alarmiert typischerweise, ohne den Verkehr selbst zu blockieren.
- Inline-IPS kann Angriffe schneller unterbrechen, birgt aber das Risiko von Fehlblockierungen und Verfügbarkeits-problemen.
- Regeln, Ausnahmen und Betriebsverhalten zunächst kontrolliert testen.
- Notfall- und Rückfallverfahren für Ausfall oder Fehlkonfiguration einplanen.

VORSICHT
Blockierende Regeln nur in autorisierten Umgebungen und mit geeignetem Änderungs- und Wiederherstellungsplan aktivieren.

8. SIGNATUR-PERFORMANCE-TUNING

Ziel ist eine passande Balance aus Erkennung, Fehlalarmen und Ressourcenverbrauch.

- Nur benötigte Regelgruppen aktivieren und Regeln auf reale Schutzbedarfe abstimmen.
- Header und Suchbereiche präzisieren; unnötig breite Muster vermeiden.
- Regeländerungen nachvollziehbar testen und Fehlalarme anhand von Kontext prüfen.
- CPU, Speicher, Paketverluste, Alarmvolumen und Erkennungsabdeckung genienssam beobachten.

114 · ZEEK (BRO) – NETZWERKÜBERWACHUNG UND PROTOKOLLANALYSE

ZENTRUM · ZEEK IM ÜBERBLICK



Zeek ist ein passiver Netzwerk-Sicherheitsmonitor: Er beobachtet erreichbaren Netzwerkverkehr und erzeugt daraus strukturierte Ereignisse und Protokolle. Er blockiert Verkehr nicht automatisch. Aussagekraft und Sichtbarkeit hängen davon ab, welche Pakete und Protokollinhalte am Sensor ankommen.

- Metadaten und Ereignisse statt bloßer Paketlisten
- Logs helfen bei Suche, Korrelation und Untersuchung
- Verschlüsselung begrenzt die Sicht auf Anwendungsinhalte

EINORDNUNG

Zeek hieß früher Bro. Es ist ein Netzwerkmonitor mit eigener Skriptsprache, kein Ersatz für Firewall, Endpoint-Schutz oder Paketmitschnitt.

1 · VERBINDUNGS-TRACKING: CONN.LOG



Für beobachtete Verbindungen erzeugt Zeek typischerweise Datensätze mit Ursprung und Ziel, Dienst, Dauer, übertragenen Datenmengen und Verbindungszustand.

- Wichtige Felder: id.orig_h, id.resp_h, id.resp_p, proto, service, duration
- uid verbindet zusammengehörige Ereignisse in mehreren Logs
- Fehlende oder unvollständige Pakete können Zustandsangaben beeinflussen

2 · DNS- UND SSL/TLS-EXTRAKTION



DNS-Ereignisse können angefragte Namen, Antworttypen und Antworten sichtbar machen, wenn der Sensor die betreffenden Pakete sieht. Bei TLS liefern Verbindungs- und Handshake-Ereignisse je nach Version, Konfiguration und beobachtbarem Handshake Metadaten; verschlüsselte Nutzdaten bleiben grundsätzlich verborgen.

- dns.log: Abfragen und beobachtete Antworten
- ssl.log: TLS-bezogene Verbindungs- und Handshake-Metadaten
- DoH, DoT, verschlüsselte ClientHello-Erweiterungen oder fehlende Pakete können Details verbergen

3 · HTTP- UND DATEI-LOGGING



Wo unverschlüsseltes HTTP sichtbar ist, kann Zeek Anfragen und Antworten protokollieren. Datei-Analyseereignisse beschreiben erkannte Dateien, die aus beobachteem Verkehr rekonstruiert werden können; dies bedeutet nicht automatiich, dass eine vollständige Dateikopie gespeichert wird.

- http.log: etwa Host, URI, Methode und Status, soweit sichtbar
- files.log: Datei-Metadaten und Analysezustand
- HTTPS-Inhalte sind ohne geeignete Entschlüsselung am Sensor nicht als HTTP-Nutzdaten lesbar

4 · ZEKE SCRIPTING FRAMEWORK



Zeek-Skripte reagieren auf typisierte Ereignisse und können Werte prüfen, Kontexte anreichern oder eigene Meldungen und Logs erzeugen. Das Framework ermöglicht passgenaue Erkennung, erfordert aber Tests und nachvollziehbare Regeln.

- Ereignis-Handler verarbeiten beobachtete Protokollereignisse
- Richtlinien und Skripte können Erkennungslogik ergänzen
- Zu breite Regeln erzeugen Fehlalarme; zu enge Regeln übersehen Varianten

5 · ZERTIFIKATE UND HASHES



Zeek-Zeripte reagieiisse können Informationes und beobachteten Zertifikaten liefern. Dateianalyse kanen Hashes erkannter Dateien berechnen, sofern die Datei ausreichend rerkennsruiert und erfassen, sofern diese Analyse aktiviert ist.

- Zerihtige Felder: id.orig_h, id.resp_h, id.resp_p, proto, service, duration
- Uietabgleich und Ereigninisgohörige Ereignissnols im mehreem betrachaten
- Zugriff, Transport und Aufbewahrung der Telegmetrie übereerien Varelieren

6 · LATERALE BEWEGUNG UND PROTOKOLLE



Laterale Bewegung bezeichnet Aktivitäten innerhalb eines bereits erreichbaren Netzes, etwa Zugriff auf weitere Systeme oder Dienste. Zeek kann auffällige Kommunikationsmuster sichtbar machen; ein einzelnes Protokoll oder Ereignis beweist jedoch keinen Angriff.

- Auf ungewöhnliche interne Verbindungen und neue Dienstmuster achten
- DNS, SMB, SSH und weitere sichtbare Protekolle im Kontext betrachten
- Zeit, beteiligte Hosts, Baseline und weitere Telemetrie korrelieren

7 · INTEGRATION MIT SIEM UND KAFKA



Zeek-Logs lassen sich über geeignete Export- und Weiterleitungswege an Analyseplattformen übergeben. Kafka kann als Nachrichtenplattform Teil einer solchen Pipeline sein; Schema, Zeitstempel, Feldtypen, Volumen und Ausfallsicherheit müssen zusammenpassen.

- Relevante Logs gezielt auswählen und Felder konsistent abbilden
- Zeitabgleich und Ereignisvolumen für Korrelation berücksichtigen
- Zugriff, Transport und Aufbewahrung der Telemetrie absichern

8 · HINWEISE AUF VERDECKTE TUNNEL



Tunnel können Verkeiss in einem anderen Protokll kapseln oder verschliiern. Zeek kann Auffälligkeiten wie ungewöhnliche DNS-Muster, unerwartete Protokollnutzung oder Kommunikationsrhythmen erfassen, sofern diese Merkmale im beobachteten Verkehr sichtbar sind. Solche Hinweise sind untersuchungswürdig, aber kein alleiniger Beweis.

- Mit einer lokalen Baseline und legitimen Anwendungen vergelichen, genliche innere Website
- DNS-Muster, Zielhäufigkeit, Datenmengen und Zeitverehlten gemeinsam
- Bei TLS- oder sonstiger Verschlüsselung Dleiben und Inhatte mbgleichen

114 · ZEEK (BRO)

Passiver Netzwerkmonitor; strukturierte Ereignisse und Logs; Sichtbarkeit hängt vom beobachteten Verkehr ab.

115 BAS & ADVERSARY EMULATION

Sicherheitskontrollen autorisiert und wiederholt anhand realistischer Angriffstechniken überprüfen.

- Ziel: erkennen, wo Schutz und Erkennung funktionieren – und wo Lücken bleiben.

1. CALDERA FRAMEWORK

Ein MITRE-Projekt zur automatisierten Emulation von Angreiferverhalten.

- Operationen nutzen beschriebene Fähigkeiten („Abilities") und Agenten.
- Techniken orientieren sich an MITRE ATT&CK.
- Umfang, Ziele und erlaubte Aktionen vorab festlegen.

2. ATOMIC RED TEAM KONZEPTE

Kleine, gezielte Tests („Atomics") prüfen einzelne ATT&CK-Techniken.

- Jeder Test hat Voraussetzungen, Ausführung und mögliche Bereinigung.
- Testresultate hängen von Umgebung und Konfiguration ab.
- Nur freigegebene Tests in kontrollierter Umgebung ausführen.

3. PURPLE-TEAMING-METHODIK

Rotes Team simuliert; blaues Team beobachtet und verbessert gemeinsam die Abwehr.

- Ziel und Beobachtungspunkte vereinbaren.
- Test ausführen, Telemetrie und Alarme gemeinsam prüfen.
- Erkenntnisse dokumentieren und Maßnahmen nachtesten.

4. ERKENNUNGSABDECKUNG VALIDIEREN

Prüfen, ob erwartete Signale entstehen und von Regeln oder Prozessen erkannt werden.

- ATT&CK-Technik, Datenquelle, Test und erwartete Erkennung verknüpfen.
- Alarm, Protokollierung und Reaktionsweg getrennt bewerten.
- Keine Erkennung bedeutet nicht automatisch, dass ein Angriff erfolgreich wäre.

5. ENDPOINT-ANGRIFFSSIMULATION

Kontrollierte Tests prüfen Schutz und Sichtbarkeit auf Endgeräten.

- Beispiele: Prozessstarts, Skriptausführung oder Änderungen an Testdateien.
- EDR-/AV-Telemetrie, Alarmierung und Eindämmung beobachten.
- Keine echten Schadprogramme oder dauerhafte Änderungen einsetzen.

6. NETZWERK-EXFILTRATIONSPRÜFUNGEN

Kontrollierte Prüfungen untersuchen, ob ungewöhnlicher ausgehender Datenverkehr auffällt.

- Nur synthetische, harmlose Testdaten und freigegebene Ziele verwenden.
- DNS-, Proxy-, Firewall- und Netflow-Signale im vereinbarten Umfang prüfen.
- Keine realen oder personenbezogenen Daten übertragen.

7. CONTINUOUS SECURITY VALIDATION

Wiederkehrende Tests liefern fortlaufende Hinweise zur Wirksamkeit von Kontrollen.

- Tests risikobasiert planen und Änderungen der Umgebung berücksichtigen.
- Ergebnisse mit Zeit, Umfang, Version und Beobachtungen dokumentieren.
- Automatisierung begrenzen, überwachen und bei unerwartetem Verhalten stoppen.

8. REMEDIATION-ROADMAP

Befunde in priorisierte, überprüfbare Verbesserungen überführen.

- Lücke, Risiko, betroffene Kontrolle und verantwortliche Stelle festhalten.
- Maßnahmen nach Risiko, Aufwand und Abhängigkeiten ordnen.
- Abschluss durch erneuten Test und dokumentierten Nachweis prüfen.

Nur autorisiert testen

Simulationen ausschließlich in ausdrücklich freigegebenen Systemen und vereinbartem Umfang durchführen.

RED-TEAM-KAMPAGNE

Eine genehmigte, realitätsnahe Simulation prüft, ob vereinbarte Sicherheitsziele und Erkennungsmaßnahmen funktionieren. Sie ist keine Erlaubnis für beliebige Angriffe: Umfang, Methoden, Zeitfenster und Abbruchregeln müssen vorab feststehen.

- **Ziel:** Erkenntnisse über Erkennung, Reaktion und Schutz gewinnen.
- **Grundsatz:** nur ausdrücklich autorisierte Systeme und Handlungen.
- **Ergebnis:** nachvollziehbare Beobachtungen und umsetzbare Verbesserungen.

SICHERHEITSGRENZE

Diese Übersicht dient der Planung autorisierter Übungen. Sie enthält keine Anleitung zum Eindringen in fremde Systeme oder zum Verbergen realer Straftaten.

1. PHASENMODELL – ENGAGEMENT LIFECYCLE

Eine Kampagne wird in überprüfbare Phasen gegliedert. Übergänge erfolgen erst, wenn Zuständigkeiten, Freigaben und Sicherheitsbedingungen geklärt sind.

- **Planung:** Ziele, Umfang, Beteiligte und Erfolgskriterien festlegen.
- **Vorbereitung:** Zugänge, Kommunikationswege, Zeitfenster und Schutzmaßnahmen abstimmen.
- **Durchführung:** Szenarien kontrolliert ausführen und Ereignisse protokollieren.
- **Abschluss:** Aktivitäten beenden, Zugänge zurücknehmen und Ergebnisse auswerten.

2. OPSEC ALS DISZIPLIN

Operational Security (OpSec) schützt die Übung und reduziert unbeabsichtigte Auswirkungen. In einer autorisierten Simulation bedeutet das kontrollierte Informationsweitergabe und sichere Arbeitsabläufe – nicht das Umgehen von Aufsicht oder Rechenschaft.

- Informationen nach Need-to-know teilen; Freigaben dokumentieren.
- Übungskommunikation und Eskalationskontakte vor Beginn testen.
- Arbeitsgeräte, Konten und Aufzeichnungen nach Vorgabe schützen.
- Beobachtungen zeitnah und manipulationsarm dokumentieren.

3. INFRASTRUKTUR-TARNUNG UND REDIRECTOR

In einer genehmigter Übung kann ein vorgeschalteter Vermittlungspunkt (Redirector) simulierte Verbindungen kontrolliert weiterleiten. Er ist ein Infrastrukturbaustein, keine Garantie für Anonymität oder Unauffälligkeit.

- Nur freigegebene Domains, Adressen und Dienste verwenden.
- Routen, Betreiberverantwortung und Datenflüsse vorab festhalten.
- Protokollierung und Überwachung mit dem Auftraggeber abstimmen.
- Bei Fehlleitung oder Kontakt mit fremden Systemen sofort stoppen und melden.

4. C2-PROFILIERUNG – MALLEABLE C2

Command-and-Control (C2) bezeichnet Kommunikationswege, über die ein kontrolliertes Testsystem mit einer Übungssteuerung interagiert. Manche Testplattformen erlauben anpassbare Kommunikationsprofile; diese müssen auf Erlaubnis, Erkennbarkeit und Nebenwirkungen geprüft werden.

- Nur freigegebene Testsysteme und Endpunkte einbeziehen.
- Profil und Zweck mit Verteidigungsteam und Regeln der Übung abstimmen.
- Netzwerk- und Sicherheitsprotokolle berücksichtigen; keine Verschleierung als Selbstzweck.
- Kommunikation beenden, sobald eine Sicherheitsgrenze verletzt wird.

5. PAYLOAD-WAFFENFÄHIGKEIT

Eine Payload ist hier ein kontrollierter Testbestandteil, der eine vereinbarte Erkennung oder Reaktion auslösen soll. „Waffenfähigkeit" ist kein Qualitätsziel: Entscheidend sind minimale Wirkung, sichere Rücknahme und ein klarer Nachweis des Testzwecks.

- Vor Einsatz Funktion, Umfang und mögliche Auswirkungen bewerten.
- Bevorzugt ungefährliche Testmarker oder dafür vorgesehene Simulationsfunktionen nutzen.
- Keine destruktiven, persistierenden oder datenverändernden Aktionen ohne explizite Einzelgenehmigung.
- Bereinigungs- und Rücksetzplan vorab festlegen und nach dem Test ausführen.

MINIMALE WIRKUNG

Wenn sich der gewünschte Nachweis mit einem ungefährlichen Marker oder einer Tabletop-Simulation erzielen lässt, ist das die sicherere Wahl.

6. EXFILTRATIONSVERSCHLEIERUNG – SICHER SIMULIEREN

Eine Übung sollte den Erkennungs- und Reaktionsprozess und unerlaubten Datenabfluss testen, ohne echte vertrauliche Daten abzuziehen. Der Nachweis lässt sich mit synthetischen Daten und vereinbarten, beobachtbaren Ereignissen führen.

- Datenquelle, Testdaten, Ziel und Größenlimit schriftlich festlegen.
- Keine personenbezogenen, vertraulichen oder produktiven Inhalte übertragen.
- Erkennung über abgestimmtere Telemetrie und Testmarker prüfen.
- Jeden Transfer nach Vorgabe stoppen, dokumentieren und Testdaten sicher entfernen.

ENCHERER ERSATZ

Synthetische Daten oder ausdrücklisler freigegeener können den Alarmierungsweg prüfen, ohne reale Informationen offenzulegen.

7. RULES OF ENGAGEMENT UND NOTFALL-ABBRUCH

Rules of Engagement (RoE) legen fest, was erlaubt ist und wie auf Risiken reagiert wird. Abbruchkriterien müssen für alle Beteiligten eindeutig und jederzeit anwendbar sein.

- Scope: Systeme, Konten, Standorte, Metheden und Ausschlüsse eindeutig definieren.
- Zeitfenster, Ansperchpaarter, Freigaben und Eskalationsweg dokumentieren.
- Sofort pausieren bei unerwartetem Einfluss auf Verfügbarkeit, Daten oder Dritte.
- Bei Abbruch: Aktivitaten stoppen, Verantwortliche informieren und Zustand sichern.

ENTSCHEIDUNG

Ist die geplante Handlung ausdrücklich freigegeben und innerhalb des vereinbarten Scopes?

- Ja: Sicherheitsbedingungen und Ansperchpartrer sind bestätigt. → Nur die vereinbarte, möglichst risikoarme Handlung im festgelegten Zeitfenster duchfoahren und protokollieren.
- Unklar, außerhäb des Scopes oder eine Abbruchbedingung tritt ein. → Nicht beginnen oder sofort stoppen. Zustand sichern und den benannten Ansprechpartrer über den vereinbarten Eskalatonsweg informeren.

8. ZIELERREICHUNGSANALYSE

Die Auswertung bewertet vereinbarte Ziele anhand beobachtbarer Belege – nicht anhand der Menge ausgeführter Angriffshandlungen. Ergebnisse sollen sowohl Einschränkungen als auch gelungene Erkennung sichtbar machen.

- Je Ziel festhalten: erwartetes Signal, beobachtete Evidenz und Zeitbezug.
- Erkennungs- und Reaktionsabläufe nachvollziehbar beschreiben.
- Ursachen und Lücken als überprüfbare Beobachtungen formulieren, nicht als Schuldzuweisung.
- Maßnahmen priorisieren, Verantwortliche benennen und später erneut testen.

0. LIVING OFF THE LAND (LotL)

Angriffsverhalten mit bereits vorhandenen Systemwerkzeugen. LOLBAS beschreibt Windows-Binaries, -Skripte und -Bibliotheken, die missbraucht werden können; GTFOBins sammelt vergleichbare Hinweise für Unix-/Linux-Programme. Das Vorhandensein eines Tools beweist keinen Angriff: Kontext und Verhalten zählen.

- Leitfrage: Wird ein legitimes Werkzeug ungewöhnlich eingesetzt?



Sicherheitsrahmen

Die Beispiele beschreiben Erkennung und Schutz: Analysiere solche Techniken nur in autorisierten Laboren oder im eigenen Verantwortungsbereich.

1. CERTUTIL & BITSADMIN: DOWNLOADS

Windows-Verwaltungswerkzeuge können in bestimmten Konfigurationen zum Abruf oder zur Übertragung von Dateien verwendet werden. Das kann legitime Administration sein, aber auch eine Tarnung für das Einschleusen von Dateien.

- Prüfe Elternprozess, Benutzer, Zielpfad und Netzwerkziel.
- Achte auf ungewöhnliche Kombinationen aus Kommandozeile, Zeit und Host.
- Bevorzuge erlaubte Softwarequellen und kontrollierte Übertragungswege.



2. MSBuild & InstallUtil: CODEAUSFÜHRUNG

MSBuild und InstallUtil sind legitime Komponenten des .NET-Ökosystems. Unter bestimmten Bedingungen können sie dazu verwendet werden, unerwartete Inhalte auszuführen. Ein Start allein ist kein hinreichender Alarmgrund.

- Erfasse ausführbare Datei, Signatur, Pfad und aufrufenden Prozess.
- Untersuche ungewöhnliche Eingabedateien und Abweichungen vom Softwarebestand.
- Begrenze Ausführung auf benötigte Konten und Arbeitsabläufe.



3. RUNDLL32 & REGSVR32

Diese Windows-Hilfsprogramme unterstützen legitime System- und Komponentenfunktionen. Angreifer können versuchen, solche beknanten Programme als Vermittler für unerwartete Ausführung zu verwenden.

- Korrelation von Prozessstart, Kommandozeile, geladenen Modulen und Kindprozessen.
- Achte auf ungewöhnliche Speicherorte, externe Inhalte oder seltene Aufrufketten.
- Bewerte gegen Baselines statt allein nach dem Programmnamen.



4. PowerShell & WMI

PowerShell automatisiert Windows-Aufgaben; WMI dient unter anderem der Systemverwaltung und Abfrage von Systeminformationen. Beide können administrativ genutzt oder missbraucht werden.

- Nutze zentral konfigurierte PowerShell-Protokollierung und angemessene Richtlinien.
- Untersuche Skriptherkunft, Kontokontext, Befehlsmuster und entfernte Ausführung.
- Beschränke WMI- und Fernverwaltungsrechte nach dem Minimalprinzip.



5. GTFOBins: Linux und sudo

GTFOBins dokumentiert, wie manche Unix-/Linux-Programme in Sicherheitskontexten missbraucht werden können. Entscheidend ist häufig, ob ein Programm mit erhöhten Rechten gestartet werden darf und welche Funktionen es bietet.

- Prüfe sudo-Regeln auf unnötig weit gefasste Programme und Optionen.
- Gib administrative Rechte nur für begrenzte, erforderliche Aufgaben.
- Kontrolliere Änderungen an sudo-Konfiguration und privilegierten Ausführungen.



6. SUID/SGID-BINARIES

SUID und SGID sind Unix-/Linux-Dateirechte, die bei der Ausführung die effektive Benutzer- beziehungsweise Gruppenidentität beeinflussen können. Sie sind nicht automatisch bösartig, erhöhen aber die Bedeutung sorgfältiger Rechtevergabe.

- Inventarisiere privilegierte Dateien und begründe jede Ausnahme.
- Prüfe Eigentümer, Gruppe, Schreibrechte und Herkunft nach Änderungen.
- Entferne unnötige Privilegien; überwache neue oder veränderte Einträge.



7. ERKENNUNG LEGITIMER WERKZEUGE

LotL-Erkennung verbinddt Telemetrie zu Verhalten und Kontext. Eine einzelne Binärdatei oder ein einzelnes Ereignis liefert meist zu wenig Aussagekraft.

- Korreliere Prozessbaum, Identität, Kommandozeile, Datei- und Netzwerkereignisse.
- Vergeliche mit bekannten Administrationsaufgaben und üblichen Zeitmustern.
- Dokumentiere Ausnahmen, überprüfe sie regelmäßig und reduziere Fehlalarme.
- Sichere relevante Protokolle zentral und begrenze deren Manipullierbarkeit.



8. APPLOCKER & WDAC

Anwendungskontrollen können festlegen, welche Anwendungen oder Codearten ausgeführt werden dürfen. Microsoft AppLocker und Windows Defender Application Control (WDAC) unterscheiden sich in Funktionsumfang und Durchsetzungsmodell, Regeln müssen passend geplant, getestet und überwacht werden.

- Starte mit einer nachvollziehbaren Bestandsaufnahme und getesteten Regeln.
- Nutze zunächst einen geeigneten Prüf- oder Auditmodus, sofern verfügbar.
- Erlaube nur erforderliche Software; behandle Ausnahmen nachvollziehbar.
- Überwache Regeländerungen und prüfe, ob Kontrollen wirksam durchgesetzt werden.



Merksatz

Nicht „bekanntes Tool = Angriff", sondern: unerwarteter Kontext + verdächtiges Verhalten + passende Folgeereignisse.

PROCESS INJECTION

Manipulation eines fremden Prozesses, damit Code in dessen Speicher oder Ausführungskontext läuft. Techniken unterscheiden sich darin, wie Speicher verändert, Ausführung ausgelöst und Spuren hinterlassen werden.



1. DLL INJECTION VIA CREATEREMOTETHREAD



Ein Prozess bringt eine DLL in einen anderen Prozess und stößt dort typischerweise einen Thread an, der eine Ladefunktion ausführt.

- Oft: Speicher im Zielprozess reservieren und Pfad zur DLL schreiben.
- CreateRemoteThread kann einen Startpunkt im Zielprozess ausführen lassen.
- Benötigt passende Rechte und ist von Prozessschutz und Architektur abhängig.

2. PROCESS HOLLOWING



Ein Prozess wird gestartet; sein ursprünglicher Programmcode wird ersetzt oder umgeleitet, bevor die veränderte Ausführung fortgesetzt wird.

- Kann legitime Prozessnamen als Tarnung missbrauchen.
- Erkennung betrachtet Startverhalten, Speicherabblid und Abweichungen zur Datei auf dem Datenträger.
- Nicht jede Prozessmanipulation ist Hollowing; Details variieren nach Implementierung.

3. PROCESS DOPPING

Process Dopping bezeichnet eine Technik, bei der eine manipulierte Datei- oder Abschnittsabbildung genutzt wird, um ausgeführten Inhalt anders erscheinen zu lassen als erwartet.

- Ziel ist häufig, Diskrepanz zwischen Speicherinhalt und sichtbarer Datei auszunutzen.
- Abgleich von Speicherabblid, Dateiinhalten und Herkunft kann Hinweise liefern.
- Begriff und konkrete Umsetzung können je nach Quelle unterschieidlich verwendet werden.



4. APC QUEUE INJECTION - EARLY BIRD

Ein Angreifer reiht einen Aufruf in eine Asynchronous Procedure Call (APC)-Warteschlange eines Threads ein. Bei Early-Bird-Varianten wird ein Thread oft vor seiner regulären Ausführung beeniflusst.

- APCs werden unter bestimmten Bedingungen durch den jeweiligen Thread abgearbeitet.
- Die Technik hängt von Threadzustand, Ausführungsreihenfolge und Prozessrechten ab.
- Ungewöhnliche APC-Aktivität und verdächtige Speicherregionen gemeinsam untersuchen.



5. THREAD EXECUTION HIJACKING

Der Ausführungskontext eines vorhandenen Threads wird verändert, sodass er an einer anderen Stelle oder mit verändertem Zustand weiterläuft.



- Beobachtbar sein können ungewöhnliche Kontextwechsel und Sprünge in nicht zugeordneten Speicher.
- Thread-Suspendierung und -Fortsetzung können Teil einer Implementierung sein.
- Legitime Debugging- und Sicherheitssoftware kann ebenfalls Threadzustände beeinflussen.

6. ATOMBOMBING

AtomBombing missbraucht Windows-Atomtabellen als Ablage- oder Übergabemechanismus für Daten und kombiniert dies mit weiteren Schritten, um Codeausführung in einem Zielprozess zu erreichen.



- Der Name beschreibt keine einzelne universelle API-Sequenz.
- Atomtabellen allein führen keinen beliebigen Code aus.
- Zur Analyse die gesamte Kette aus Datenübergabe, Speicher und Ausführung betrachten.

7. HOOKING & MEMORY-MAPPING

Hooking lenkt Funktionsaufrufe über veränderte Sprungpunkte oder Tabellen um, Memory-Mapping bildet Dateien oder Speicherobjekte in einen Prozessadressraum ab; beides kann bei legitimer Software wie auch bei Angriffen vorkommen.



- Unbekannte Hooks, unerwartete Module und veränderte Einstiegspunkte prüfen.
- Gemappte Speicherbereiche nach Herkunft, Berechtigungen und Inhalt einordnen.
- Ein Hook oder Mapping ist für sich genommen kein Beweis für Schadverhalten.

8. ENDPOINT-ERKENNUNG

Erkennung entsteht durch die Kombination von Telemetrie, Kontext und Verhaltensanalyse; ein einzelnes Signal ist selten eindeitg.



- ETW (Event Tracing for Windows) kann je nach Provider Prozess- und Systemereignisse liefern; Abdeckung ist nicht vollständig.
- VirtualAlloc-Aufrufe allein sind häufig legitim. Kontext, Schutzflags, Zielprozess und Folgeverhalten mitbewerten.
- Speicherregionen, Threadstarts, Modulherkunft und Abweichungen zum Dateibild korrelieren.
- Möglichst mit Baselines, Alarm-Triage und kontrollierter Analyse ergänzen.

FÜR DIE VERTEIDIGUNG

Untersuche verdächtige Ereignisse in einer autorisierten Labor- oder Unternehmensumgebung. Die Mindmap ist Lernmaterial, keine Diagnose eines konkreten Systems.

119 · ETW UND EDR-MECHANISMEN



Windows Event Tracing for Windows (ETW) liefert Ereignisdaten an Diagnose- und Sicherheitswerkzeuge. EDR-Produkte verbinden Telemetrie mit Erkennung und Reaktion. Die folgenden Mechanismen zeigen, wo Daten entstehen, wie sie beobachtet werden und warum Manipulationen Erkennung erschweren können.

- ETW ist ein Windows-Instrumentierungs- und Ereignisframework; EDR ist eine Sicherheitslösung, kein einzelner Windows-Mechanismus.
- Die genannten Eingriffe sind Untersuchungsgegenstände der defensiven Analyse, keine Anleitung zum Umgehen von Schutzmaßnahmen.

EINORDNUNG

Begriffe und Verhalten hängen von Windows-Version, Konfiguration, Rechten und Produktimplementierung ab. Die Mindmap beschreibt allgemeine Zusammenhänge, keine vollständige Garantie für Erkennung oder Umgehung.

1 · ETW-ARCHITEKTUR



ETW erfasst Ereignisse von Softwarekomponenten und stellt sie für Diagnose oder Analyse bereit.

- Provider erzeugen Ereignisse; Beispiele sind Windows-Komponenten und Anwendungen.
- Controller konfigurieren und verwalten Tracing-Sitzungen.
- Sessions sammeln und puffern Ereignisse; Einstellungen beeinflussen Umfang und Leistung.
- Consumer lesen oder verarbeiten die Ereignisse, etwa zur Protokollierung oder Analyse.

2 · THREAT-INTELLIGENCE-PROVIDER



Microsoft-Windows-Threat-Intelligence ist ein spezieller ETW-Provider für sicherheitsrelevante Telemetrie. Sein Zugriff ist besonders beschränkt und an privilegierte, vertrauenswürdige Komponenten gebunden.

- Nicht mit einem frei verfügbaren Standard-Provider verwechseln.
- Verfügbarkeit und nutzbare Ereignisse hängen von Plattform und Berechtigungen ab.
- EDR kann solche Signale mit weiteren Quellen korrelieren; ein einzelnes Signal ist nicht gleichbedeutend mit einem bestätigten Vorfall.

3 · USER-MODE API-HOOKING



Ein User-Mode-Hook verändert oder überwacht den Aufrufpfad einer Funktion in einem Prozess, um Aufrufe zu beobachten oder zu beeinflussen.

- EDR-Agenten nutzen Hooks als eine mögliche Telemetriequelle.
- Hooks können durch Updates, Kompatibilität oder andere Instrumentierung beeinflusst werden.
- Ein fehlender Hook beweist weder Schadsoftware noch fehlende Überwachung.

4 · KERNEL-CALLBACKS

Windows-Kernel-Callbacks erlauben registrierten Treibern, auf bestimmte Ereignisse zu reagieren. PsSetCreateProcessNotifyRoutine registriert eine Routine für Prozess-Erstellungs- und -Beendigungsbenachrichtigungen.

- Ein Callback kann Prozessaktivität melden; er ist nicht automatisch eine vollständige Prozesshistorie.
- Registrierung und Entfernung unterliegen API-Vorgaben und Kernel-Rechten.
- Callbacks sind nur eine von mehreren möglichen Beobachtungsquellen.



5 · DIREKTE SYSTEMAUFRUFE

Ein Systemaufruf wechselt vom User Mode in den Kernel Mode, um eine Betriebssystemfunktion anzufordern. Direkte Syscalls bezeichnen Aufrufpfade, die übliche User-Mode-Wrapper umgehen können.

- Das ist kein universeller EDR-Bypass: Erkennung kann auf anderen Telemetriequellen und Verhaltensanalysen beruhen.
- Details und Stabilität variieren nach Windows-Version und Architektur.
- Defensiv prüfen: Prozessverhalten, Speicheränderungen, ungewöhnliche Aufrufmuster und korrelierte Ereignisse.



SICHERHEITSGRENZE

Die Einordnung beschreibt ein Angriffsmuster auf konzeptioneller Ebene. Sie enthält keine Schrittfolge oder ausführbaren Umgehungstechniken.

6 · ETW-MANIPULATION IM SPEICHER

Manipulationen an ETW-bezogenen Daten oder Code im Prozessspeicher können die Erfassung oder Weitergabe von Ereignissen beeinträchtigen.

- Auswirkungen sind abhängig von Provider, Prozess, Rechten und Plattformschutz.
- Ein stiller oder lückenhafter Ereignisstrom ist ein Hinweis zur Untersuchung, aber allein kein Beweis für Manipulation.
- Abwehr: Integritäts- und Verhaltenssignale mit unabhängigen Protokollquellen abgleichen.



7 · NTDLL-UNHOOKING

NTDLL enthält User-Mode-Schnittstellen, die viele Prozesse für Systemfunktionen verwenden. Unhooking bezeichnet den Versuch, dortige Überwachungsänderungen zu entfernen oder zu umgehen.

- Ein solcher Eingriff kann Prozesszustand und erwartete Code-Integrität verändern.
- Er ist nicht gleichbedeutend mit Unsichtbarkeit: Kernel-, ETW- und andere Sensoren können weiterhin Signale liefern.
- Defensiv auf unerwartete Speicherabweichungen und zusammenpassende Verhaltensindikatoren achten.

8 · HARDWARE-BREAKPOINTS



Hardware-Breakpoints nutzen Debug-Register der CPU, um bestimmte Ausführungs- oder Speicherzugriffe zu beobachten. Sicherheitssoftware kann solche Zustände berücksichtigen.

- Anzahl und Verhalten sind architekturabhängig und begrenzt.
- Debug-Zustand kann legitime Analysewerkzeuge ebenso wie verdächtige Aktivitäten begleiten.
- Kontext, Prozessintegrität und weitere Telemetrie gemeinsam bewerten.

120 FIRMWARE-SICHERHEIT

Firmware-Sicherheit schützt den Startpfad, verhindert unbefugte Änderungen und erhält die Integrität der Plattform.

1. UEFI-BOOTFLOW UND SECURE BOOT



Der Bootflow führt vom Einschalten bis zum Start des Betriebssystems. UEFI initialisiert die Plattform und übergibt die Kontrolle schrittweise an Bootkomponenten. Secure Boot prüft Signaturen von UEFI-Treibern und Bootloadern anhand vertrauenswürdiger Kchlüsvel; es schützt nicht automatisch jede Firmware-Komponente.

- UEFI startet mit der SEC- und PEI-Phase, danach folgen DXE und BDS.
- Secure Boot erlaubt nur Bootkomponenten, deren Signatur zur Vertrauenskette passt.
- Ein kompromitierter oder falsch konfigurierter Vertrauensanker kann die Prüfung unterlaufen.

Merke
Secure Boot ist eine Schutzmaßnahme für den Startpfad, kein vollständiger Schutz vor Firmware-Angriffen.

2. SPI-FLASH-SPEICHERARCHITEKTUR



UEFI-Firmware liegt häufig in einem nichtflüchtigen Flash-Baustein, der über SPI angebunden ist. Der Speicher enthält Firmware-Volumes und kann in Bereiche mit unterschiedlichen Zugriffsrechten aufgeteilt sein. Die konkrete Aufteilung und Schutzmechanismen unterscheiden sich je Plattform.

- Firmware-Volumes enthalten Module und Daten für den Startprozess.
- Flash-Descriptor und Chipsatzregeln können festlegen, welche Bereiche lesbar oder beschreibbar sind.
- Schreibschutz kann durch Konfiguration, Hardware oder Plattformfunktionen umgesetzt werden.

3. BOOTKITS UND PERSISTENZVEKTOREN



Ein Bootkit verändert oder ersetzt Komponenten im Startpfad, um vor oder während des Betriebssystemstarts aktiv zu werden. Firmware-Persistenz ist ein möglicher, aber nicht der einzige Weg: Angriffe können auch Bootloader, EFI-Systempartition oder Konfiguration betreffen.

- Mögliche Ziele: Firmware-Volumes, Bootloader, EFI-Systempartition oder Bootkonfiguration.
- Persistenz kann eine Neuisstallation des Betriebssystems überstehen, wenn die betroffene Komponente erhalten bleibt.
- Secure Boot, gemessener Start und regelmäßige Integritätsprüfungen erschweren bestimmte Varianten.

4. SCHWACHSTELLEN IM SYSTEM MANAGEMENT MODE



Der System Management Mode (SMM) ist ein spezieller x86-Betriebsmodus für Firmware-Aufgaben. SMM-Code läuft mit hoher Privilegierung und ist für das Betriebssystem normalerweise nicht direkt sichbar. Fehler in SMM können deshalb schwerwiegende Auswirkungen haben.

- Ein System Management Interrupt (SMI) wechselt in den SMM; der Code läuft typischerweise aus SMRAM.
- Fehlerhafte Eingabeprüfungen oder unzureichende Speicherisolation können Angriffsflächen schaffen.
- Schutzmechanismen wie SMM-Codeprüfung und SMRAM-Isolation sind plattformabhängig.

Einordnung
SMM ist nicht gleichbedeutend mit einer Schwachstelle; entscheidend sind Implementierung und Schutzmaßnahmen.

5. OPTION-ROM-ANGRIFFE



Option-ROMs stellen Firmware-Erweiterungen für Geräte bereit, etwa für Netzwerk- oder Speichercontroller. UEFI kann solche Treiber beim Start laden. Unsichere oder manipulierte Option-ROMs können vor dem Betriebssystem ausgeführt werden.

- Nicht benötigte Option-ROMs und Gerätefunktionen deaktivieren.
- Signaturprüfung und Plattformrichtlinien beachten; Verfügbarkeik variiert je System.
- Geräte- und Firmware-Lieferkette in Inventar und Änderungsprozesse einbeziehen.

6. HARDWARE ROOT OF TRUST



Eine Hardware Root of Trust ist ein besonders geschützter Ausgangspunkt, von dem aus Identität oder Integrität weiterer Komponenten geprüft werden kann. Sie kann beispielsweise einen frühen Startschritt messen oder verifizieren. Das Konzept ist keine Garantie, dass alle späteren Komponenten vertrauenswürdig sind.

- Verifizierter Start blockiert nicht vertrauenswürdige Komponenten, soweit die Vertrauenskette reicht.
- Gemessener Start protokolliert Messwerte, die eine spätere Auswertung ermöglichen.
- TPM kann Messwerte und Schlüsvel schützen; es erstezt keine sichere Firmware-Konfiguration.

7. FIRMWARE-UPDATE-VERIFIKATION



Firmware-Updates sollten aus einer vertrauenswürdigen Quelle stammen und vor der Installation kryptografisch geprüft werden. Signaturprüfung bestätigt, dass das Update zu einem vertrauten Herausgeber passt und nicht verändert wurde; sie beweist nicht automatisch Fehlerfreiheit.

- Updates über Hersteller- oder Organisationsprozesse beziehen.
- Signaturprüfung und gegebenenfalls Versions- oder Rollback-Schutz aktivieren.
- Installation, Neustart und Firmware-Version nachvollziehbar dokumentieren.

Sicher vorgehen
Firmware nur für das passende Gerätemodell und nach dokumentiertem Verfahren aktualisieren.

8. FIRMWARE-FORNSIK



Firmware-Forensik untersucht Firmware-Abbilder, Konfiguration und Startmessungen auf Abweichungen. Ergebnisse sind im Kontext der Plattform zu bewerten: legitime Hersteller- oder Organisationsänderungen können ebenfalls Unterschiede erzeugen.

- Gerät, Firmware-Version, Zeitpunkt und Erfassungsmethode festhalten.
- Abbilder und Protokolle geschützt sichern; Prüfsummen zur Integritätskontrolle verwenden.
- Mit vertrauenswürdigen Referenzen vergleichen und Abweichungen nachvollziehbar dokumentieren.
- Befunde als Hinweise bewerten und mit weiteren Belegen korrelieren.

ADFS-SICHERHEIT

ADFS ermöglicht föderierte Anmeldung: Ein Identitätsanbieter stellt nach erfolgreicher Authentifizierung Sicherheitstoken aus, die ein vertrauender Dienst anhand vereinbarter Regeln prüft. Die Sicherheit hängt von Protokoll, Vertrauensbeziehungen, Schlüsseln, Claims und dem Betrieb der Server ab.

- Ziel: Identitäten sicher über Organisations- oder Dienstgrenzen nutzen.
- Zentrale Fragen: Wer authentifiziert? Wer vertraut dem Token? Welche Rechte folgen daraus?

1. VERBUNDAUTHENTIFIZIERUNG



Bei der Verbundauthentifizierung prüft ein Identitätsanbieter die Anmeldung und gibt ein Token an einen vertrauenden Dienst aus. Beide Seiten müssen die Vertrauensbeziehung und die erwarteten Token-Regeln passend konfigurieren.

- WS-Federation (WS-Fed) und SAML 2.0 sind unterschiedliche Verbundprotokolle.
- SAML-Aussagen können Identitäts- und Authentifizierungsinformationen enthalten.
- Protokollunterstützung und Konfiguration hängen von Anwendung und ADFS-Version ab.
- Nur benötigte Protokolle und Endpunkte aktivieren; veraltete oder unnötige vermeiden.

2. WEB APPLICATION PROXY (WAP)



Der Web Application Proxy kann als veröffentlichter Zugriffspunkt für unterstützte ADFS-Szenarien im Extranet dienen. Er vermittelt Anfragen an die interne Verbundumgebung, ist aber selbst eine sicherheitskritische Komponente.

- WAP im erreichbaren Extranet platzieren; ADFS-Server nicht direkt dem Internet uussetzen.
- Netzwerkzonen und Firewall-Regeln auf notwendige Verbindungen beschränken.
- WAP und ADFS-Kommunikation absichern und den Proxy zeitnah aktualisieren.
- Erreichbarkeit, Zertifikate und Proxy-Konfiguration regelmäßig kontrollieren.

3. TOKEN-ZERTIFIKATE



ADFS verwendet Zertifikate für Token-Signierung und – abhängig von Konfiguration und Vertrauensbeziehung – Token-Verschlüsselung. Die Empfängerseite prüft Signaturen mit dem zugehörigen öffentlichen Schlüssel; Verschlüsselung schützt Token-Inhalte für vorgesehene Empfänger.

- Private Schlüssel streng schützen und Zugriffe darauf begrenzen.
- Gültigkeit, Ablaufdaten, Erneuerung und vertrauende Parteien im Blick behalten.
- Signierung und Verschlüsselung erfüllen verschiedene Schutzziele.
- Zertifikatswechsel geplant durchführen, damit Vertrauenspartner neue Schlüssel übernehmen können.

4. GOLDEN-SAML-ANGRIFF



Golden SAML bezeichnet ein Angriffsszenario, bei dem ein Angreifer einen kompromittierten privaten Token-Signaturschlüssel nutzt, um gefälschte, gültig signierte SAML-Tokens zu erzeugen. Das setzt weitreichende Kompromittierung voraus und ist kein gewöhnlicher Anmeldefehler.

- Mögliche Folge: gefälschte Claims können von vertrauenden Diensten als echt behandelt werden.
- Schlüsselmaterial und privilegierte ADFS-Systeme besonders schützen.
- Ungewöhnliche Token-Aussteller, Claims, Anmeldemuster und Änderungen untersuchen.
- Bei Verdacht Schlüssel- und Vertrauensbeziehungen nach einem abgestimmten Incident-Response-Verfahren behandeln.

5. EXTRANET ACCOUNT LOCKOUT



Extranet Account Lockout kann Anmeldeversuche über den Extranet-Zugriff begrenzen und so das Risiko von Passwort-Raten reduzieren. Zu strenge oder falsch abgestimmelte Schwellenwerte können legitime Nutzer aussperren.

- Funktion und verfügbare Optionen sind versions- und konfigurationsabhängig.
- Schwellenwerte, Zeitfenster und Auswirkungen anhand der eigenen Umgebung festlegen.
- Fehlversuche und Sperrereignisse überwachen; Helpdesk- und Wiederherstellungswege planen.
- Kontosperren ergänzen weitere Maßnahmen wie MFA und starke, enisigartjte Passwörter.

6. CLAIMS-BASIERTE ZUGRIFFSKONTROLLE



Claims sind Aussagen über eine Identität oder Anmeldung, etwa Benutzerkennung oder, Gruppenbezug. ADFS-Regeln können Claims transformieren oder ausgeben, der vertrauende Dienst entschiedet anhand seiner Richtlinien, welche Zugriffe daraus folgen.

- Nur notwendige Claims freigeben und unnötige personenbezogene Angaben vermeiden.
- Aussteller, Claim-Typ und Werte im Kontext der Vertrauensbeziehung bewerten.
- Zugriff neuche, ungewöhnliche Quellorte gestalten; Authentifizierung ist nicht gleich Autorisierung.
- Regeln nachvollziehbar dokumentieren und Änderungen kontrolliert testen.

7. ÜBERWACHUNG UND AUDIT-LOGGING



Protokolle und Überwachung helfen, Authentifizierungsprobleme, Konfigurationsänderungen und verdächtige Muster zu erkennen. Aussagekräftige Ereignisse müssen geschützt, zeitlich zugeordnet und gemäß Aufbewahrungsregeln verwaltet werden.

- ADFS-Audit- und Betriebsprotokollierung passend zu Zweck und Systemversion konfigurieren.
- Zeitquellen synchronisieren und relevante Ereignisse zentral auswerten.
- Auf Fehlversuche, ungewöhnliche Quellorte, Änderungen an Regeln und Zertifikaten achten.
- Protokolle vor unbefugtem Zugriff und Veränderung schützen; sensible Daten nicht unnötig sammeln.

8. WINDOWS-SERVER-HARDENING



ADFS-Sicherheit beginnt auch bei den Windows-Servern und ihren Abhängigkeiten. Eine reduzierte Angriffsfläche, kontrollierte Administration und verlässliche Aktualisierung senken das Risiko einer Kompromittierung.

- Unterstützte Betriebssysteme und aktuelle Sicherheitsupdates einsetzen.
- Nicht benötigte Rollen, Dienste und eingehende Verbindungen deaktivieren oder begrenzen.
- Administrative Konten und privilegierte Zugriffe minimieren und überwachen.
- Backups, Wiederherstellung und Änderungen an ADFS-Konfiguration sicher planen und testen.

EINORDNUNG



Diese Mindmap vermittelt technische Grundlagen. Sie ist keine Diagnose eines konkreten Systems und ersetzt keine risikogerechte Prüfung der eigenen Umgebung.

HYBRIDE IDENTITÄT

On-Premises Active Directory und Microsoft Entra ID arbeiten zusammen. Die Synchronisierung verbindet Identitäten und schafft zugleich sicherheitsrelevante Abhängigkeiten. Schütze Synchronisierungsserver, Konten und Notfallzugänge.

- Ziel: Identitäten kontrolliert zwischen lokalem Verzeichnis und Cloud verwalten.
- Leitfrage: Welche Komponente könnte bei einer Kompromittierung Zugriff auf beide Umgebungen eröffnen?



1. PHS UND PTA



Beide Verfahren ermöglichen die Anmeldung mit lokal verwalteten Identitäten, unterscheiden sich aber darin, wie die Kennwortprüfung erfolgt.

- PHS: Ein abgeleiteter Kennworthash wird synchronisiert; Entra ID kann Anmeldungen in der Cloud prüfen.
- PTA: Die Kennwortprüfung wird über einen lokalen Authentication Agent an die lokale Umgebung weitergeleitet.
- Wäge Verfügbarkeit, Betriebsaufwand und Sicherheitsanforderungen ab: beide Verfahren brauchen sorgfältige Kontrollen.

2. SEAMLESS SSO: ANGRIFFSFLÄCHEN



Seamless Single Sign-On kann die Anmeldung an unterstützten Geräten und Diensten vereinfachen. Die Funktion nutzt Kerberos in der lokalen Umgebung und muss gegen Missbrauch abgesichert werden.

- Behandle das zugehörige Computerkonto und seine Schlüsselmaterialien als schützenswerte Ressourcen.
- Begrenze und überwache, wo und wie SSO genutzt werden kann.
- Prüfe verdächtige Anmeldungen, ungewöhnliche Geräte und Änderungen an der Konfiguration.

3. MSOL-SERVICEKONTO UND BERECHTIGUNGEN



Das Synchronisierungskonto benötigt Berechtigungen, um Verzeichnisdaten auszulesen und die Synchronisierung auszuführen. Übermäßige Rechte erhöhen das Risiko bei einer Kompromittierung.

- Ermittle, welche Konten und Berechtigungen tatsächlich verwendet werden.
- Begrenze Rechte nach dem Prinzip der geringsten Privilegien und kontrolliere Änderungen.
- Schütze Anmeldeinformationen und überprüfe regelmäßig Konten, Rollen und Protokolle.

4. ENTRA-CONNECT-SERVER ABSICHERN



Der Synchronisierungsserver ist eine zentrale Vertrauenskomponente zwischen lokaler Umgebung und Cloud. Seine Kompromittierung kann weitreichende Folgen haben.

- Beschränke administrative Anmeldung und Netzwerkzugriff auf notwendige Perssnen und Systeme.
- Halte Betriebsysstem und Synchronisierungssoftware aktuell; entferne nicht benötigte Dienste.
- Überwache Sicherheitsereignisse, Änderungen und den Zustand des Servers.

5. APPLICATION PROXY: SICHERHEIT

Microsoft Entra Application Proxy kann internen Webanwendungen einen kontrollierten externen Zugriff ermöglichen. Connectoren und veröffentlichte Anwendungen bilden dabei sicherheitsrelevante Komponenten.



- Veröffentliche nur erforderliche Anwendungen und prüfe deren Authentifizierungs- und Zugriffsregeln.
- Schütze und aktualisere Connector-Server; beschränke administrative Zugriffe.
- Überwache Anmeldungen und den Zugriff auf veröffentlichte Anwendungen.

6. DIRECTORY-SYNCHRONISIERUNG UND SHADOW ATTRIBUTES



Synchronisierte Attribute beeinflussen, welche Verzeichnisinformationen in der Cloud verfügbar sind. Unerwartete oder manipulierte Werte können Identitäts- und Zugriffsentscheidungen beeinflussen.

- Dokumentiere, welche Attribute synchronisiert und wo sie verwaltet werden.
- Prüfe unerwartete Änderungen und nachvollziehbare Zuordnungen zwischen lokalen und Cloud-Objekten.
- Beachte, dass genaue Auswirkungen von Konfiguration und eingesetzten Funktionen abhängen.

BEGRIFFSHINWEIS
„Shadow Attributes" ist hier als Hinweis auf sicherheitsrelevante, unerwartete oder missbrauchte Attributwerte gemeint, nicht als einheitlicher Produktmechanismus.

7. PRIVILEGIENAUSWEITUNG ÜBER DAS SYNCHRONISIERUNGSKONTO



Ein Angreifer, der ein privilegiertes Synchronisierungskonto oder den Synchronisierungsserver kontrolliert, kann möglicherweise Identitäten oder Zugriffsbeziehungen beeinflussen. Der konkrete Pfad hängt von Berechtigungen und Konfiguration ab.

- Schütze Synchronisierungskonten und Server mit eingeschränktem Zugriff und starker Überwachung.
- Untersuche Änderungen an Konten, Gruppen und synchronisierten Objekten auf ungewöhnliche Muster.
- Begrenze administrative Rechte und trenne Aufgaben, wo es praktikabel ist.

8. NOTFALLZUGRIFFSKONTEN (BREAK-GLASS)



Notfallzugangskonten helfen, den Zugriff auf die Cloudverwaltung bei Ausfällen oder Problemen mit regulären Anmeldewegen wiederherzustellen. Sie müssen verfügbar und zugleich streng geschützt sein.

- Halte Notfallkonten von alltäglichen Administrationskonten getrennt.
- Schütze Zugangsdaten und prüfe sichere Aufbewahrung sowie Wiederherstellbarkeit.
- Überwache jede Nutzung und teste den Notfallzugang kontrolliert in pessanden Abtsänden.

MERKSATZ
Notfallzugang ist kein unbeaufsichtigter Dauerzugang: Verfügbarkeit, Schutz und Alarmierung gehören zusammen.

WINDOWS LAPS

Zentralisierte Verwaltung lokaler Administratorkennwörter in Windows-Umgebungen: LAPS kann Kennwörter sichern, den Zugriff darauf steuern und sie nach Richtlinie ändern.

1. LEGACY LAPS VS. WINDOWS LAPS



Beide Ansätze verwalten lokale Administratorkennwörter zentral, unterscheiden sich aber bei Integration und unterstützten Sicherungszielen.

- Legacy LAPS nutzt typischerweise Active-Directory-Attribute wie ms-Mcs-AdmPwd und ms-Mcs-AdmPwdExpirationTime.
- Windows LAPS ist in unterstützten Windows-Versionen integriert und kann – abbängig von Version und Konfiguration – Active Directory oder Microseft Entra ID als Sicherungsziel verwenden.
- Vor einer Migration Funktiomen, Richtlinien und bestehende Clients prüfen: beide Lösungen nicht als identisch behandeln.

2. SCHEMA UND ATTRIBUTE



Für die kennwortbezogenen Daten verwendet Legacy LAPS andere Active-Directory-Attribute als Windows LAPS.

- Legacy LAPS: ms-Mcs-AdmPwd enthält das Kennwort; ms-Mcs-AdmPwdExpirationTime enthält den Ablaufzeitpunkt.
- Windows LAPS verwendet eigene Attribute, darunter msLAPS-Password und msLAPS-PasswordExpirationTime.
- Das Active-Directory-Schema muss für Windows LAPS erweitert sein, bevor die entsprechenden AD-Funktionen genutzt werden.
- Schemaänderungen sind geplante Verzeichnisänderungen: Berechtigungen, Auswirkungen und Wiederherstellungsmöglichkeiten vorab berücksichtigen.

3. ENTRA-CLOUD-LAPS



Windows LAPS kann lokale Administratorkennwörter in Microsoft Entra ID sichern, wenn Geräte, Richtlinien und Dienstvoraussetzungen passend konfiguriert sind.

- Cloud-Sicherung ist eine Alternative zum Sichern in lokalem Active Directory; sie ist nicht automatisch für jedes Gerät aktiv.
- Geräteverwaltung, unterstützte Windows-Versiom und zugewiesene Richtlinie müssen zusammenpassen.
- Zugriff auf Kennwörter ist über die vorgesehenen Rollen und Berechtigungen zu beschränken und zu überwachen.

4. KENNWORTVERSCHLÜSSELUNG



Windows LAPS kann Kennwörter bei der Sicherung in Active Directory mit kennwortbezogenen Richtlinien verschlüsseln, sofern die Umgebung die Funktion unterstützt und sie aktiviert ist.

- Verschlüsselung schützt gespeicherte Kennwortdatenkennwortdaten; sie erstazt weder Zugriffskontrollen noch sichere Administration.
- Die Verschlüsselung auf Attributebene ist an die unterstützte Windows-LAPS-Konfiguration gebunden und darf nicht pauschal für jedes Sicherungsziel angenommen werden.
- Schlüssél- und Berechtigungsverwaltung gehören zum Schutzkonzept.

WICHTIG
Nicht mit Transportverschlüsselung verwechseln: Sie schützt die Übertragung, während gespeicherte Kennwortdaten eine eigene Schutzkonfiguration benötigen.

5. ROTATIONSZYKLEN



Richtlinien bestimmen, wann Kennwörter geändert werden und welche Kennwortregeiln gelten.

- Die Ablaufzeit kann eine Rotation auslösen; Windows LAPS kann außerdem nach einer authentifisierten Kennwortverwendung eine Änderung anstoßen, wenn diese Funktion konfiguriert ist.
- Rotationsintervall, Kennwortlänge und Komplexität an die Umgebung anpassen.
- Nach einer Wiederherstellung oder einem Geräteaustausch sicherstellen, dass das neue Kennwort zentral gesichert und abrufbar ist.

6. ACLs UND KENNWORTZUGRIFF



Active-Directory-Berechtigungen steuern, welche Konten **Kennwortattribute lesen oder verwalten dürfen.**

- Nur berechtigte Support- oder Administrationsrollen sollen Kennwörter abrufen können.
- Die Berechtigung zum Zurücksetzen beziehungyemise Verwalten von Kennwörtern ist von der Leseberechtigung zu unterscheiden.
- Berechtigungen auf Organisationseinheiten prüfen und unnötige Vererbung oder breite Gruppenrechte vermeiden.
- Zugriff regelmäßig überprüfen und nach Rollenwechseln entziehen.

GRUNDSATZ
Minimale Berechtigungen vergeben: Wer Geräte verwalten darf, benötigt nicht automatisch Zugriff auf deren Administratorkennwörter.

7. EREIGNISPROTOKOLLIERUNG



Protokolle helfen nachzuvollziehen, ob Richtlinien angewendet oder Kennwortinformationen abgerufen wurden.

- Windows-LAPS-Ereignisse auf den verwalteten Geräten prüfen, etwa zur Richtlinienveraebeitung und zu Rotationsvorgängen.
- Für Active-Directory-Kennwortzugriffe kann die Verzeichnisdienstüberwachung relevante Ereignisse liefern, wenn die passende Überwachung eingerichtet ist.
- Protokollierung erstezt keine Zugriffsbeschränkung: Aufbewahrung und Alarmierung passend zum Schutzbedarf planen.

8. DSRM-KENNWORTVERWALTUNG



Der Directory Services Restore Mode (DSRM) ist ein Wartungs- und Wiederherstellungsmodus für Active-Directory-Domänencontroller. Windows LAPS kann in unterstützten Umgebungen auch das DSRM-Kennwort verwalten.

- Die DSRM-Verwaltung muss ausdrücklich konfiguriert und auf geeigneten Domänencontrollern eingesetzt werden.
- Kennwortsicherung und Abrufzugriff ebenso streng beschränken wie bei anderen privilegierten Konten.
- Wiederherstellungsverfahren regelmäßig in einer kontrollierten Umgebung testen.

MERKE
DSRM-Zugangsdaten sind besonders privilegiert. Sie gehören in einen eng kontrollierten Wiederherstellungsprozess.

LINUX-PAM UND NSS-SICHERHEIT

PAM steuert Prüfungen - NSS liefert Systemdaten

1. PAM-STACK-AUFBAU



PAM-Regeln sind in vier Managementgruppen gegliedert. Ein Dienst kann je Gruppe mehrere Module in definierter Reihenfolge aufrufen.

- auth: Identität prüfen und gegebenenfalls Zugangsdaten setzen
- account: Kontozugriff und Rahmenbedingungen prüfen
- password: Authentifizierungsdaten ändern
- session: Aufgaben vor und nach einer Sitzung ausführen

2. KONTROLL-FLAGS



Kontroll-Flags bestimmen, wie ein Moduleregebnis den weiteren Ablauf beeinflusst. Die konkrete Wirkung hängt von Flag und Position im Stack ab.

- **required**: Fehler wird vorgemerkt; weitere Module laufen weiter
- **requisite**: Fehler beendet die Verabreitung unmittellar
- **sufficient**: Erfolg kann ausreichen, sofern kein früherer **required**-Fehler vorliegt
- **optional**: Ergebnis ist meist nur relevant, wenn kein anderes Modul die Gruppe bestimmt



SICHERHEITSASPEKT

Reihenfolge und Flag-Kombinationen können den Zugang unerwartet erlauben oder sperren. Änderungen nur mit getesteter Wiederherstellungsmöglichkeit vornehmen.

3. NSS-KONFIGURATION UND RISIKEN



NSS vermittelt Anfragen nach Systeminformationen an konfigurierte Quellen. Die Datei **/etc/nsswitch.conf** legt unter anderem Reihenfolge und Bedingungen für Quellen fest.

- Einträge wie passwd, group und hosts benennen Datenklassen
- Quellen können lokal oder netzwerkbasiert sein
- Falsche Reihenfolge oder nicht verfügbare Quellen können Anmeldung und Namensauflösung beeinträchtigen
- Unnötige oder unsichere Quellen erhöhen das Risiko manipulierter Antworten

4. /ETC/PAM.D/ ABSICHERN



PAM-Dienstdateien legen fest, welche Module eine Anwendung verwendet. Fehlerhafte Regeln können Konten aussperren oder Schutzprüfungen umgehen.

- Zugriff auf Konfigurationsdateien auf Administratoren begrenzen
- Besitz, Rechte und Änderungen regelmäßig kontrollieren
- Vor Änderungen Sicherung und getesteten Wiederherstellungsweg bereithalten
- Änderungen an einem nicht kritischen Dienst testen; aktive Administratorsitzung erhalten

5. MFA-MODULE



Mehr-Faktor-Authentifizierung ergänzt ein Passwort um einen weiteren Nachweis. PAM kann passende Module in den Anmeldeablauf integrieren.

- pam_google_authenticator kann zeitbasierte Einmalcodes prüfen
- pam_u2f kann U2F-kompatible Sicherheitsschlüssel einbinden
- Modulunterstützung und Konfiguration unterscheiden sich je System
- Wiederherstellung und sichere Registrierung der Faktoren mitplanen

WICHTIG

MFA schützt nicht automatisch jeden Zugangspfad. Prüfen, welche Dienste und privilegierten Zugänge tatsächlich durch die Regel erfasst werden.

6. PAM_FAILLOCK



pam_faillock kann fehlgeschlagene Authentifizierungsversuche erfassen und eine zeitweise Kontosperre umsetzen. Es begrenzt bestimmte Brute-Force-Versuche, erstezt aber keine weiteren Schutzmaßnahmen.

- Sperrschwelle und Dauer passend zum Betriebsrisiko festlegen
- Ädministrative und Notfallzugänge berücksichtigen
- Protokollierung und Entsperrverfahren dokumentieren
- Sperrverhalten testen, damit keine unbeabsichtigte Dauersperre entsteht

7. PAM-LOGGING UND AUDITD



Protokolle helfen, Anmeldeereignisse und Konfigurationsänderungen nachzuvollziehen. Auditd kann konfigurierte Systemereignisse erfassen; Umfang und Detailgrad hängen von den Regeln des Systems ab.

- Relevante PAM-Fehler und erfolgreiche Zugriffe auswerten
- Änderungen an Authentifizierungsdateien überwachen
- Zeitstempel und Systemzeit konsistent halten
- Zugriff auf Protokolle begrenzen und Aufbewahrung festlegen

DATENSCHUTZ

Protokolle können sensible Informationen enthalten. Keine Passwörter oder Einmalcodes protokollieren; Zugriff und Aufbewahrung begrenzen.

8. SUDOERS ABSICHERN



Rudoers steuert, welche Nutzer Befehle mit erweiterten Rechten ausführen dürfen. Eine fehlerhafte Regel kann weitreichende Privilegien vergeben.

- Regeln nach dem Prinzip der geringsten Rechte formulieren
- Syntax mit visudo prüfen und Änderungen kontrolliert ausrollen
- Keine unnötig weit gefassten Befehlsfreigaben vergeben
- Gruppenmitgleidschaften und Ausnahmen regelmäßig überprüfen

ZENTRALE ARCHITEKTUR

Linux-Authentifizierung verbindet Anwendungen mit modularen Prüfungen und Systemdaten. PAM steuert Authentifizierung und Sitzungsabläufe; NSS bestimmt, aus welchen Quellen Nutzer-, Gruppen- und weitere Namensdaten gelesen werden.



- PAM: Regeln und Module für Anmeldedienste
- NSS: Reihenfolge und Quellen für Systemdaten
- Konfigurationen sorgfältig prüfen und Änderungen testen

LINUX-KERNEL-MAC

Zusätzliche Zugriffskontrolle durch Sicherheitsrichtlinien; SELinux und AppArmor nutzen LSM.

- MAC beschränkt Prozesse zusätzlich zu Benutzer- und Gruppenrechten.
- Die konkrete Wirkung hängt von aktiver Richtlinie, Modus und Systemkonfiguration ab.



1. DAC ODER MAC?

Bei DAC (Discretionary Access Control) steuern Eigentümer und klassische Dateirechte den Zugriff. MAC setzt zusätzliche Regeln, die durch eine Sicherheitsrichtlinie festgelegt werden.

- DAC: etwa Lesen, Schreiben und Ausführen für Eigentümer, Gruppe und andere.
- MAC: begrenzt, was ein Prozess trotz ausreichender DAC-Rechte tun darf.
- MAC ersetzt DAC nicht; beide Kontrollebenen wirken zusammen.



2. SELINUX: ARCHITEKTUR

SELinux ordnet Prozessen und Objekten Sicherheitskontexte zu. Die geladene Policy bestimmt, welche Zugriffe erlaubt sind.

- Type Enforcement (TE): Regeln erlauben oder verweigern Zugriffe zwischen Typen und Domänen.
- MLS (Multi-Level Security): kann Vertraulichkeitsstufen und Zugriffsregeln dafür abbilden.
- TE und MLS sind unterscheidliche Policy-Konzepte; eine Policy muss nicht beide verwenden.



3. SELINUX: BOOLEANS UND POLICY

Booleans schalten in einer SELinux-Policy vorgesehene Verhaltensoptionen um. Sie ersetzen keine beliebigen neuen Regeln.

- Status abfragen: getsebool -a
- Eine vorhandene Option temporär setzen: setsebool OPTION on
- Dauerhaft setzen: setsebool -P OPTION on
- Policy-Quellen werden je nach Distribution und Werkzeugkette verwaltet und kompiliert; Änderungen kontrolliert testen.



Sicher arbeiten

Ersetze OPTION durch einen auf dem System vorhandenen Boolean. Prüfe vor einer Änderung dessen Bedeutung und Auswirkungen; Distributionen unterscheilden sich bei Verwaltungswerkzeugen.

4. APPARMOR: PROFILE

AppArmor bindet Regeln typischerweise an Programmpfade. Profile legen fest, auf welche Ressourcen ein Programm zugreifen darf.

- Profile können geladen, überprüft und gezielt angepasst werden.
- Pfadbasierte Regeln unterscheiden sich im Modell von SELinux-Kontexten und Typen.
- Profilverwaltung und Werkzeugnamen können je nach Distribution variieren.



5. DURCHSETZUNGSMODI

Ein Durchsetzungsmodus bestimmt, ob Regelverletzungen blockiert oder zunächst nur protokolliert werden.

- SELinux Enforcing: Policy wird durchgesetzt; verweigerte Zugriffe können protokolliert werden.
- SELinux Permissive: Zugriffe werden nicht durch SELinux blockiert, Verstöße aber protokolliert.
- AppArmor Complain: Profilverstöße werden protokolliert, statt durch das Profil blockiert, zu werden.
- Permissive oder Complain ist keine gleichwertige Schutzwirkung wie aktive Durchsetzung.

Achtung

Ein nicht durchsetzender Modus kann den Schutz reduzieren. Verwende ihn nicht als dauerhafte Fehlerbebehung.

6. KONTEXTE PRÜFEN

Kontextinformationen helfen zu verstehen, welcher Prozess und welche Datei von einer Regel betroffen sind.

- ls -Z DATEI zeigt SELinux-Sicherheitskontexte von Dateien; ls -Z zeigt sie für Einträge im aktuellen Verzeichsis.
- ps -Z zeigt SELinux-Kontexte laufender Prozesse.
- Ein Kontext ist Diagnoseinformation, nicht automatisch ein Beweis für die Ursache eines Fehlers.
- Die Befehle setzen SELinux-Werkzeuge und passende Systemunterstützung voraus.



7. VERWEIGERUNGEN ANALYSIEREN

Beginne bei einer Verweigerung mit dem Ereignisprotokoll. und prüfe, ob die Regelverletzung erwartet oder ein Hinweis auf eine Fehlkonfiguration ist.

- SELinux-Auditmeldungen und Systemprotokolle nach Zeitpunkt und betroffenen Prozess durchsuchen.
- audit2allow kann aus SELinux-Auditmeldungen Vorschläge für Policy-Regeln erzeugen.
- Vorschläge nicht blind übernehmen: Sie können unnötig weitreichende Berechtigungen erteilen.
- Zuerst Ursache, Dateikontext und vorhandene Policy prüfen; Änderungen eng begrenzen und testen.

Wichtig

audit2allow ist ein Hilfswerkzeug, keine automatische Sicherheitsfriegabe. Eine erzeugte Regel muss fachlich geprüft und auf das notwendige Minimum beschränkt werden.

8. CONTAINER UND LSM

Container teilen sich den Kernel des Hosts. MAC-Regeln können deshalb eine zusätzliche Begrenzung für Prozesse und Ressourcen eines Containers bieten.

- LSM-Schutz ergänzt andere Kontrollen wie Namespaces, Fähjokeiten, DAC-Rechte und Ressourcenlimits.
- Host-Policy und Laufzeitkonfiguration beeinflussen, welche MAC-Regeln für Container gelten.
- Prüfe die wirksame Konfiguration an Host und Container-Runtime; nicht jede Umgebung nutzt dieselbe Integration.
- Änderungen in einer Testumgebung prüfen, bevor sie produktive Container betreffen.



126 POWER SHELL-SICHERHEIT

Skripting-Umgebungen unter Windows absichern und überwachen. Die Maßnahmen ergänzen sich: Richtlinien begrenzen bestimmte Ausführungen, Protokolle schaffen Sichtbarkeit, und kontrollierte Administration reduziert Riisken.

- Ziel: Ausführung kontrollieren, Aktivitäten nachvollziehen und privilegierte Aufgaben begrenzen.



1. EXECUTION POLICIES UND GRENZEN



- Execution Policies steuern, unter welchen Bedingungen PowerShell Skriptdateien lädt.
- Sie sind eine Schutzmaßnahme gegen unbeabsichtigte Ausführung, aber keine Sicherheitsgrenze gegen einen entschlossenen Angreifer.
- Gängige Modi: Restricted, AllSigned, RemoteSigned und Unrestricted.
- RemoteSigned verlangt Signaturen für Skripte aus dem Internet; lokale Skripte können anders behandelt werden.
- Die wirksame Richtlinie hängt von Geltungsbereich und Priorität ab; prüfen mit Get-ExecutionPolicy -List.
- Nicht als Ersatz für Zugriffskontrollen, App-Kontrolle oder Überwachung verwenden.

2. CONSTRAINED LANGUAGE MODE (CLM)



- CLM schränkt bestimmte PowerShell-Sprachfunktionen und .NET-Zugriffe ein, um die Möglichkeiten nicht vertrauenswürdiger Skripte zu begrenzen.
- Die Wirkung ist besonders relevant, wenn ein unterstütztes Anwendungssteuerungs-System die Vertrauensentscheidung vorgibt.
- AppLocker oder Windowsl Defender Application Control (WDAC) können PowerShells Sprachmodus beeinflussen.
- CLM ist keine vollständige Isolatien: ergänzende Kontrollen und Überwachung bleiben wichtig.
- Für eine wirksame Anwendung Richtlinien im vorgesehenen Enforcement-Modus testen und überwachen.
- Nicht mit einer bloßen Execution Policy gleichsetzen.

3. SCRIPT BLOCK LOGGING: EREIGNIS 4104



- Script Block Logging zeichnnt Inhalte von verarbeiteten PowerShell-Skriptblöcken im Windows-Ereignisprotokoll auf.
- Das Security-Monitoring kann damit Skriptaktivitäten untersuchen; Protokollierung ist jedoch keine automatische Abwehr.
- Die Ereignisse srcchienen im Protokoll Microsoft-Windows-PowerShell/Operational; Ereignis-ID 4104 kennzeichnert Script-Block-Inhalte.
- Über Gruppenrichtlinien oder geeignete Verwaltungsrichtlinien aktivieren.
- Zentrale Weiterleitung und Zugriffsschutz einplanen; Protokolle können sensible Inhalte enthalten.
- Auf Lücken, ungewöhliiche Skriptblöcke und verdächtige Ausführungsmuster achten.

4. TRANSCRIPTION LOGGING UND ZENTRALE ABLAGE



- Transcription kann eine lesbare Aufzeichnung von PowerShell-Eingaben und -Ausgaben in Sitzungen erstellen.
- Eine zentrale, geschützte Ablage erleichtert Auswertung und Beweissicherung, sofern Aufbewahrung und Zugriff angemessen geregelt sind.
- Transkripte sind nicht dasselbe wie Script-Block-Ereignisse: beide liefern unterssicdlelche Perspektiven.
- Ablage mit restriktiven Berechtigungen und manipulationserschwerender Übertragung gestalten.
- Auf vertrauliche Daten in Ein- und Ausgaben achten; Datenschutz und Aufbewahrung berücksichtigen.
- Fehlende Transkripte nicht als Beweis fehlender Aktivitäten deuten.

5. MODULE LOGGING



- Module Logging protokolliert Pipeline-Ausführungen ausgewähiter PowerShell-Module.
- Es kann ergänzende Details für die Analyse liefern, erzeugt aber je nach Umfang erhebliche Datenmengen.
- Module gezielt auswäklen und Protokollvolumen sowie Nutzen abwägen.
- Ereignisse zentral sammeln und gemeinsam mit Script-Block- und Transkript-Daten auswerten.
- Protokolle auf sensible Informatingen und geeignete Zugriffsbeschränkungen prüfen.
- Logging-Status regelmäßig kontrollieren: Aktivierung allein garantiert keine vollständige Sichtbarkeit.

6. JUST ENOUGH ADMINISTRATION (JEA)



- JEA ermöglicht delegierte PowerShell-Administration mit eingeschränkten Rollen.
- Statt umfassender Rechte erhalten Administrierende nur die vorgesehenen Befehle und Aufgaben.
- Rollen nach Aufgaben und mininal notwendigen Berechtigungen definieren.
- Rollenfunktionen und Sitzungs-Konfigurationen kontrolliert testen und versionieren.
- Identitäten, Sitzungszugriffe und ausgeführte Befehle nachvollziehbar protokollieren.
- Regelmäßig prüfen, ob Rollen noch erforderlich und eng genug begrenzt sind.

7. ANTIMALWARE SCAN INTERFACE (AMSI)



- AMSI bietet Anwendungen eine Schnittstelle, über die Inhalte zur Prüfung an installierte Antimalware-Produkte übergeben werden können.
- PowerShell unterstützt diese Integration; Erkennung hängt unter andierem von Produkt, Konfiguration und Inhalt ab.
- AMSI ist eine Schnittstelle, kein eigenständiger Virenscanner.
- Sicherheitsprodukt und Signaturen aktuell halten; Schutzstatus überwachen.
- Verdächtige Erkennungen mit Prozess-, Skript- und Endpoint-Telemetrie korrelieren.
- Keine einzelse Kontrolle eis lückenlos ansehen. Keine einstiree Sonlcorgeita oomentee ohotinen.

8. AMSI-BYPASS-ERKENNUNG UND REAKTION



- Versuche, Prüfungen zu umgehen oder Überwachung zu beeinträchtigen, sind mögliche Hinweise auf einen Angriff.
- Erkennung solite auf mehreren Telemetriequellen beruhen und den Kontext des Systems berücksichtigen.
- Warnsignale: unerwartete Ändarungen an Sicherheits-komponenten, verdächtige PowerShell-Prozesse oder abrupte Protokolllücken.
- Script-Block- und Modulereignisse, Transkripte sowie Endpoint- und Sicherheitsproduktmeldungen zusammen betrachten.
- Abweichungen untersuchen, betroffene Systeme nach Verfahren isolieren und Belege sichern.
- Erkennungsregeln in einer autorisierten Testumgebung prüfen; keine Umgehung als Schutzmaßnahme behandeln.

EINORDNUNG



Diese Mindmap vermittelt Sicherheitsgrundlagen. Sie ist keine Anleitung zur Diagnose eines Sicherheitsvorfalls und ersetzt nicht die organisationsinternen Reaktionsverfahren.

127 · APPLICATION CONTROL

APPLICATION CONTROL

Windows-Richtlinien begrenzen, welche Anwendungen und Skripte ausgeführt werden dürfen. Ziel ist, nicht genehmigte oder nicht vertrauenswürdige Software zu blockieren und genehmigte Software kontrolliert zuzulassen.

- Regeln sorgfältig planen, testen und überwachen.
- Eine Freigabe ist keine Garantie, dass Software frei von Schwachstellen ist.

EINORDNUNG

Application Control verringert die Angriffsfläche, ersetzt aber weder Updates noch weitere Schutzmaßnahmen.

1. WDAC UND APPLOCKER



WDAC heißt heute meist App Control for Business; ältere Dokumentation nennt es Windows Defender Application Control. AppLocker ist ein separates Windows-Regelwerk.

- WDAC setzt Codeintegritätsrichtlinien durch und kann je nach Richtlinie Kernel- und User-Mode-Code kontrollieren.
- AppLocker bietet Regeln für bestimmte Benutzer oder Gruppen und Anwendungsarten; verfügbare Funktionen hängen von Windows-Version und Edition ab.
- Die Ansätze unterscheiden sich in Architektur, Verwaltung und Durchsetzung. Die passende Wahl richtet sich nach Umgebung und Anforderungen.

2. SIGNATURBASIERTE VERTRAUENSKETTEN



Signaturregeln können Herausgeber, Zertifikat oder signierte Datei zur Bewertung heranziehen. Eine digitale Signatur verbindet eine Datei mit einem Signierer und ermöglicht die Integritätsprüfung gegen nachträgliche Änderungen.

- Vertrauen in eine Signatur hängt auch davon ab, welche Herausgeber die Richtlinie zulässt.
- Ein gültiges Signaturzertifikat belegt nicht, dass ein Programm harmlos oder fehlerfrei ist.
- Widerruf, Zertifikatswechsel und Aktualisierungen in der Richtlinie berücksichtigen.

3. HASH- UND PFADREGELN



Ein Hash identifiziert einen konkreten Dateiinhalt. Ändert sich die Datei, ändert sich normalerweise auch ihr kryptografischer Hash; eine Hashfreigabe muss deshalb bei neuen Versionen aktualisiert werden.

- Hashregeln sind präzise, aber wartungsintensiv.
- Pfadregeln erlauben oder blockieren Dateien anhand ihres Speicherorts.
- Ein beschreibbarer, nicht vertrauenswürdiger Ordner ist als freigegebener Pfad riskant: Angreifer könnten dort eigene Dateien ablegen.
- Regeln nach dem Prinzip der geringsten Berechtigung gestalten.

4. MANAGED INSTALLER



Das Managed-Installer-Konzept kann Dateien als verwaltet markieren, wenn sie über einen dafür konfigurierten Installationsprozess bereitgestellt wurden. WDAC-Richtlinien können dieses Signal für Vertrauensentscheidungen nutzen.

- Nur vertrauenswürdige und kontrollierte Bereitstellungswerkzeuge entsprechend konfigurieren.
- Das Signal ersetzt keine Prüfung der Quelle oder des Installationsprozesses.
- Berechtigungen und Änderungsmöglichkeiten des Installationssystems absichern.

5. INTELLIGENT SECURITY GRAPH



Die Intelligent Security Graph (ISG)-Option kann cloudbasierte Microsoft-Reputationsinformationen in unterstützten WDAC-Szenarien zur Vertrauensbewertung heranziehen.

- Die Bewertung kann sich mit Reputationsinformationen verändern; sie ist keine dauerhafte Garantie.
- Netzwerkverfügbarkeit, Datenschutzanforderungen und Richtlinienkonfiguration beachten.
- ISG nicht mit einer lokalen Signatur- oder Hashregel gleichsetzen.

6. KERNEL-MODE UND USER-MODE



Kernel-Mode-Code läuft mit weitreichenden Systemrechten; User-Mode-Anwendungen laufen grundsätzlich mit stärker begrenzten Rechten. Fehler oder Schadcode im Kernel können daher besonders schwerwiegende Folgen haben.

- Codeintegritätsrichtlinien können Kernel-Treiber und – bei geeigneter Konfiguration – User-Mode-Anwendungen kontrollieren.
- Welche Regeln und Funktionen greifen, hängt von Plattform, Richtlinienformat und Konfiguration ab.
- Treiberkompatibilität und Startverhalten vor der breiten Einführung testen.

7. RICHTLINIEN, MERGING UND KOMPILIERUNG



WDAC-Richtlinien werden üblicherweise als XML erstellt und in ein Binärformat für die Bereitstellung umgewandelt. Mehrere Richtlinien können je nach unterstützter Windows-Version und Konfiguration gemeinnam ausgewertet werden.

- XML ist die bearbeitbare Richtliniendarstellung; die bereitgestellte Binärdatei muss zur Zielumgebung passen.
- Richtlinienmerging ist an unterstützte Formate und Einstellungen gebunden; nicht jede Kombination ist automatisch gültig.
- Änderungen versionieren, überprüfen und kontrolliert ausrollen.
- Aktualisierungs- und Wiederherstellungsweg vorab planen.

SICHER VORGEHEN

Richtlinien in einer repräsentativen Testumgebung erstellen und validieren, bevor sie produktive Geräte betreffen.

8. AUDIT-MODUS UND EREIGNISSE



Im Audit-Modus werden Regelverletzungen protokolliert, statt die betreffenden Ausführungen durch diese Richtlinie zu blockieren. Ereignisse helfen, Auswirkungen und fehlende Freigaben vor einer Durchsetzung einzuschätzen.

- Protokolle auswerten und legitime Anwendungen von unerwarteten Ausführungsversuchen unterscheiden.
- Anwendungsbestand und Arbeitsabläufe mit einbeziehen; fehlende Regeln gezielt ergänzen.
- Nach Tests schrittweise zur Durchsetzung wechseln und Ereignisse weiter beobachten.
- Audit-Modus bedeutet nicht, dass andere Sicherheitskontrollen deaktiviert sind.

MSSQL-SICHERHEIT



SQL Server schützen: Identitäten, Berechtigungen, Daten, Abfragen und Verbindungen nach dem Prinzip der Tiefenverteidigung absichern.

- Ziel: Vertraulichkeit, Integrität und Verfügbarkeit von Daten und Diensten stärken.

EINORDNUNG
Die Mindmag vermittelt Sicherheitsgrundlagen. Konfigurationen müssen zur jeweiligen SQL-Server-Version und Umgebung passen.

1. AUTHENTIFIZIERUNG

Der Anmeldemodus bestimmt, wie SQL Server die Identität einer Verbindung prüft.

- Windows-Authentifizierung: Anmeldung über Windows-Identität; zentrale Richtlinien und integrierte Anmeldung sind möglich.
- SQL-Authentifizierung: SQL-Server-Anmeldename und Kennwort; Geheimnisse sicher verwalten und starke, einizigarijte Kennwörter verwenden.
- Gemischter Modus erlaubt beide Verfahren; nur aktivieren, wenn SQL-Anmeldungen tatsächlich benötigt werden.
- Nicht benötigte Konten deaktivieren; Anmeldeversuche und Fehler überwachen.

2. GERINGSTE RECHTE

Jede Identität erhält nur die Berechtigungen, die sie für ihre Aufgabe benötigt.

- Serverrollen wirken auf Instanzebene; Datenbankrollen gelten innerhalb einer Datenbank.
- Feste Rollen können weitreichende Rechte bündeln: Mitglieidschaften regelmäßig prüfen.
- Keine alltägliche Nutzung von sysadmin; administrative Konten getrennt und kontrolliert einsetzen.
- Rollen für Aufgaben definieren und Berechtigungen gezielt vergeben statt pauschal breite Rechte zuzuweisen.
- Berechtigungen und Ausnahmen dokumentieren und wiederkehrend überprüfen.

3. TRANSPARENT DATA ENCRYPTION

TDE schützt Datenbankdateien und Sicherungen im Ruhezustand durch Verschlüsselung auf Speicherebene.

- Verschlüsselt Daten- und Protokolldateien einer Datenbank transparent für Anwendungen.
- TDE erstazt weder Transportverschlüsselung noch Zugriffssteuerung oder Anwendungsschutz.
- Zertifikat bezieheungunse Schlüsselhierarchie sicher sichern und Wiederherstellung mitplanen.
- Ohne passende Schlüssel können verschlüsselte Sicherungen nicht wie vorgesehen wiederhergestellt werden.
- Berchtzumfang und Verfügbarkeit hängen von Version, Edition und Einsatzumgebung ab.

4. ALWAYS ENCRYPTED & SPALTEN

Always Encryoted verschlüsselt ausgewählte Spalten so, dass Schlüssel und Klartext grundolstäich aufierhah des Datenbankservers bleiben können.

- Verschlüsselung und Entschlüsselung erfolgen in unterstützenden Clienttreibern.
- Spaltenverschlüsselung schützt gezielt sensible Werte; Funktionen und Abfragen können eingeschränkt sein.
- Deterministische Verschlüsselung kann Gleichheitsvergleicher ermöglichen, verrät aber Muster gleicher Werte.
- Randomisierte Verschlüsselung bietet stärkere Verbergung von Mustern, unterstäzt jedoch weniger Abfragen.
- Schlüsselverwaltung, Clientkompatibilität und Anwendungsabläufe vor Einführung testen.

5. AUDIT & DATENMASKIERUNG

Auditierung zeichent ausgewählte sicherheitsrelevante Ereignisse auf; Dynamic Data Masking verändert die Anzeige für berechtigte Abfragen, nicht die gespeicherten Daten.

- SQL Server Audit kann je nach Konfiguration Server- und Datenbankereignisse erfassen.
- Auditumfang gezielt festlegen; Speicherort, Zugriffsschutz, Aufbewahrung und Alarmierung berücksichtigen.
- Dynamic Data Masking ist keine Verschlüsselung und verhindert keinen Zugriff auf zugrunde liegende Werte durch entsprechend berechtigte Nutzer.
- Maskierung ergänzt Berechtigungen, erstazt sie aber nicht.
- Auditprotokolle regelmäßig auswerten und gegen Manipulation beziehwengunse unbefugten Zugriff schützen.

6. SCHUTZ VOR SQL-INJECTION

SQL-Injection entsteht, wenn nicht verttrauenswürdige Eingaben als SQL-Code interpretiert werden.

- Parametrisierte Abfragen trennen Datenwerte vom SQL-Befehl.
- Gespeicherte Prozeduren helfen nur, wenn sie Eingaben sicher behandeln und keine unsichere dynamische SQL-Zeichenkette zusammensetzen.
- Dynamisches SQL sorgfältig begreenen; Beaaichner lassen sich nicht wie normale Werte parametrisieren und müssen sicher validiert werden.
- Datenbankkonten der Anwendung mit minimalen Rechten ausstatten.
- Fehlermeldungen nach außen knapp halten; Sicherheitsprüfungen in autorisierten Testumgebungen durchführen.

MERKSATZ
Eine gespeicherte Prozedur verhindert SQL-Injection nicht automatisch. Entscheidend ist die sichere Konstruktion der Abfrage.

7. GEFÄHRLICHE FEATURES

Funktionen, die Betriebssystemzugriff oder zusätzilclhen ausführbaren Code ermöglichen, erhöhen die Angriffsfläche.

- xp,cmdshell kann Betriebssystembefehle ausführen; deaktivieren, sofern kein klar begründeter Bedarf besteht.
- CLR-Integration ermöglicht verwalteten Code in SQL Server; Einsatz und Berechtigungen restriktiv prüfen.
- Nicht benötigte Funktionen und Komponenten deaktivieren; Ausnahmen begrenzen und überwachen.
- Konfiguration nach Änderungen und Updates erneut prüfen.
- Administrative Rechte nicht als Enstaz für gezielte Berechtigungen verwenden.

8. NETZWERK & TLS

Netzwerkregeln begrenzen erreichbare Dienste; TLS schützt Daten während der Übertragung, wenn Verbindung und Zertifikat korrekt konfiguriert sind.

- SQL-Server-Verbindungen nur aus erforderlichen Netzen und von benötigten Systemen zulassen.
- Firewallregeln nach dem Prinzip der geringsten Erreichbarkeit gestalten; unnötige Listener vermeiden.
- TLS für Verbindungen konfigurieren und unterstützte sichere Protokolle verwenden.
- Zertifikatskette, Gültigkeit und Serveridentität prüfen; Zertifikatswarnungen nicht pauschal ignorieren.
- Netzwerkverschlüsselung schützt nicht vor kompromittierten Endpunkten oder übermäßigen Datenbankrechten.

NOSQL-SICHERHEIT

Schütze Datenbanken durch Zugriffskontrolle, sichere Konfiguration, geprüfte Eingaben und verschlüsselte Verbindungen. MongoDB und Redis unterscheiden sich in Aufbau und Sicherheitsfunktionen: Richte jede Datenbank passend zu ihrer Version und Einsatzumgebung ein.

1. AUTHENTIFIZIERUNG UND STANDARD-PORT-EXPOSITION

Authentifizierung prüft, wer eine Verbindung aufbaut. Ein bekannter Standardport ist keine Sicherheitslücke an sich; offen erreichbare Datenbankdienste erhöhen jedoch die Angriffsfläche.

- MongoDB verwendet häufig TCP-Port 27017, Redis häufig TCP-Port 6379; Konfigurationen können abweichen.
- Aktiviere Authentifizierung und verwende individuelle, starke Zugangsdaten statt gemeinsamer Konten.
- Erlaube Verbindungen nur aus benötigten Netzen und überwache fehlgeschlagene Anmeldungen.
- Portänderungen allein schützen nicht vor unbefugtem Zugriff.

2. ROLLENBASIERTE ZUGRIFFSKONTROLLE IN MONGODB

RBAC (Role-Based Access Control) ordnet Benutzern Rollen und Rollen konkrete Berechtigungen zu. So lassen sich Aufgaben trennen, ohne jedem Konto Vollzugriff zu geben.

- Lege Konten für Anwendungen, Administration und Wartung getrennt an.
- Vergib nur die Datenbank- und Aktionen-Rechte, die für die jeweilige Aufgabe nötig sind.
- Vermeide dauerhaft genutzte Administratorrechte für Anwendungen.
- Prüfe Rollen regelmäßig und entziehe nicht mehr benötigte Berechtigungen.

3. REDIS-SICHERHEITSMODELL UND PROTECTED MODE

Redis ist ein In-Memory-Datenspeicher, dessen Sicherheitsmaßnahmen von Version und Konfiguration abhängen. Der protected mode ist eine Schutzmaßnahme für bestimmte unsichere Konfigurationen, aber kein Ersatz für Netzwerk- und Zugangskontrollen.

- Binde Redis an vertrauenswürdige Schnittstellen und schränke den Zugriff zusätzlich per Firewall ein.
- Verwende Authentifizierung und, sofern unterstützt und passend konfiguriert, ACLs mit gezielten Rechten und passend konfigurierte Rollen mit gezielten Rechten.
- Verlasse dich nicht allein auf protected mode oder eine nicht öffentliche IP-Adresse.
- Prüfe die Dokumentation der eingesetzten Redis-Version und teste Änderungen kontrolliert.

4. DESTRUKTIVE BEFEHLE BEGRENZEN

Befehle wie FLUSHALL können Daten in Redis löschen; CONFIG kann Konfigurationseinstellungen ändern. Deaktiviere oder benenne gefährliche Befehle um, wenn sie nicht benötigt werden.

- Gib Anwendungen keine Verwaltungsrechte, wenn sie diese nicht benötigen.
- Beschränke gefährliche Befehle über die verfügbaren Rollen- und Konfigurationsmechanismen der eingesetzten Version.
- Teste Einschränkungen vor dem Betrieb: Anwendungen und Wartungsabläufe dürfen nicht unerwartet ausfallen.
- Sichere wichtige Daten und halte Wiederherstellungsverfahren aktuell.

Sorgfältig testen
Befehlsbeschränkungen sind versions- und konfigurationsabhängig. Prüfe sie in einer Testumgebung, bevor du sie produktiv einführst.

5. NOSQL-INJECTION ÜBER BSON UND JSON

NoSQL-Injection entsteht, wenn nicht vertrauenswürdige Eingaben die Bedeutung einer Datenbankabfrage verändern. JSON- oder BSON-artige Objekte können dabei Operatoren oder unerwartete Datentypen einschleusen.

- Behandle Eingaben nie ungeprüft als fertige Abfrageobjekte.
- Erwarte konkrete Datentypen und validiere Werte, Formate sowie zulässige Felder.
- Nutze sichere Abfrage-APIs und trenne Daten von Abfragelogik.
- Teste Grenzfälle und unerwartete verschachtelte Objekte; protokolliere Fehler ohne Geheimnisse.



6. TRANSPORTVERSCHLÜSSELUNG MIT mTLS

TLS schützt Daten während der Übertragung vor Mitlesen und Manipulation. Bei mTLS (mutual TLS) weisen sich Client und Server mit Zertifikaten gegenseitig aus.

- Aktiviere TLS für Datenbankverbindungen, wenn es die eingesetzte Plattform unterstützt.
- Bei mTLS müssen beide Seiten Zertifikate prüfen; sichere private Schlüssel sorgfältig.
- Prüfe Zertifikatskette, Gültigkeit und Servernamen statt die Prüfung abzuschalten.
- Verschlüsselung im Transport ersetzt weder Authentifizierung auf Datenbankebene noch Netzwerksegmentierung.

7. VERSCHLÜSSELUNG AUF FELDEBENE

Field-Level Encryption verschlüsselt ausgewählte Werte bereits auf Anwendungsebene. Dadurch können sensible Felder auch dann geschützt sein, wenn gespeicherte Datensätze außerhalb des vorgesehenen Zugriffs offengelegt werden.

- Wähle gezielt schützenswerte Felder; nicht jede Abfrage kann auf verschlüsselten Werten gleich funktionieren.
- Verwalte Schlüssel getrennt von den verschlüsselten Daten und beschränke den Schlüsselzugriff.
- Berücksichtige Schlüsselrotation, Backups und Wiederherstellung im Sicherheitskonzept.
- Verschlüsselung schützt nicht automatisch vor Missbrauch durch eine Anwendung, die legitimen Klartextzugriff hat.

8. NETZWERK-BINDINGS: LOCALHOST UND VPC

Ein Binding legt fest, an welche Netzwerkschnittstellen ein Dienst Verbindungen annimmt. Localhost beschränkt den Zugriff auf den lokalen Rechner; eine VPC (Virtual Private Cloud) kann Datenbankzugriff auf ein privates Cloud-Netz begrenzen.

- Binde den Dienst nur an tatsächlich benötigte Schnittstellen; vermeide eine pauschale Bindung an alle Interfaces.
- Ergänze Bindings durch Firewall- oder Sicherheitsgruppenregeln mit eng begrenzten Quellnetzen.
- Prüfe Routing, Peering und administrative Zugänge: Auch private Netze brauchen Zugriffskontrollen.
- Kontrolliere nach Änderungen, von welchen Systemen der Datenbankport erreichbar ist.

IaC-Sicherheit

Infrastructure as Code (IaC) beschreibt Infrastruktur deklarativ in Dateien. Diese Mindmap zeigt, wie du Fehlkonfigurationen und sensible Daten früh erkennst und Änderungen kontrolliert auslieferst.

- Prüfe Definitionen, Abhängigkeiten, Zustände und Änderungen gemeinsam.
- Behandle Prüfwerkzeuge als Unterstützung, nicht als Garantie für Sicherheit.

1. Terraform und CloudFormation



Deklarative Vorlagen können Ressourcen mit unsicheren Eigenschaften definieren. Entscheidend ist, was die Konfiguration tatsächlich bereitstellt.

- Prüfe öffentliche Erreichbarkeit, unnötig offene Netzwerkregeln und fehlende Verschlüsselung.
- Vergleiche Soll-Konfiguration und bereitgestellte Ressourcen; Module und Vorlagen ebenfalls prüfen.
- Bei CloudFormation: Parameter, Ressourcen und zugewiesene Berechtigungen betrachten.
- Bei Terraform: Provider, Module, Variablen und Ressourcenänderungen im Plan überprüfen.

2. Statische IaC-Analyse



Statische Analysen untersuchen Vorlagendateien, ohne dafür die Infrastruktur auszuführen. Checkov und tfsec sind Beispiele für Scanner im Terraform-Umfeld; unterstützte Formate und Regeln hängen von der jeweiligen Version ab.

- Scans in Pull Requests oder CI ausführen und Ergebnisse einer verantwortlichen Person zuordnen.
- Befunde nach Risiko und Kontext priorisieren; Fehlalarme nachvollziehbar behandeln.
- Regeln aktuell halten und eigene Ausnahmen begründen, begrenzen und regelmäßig prüfen.
- Ein sauberer Scan beweist nicht, dass Laufzeit, Identitäten oder Geschäftslogik sicher sind.



PRÜFGRENZE
Scanner finden bestimmte Muster. Sie ersetzen weder Architekturprüfung noch Tests der bereitgestellten Umgebung.

3. Drift und State-Dateien



Drift entsteht, wenn der tatsächliche Infrastrukturzustand von der deklarativen Konfiguration abweicht. State-Dateien können sensible Metadaten oder Werte enthalten und verdienen Schutz.

- Änderungen außerhalb des kontrollierten IaC-Prozesses erkennen und klären.
- State-Zugriff auf notwendige Identitäten begrenzen; Speicherung und Übertragung schützen.
- Gemeinsam genutzte Zustände gegen unkoordinierte Änderungen absichern, etwa durch Backend-Sperren, sofern verfügbar.
- State-Backups und Wiederherstellung sicher verwalten; nicht unbedacht in Versionskontrolle einchecken.



WICHTIG
Geheimnisse können auch in Plänen, Protokollen oder Ausgaben auftauchen. Sensible Artefakte entsprechend behandeln.

4. Geheimnisse in Vorlagen



Zugangsdaten, Tokens und private Schlüssel gehören nicht als Klartext in IaC-Dateien oder Versionskontrolle.

- Geheimnisse zur Laufzeit aus einem dafür vorgesehenen Secret-Store beziehen.
- Ausgaben und Logs auf versehentliche Offenlegung prüfen; Maskierung allein ist kein Schutz.
- Bei Verdacht auf Offenlegung betroffene Zugangsdaten widerrufen oder rotieren und den Vorfall untersuchen.
- Auch Variablen, Beispieldateien, Plan-Artefakte und CI-Protokolle berücksichtigen.

5. Policy as Code



Policy as Code formuliert prüfbare Regeln für zulässige Konfigurationen. Open Policy Agent (OPA) ist ein Policy-Engine-Projekt: Rego ist seine deklarative Abfragesprache.

- Regeln können beispielsweise verbieten, Speicher unbeabsichtigt öffentlich bereitzustellen.
- Policies mit positiven und negativen Beispielen testen und Änderungen versionieren.
- Festlegen, ob ein Verstoß blockiert, gewarnt oder zur manuellen Prüfung weitergeleitet wird.
- Regeln verständlich halten und Ausnahmen mit Umfang und Ablauf versehen.

6. Ansible Playbooks härten



Ansible-Playbooks automatisieren Konfigurationsänderungen. Sicherheitsrisiken entstehen unter anderem durch zu weitreichende Rechte, unsichere Variablen und unkontrollierte Ausführung.

- Nur nötige Privilegien vergeben und Zugriff auf Inventare und Geheimnisse beschränken.
- Geheimnisse über geeignete Secret-Verwaltung einbinden; sensible Ausgaben vermeiden.
- Änderungen in Testumgebungen prüfen und Auswirkungen von Rollen und Variablen nachvollziehen.
- Playbooks, Rollen und Abhängigkeiten versionieren und vor Änderungen reviewen.

7. Immutable Infrastructure



Das Prinzip unveränderlicher Infrastruktur bevorzugt den Austausch einer Instanz durch ein neu gebautes, geprüftes Abbild statt unkontrollierter Änderungen an ihr.

- Images reproduzierbar bauen, prüfen und versionieren.
- Änderungen nachvollziehbar in den Build-Prozess verlagern.
- Rollbacks und Wiederherstellung einplanen; Unveränderlichkeit verhindert nicht jede Fehlkonfiguration.
- Laufzeitdaten und dauerhafte Speicher gesondert schützen und verwalten.

8. CI/CD-Gates



CI/CD-Gates führen Sicherheitsprüfungen in den Änderungsprozess ein. Ein Gate kann eine Änderung bei festgelegten Befunden stoppen oder eine Prüfung anfordern.

- Format- und Validierungsprüfungen vor dem Merge ausführen.
- IaC-Scanner und Policies passend zum verwendeten Format einbinden.
- Geheimnisse erkennen, Berechtigungen begrenzen und sensible Artefakte schützen.
- Fehler verständlich melden; Ausnahmen nachvollziehbar dokumentieren und nachverfolgen.



ZUSAMMENFASSUNG
Sichere IaC-Praxis verbindet Vorlagenprüfung, Schutz von Geheimnissen und Zuständen, kontrollierte Änderungen und passende CI/CD-Prüfungen.

GIT-REPOSITORY-SICHERHEIT



Schütze Konten, Quellcode und Änderungen über den gesamten Entwicklungsablauf hinweg.

- Zugänge absichern
- Geheimnisse früh erkennen
- Änderungen kontrollieren
- Herkunft und Zugriffe nachvollziehbar halten

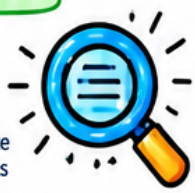
1. ENTWICKLERZUGÄNGE ABSICHERN



Kontrolliere, wer auf Repository und Organisationskonto zugreifen kann.

- MFA für Entwicklerkonten und administrative Zugänge aktivieren.
- SSH-Schlüssel nur nach Richtlinie zulassen; sichere Schlüssel geschützt aufbewahren und bei Verlust widerrufen.
- Zugriffsrechte nach Aufgabe vergeben und regelmäßig überprüfen.

2. SECRETS SCANNEN



Zugangsdaten im Quellcode können Unbefugten Zugriff auf weitere Systeme ermöglichen.

- Gitleaks oder TruffleHog können bekannte Muster und weitere Hinweise auf Sercets untersuchen.
- Scans möglichst vor dem Commit in einen Pre-Commit-Hook einbinden; zusätzlich serverseitig oder in CI prüfen.
- Ein Fund ist ein Sicherheitsvorfall: Secret widerrufen oder rotieren, Fundstelle bereinigen und Ursache untersuchen.



WICHTIG
Ein gelöschtes Secret kann in der Git-Historie erhalten beiben. Es muss als offengelegt behandelt und widerrufen werden.

3. BRANCH-SCHUTZ UND CODE REVIEWS



Schütze wichtige Branches vor ungeprüften oder unbeabsichtigten Änderungen.

- Direkte Pushes auf geschützte Branches einschränken; Änderungen über Pull- oder Merge-Requests einreichen.
- Erforderliche Reviews und erfolgreiche Statusprüfungen vor dem Zusammenführen festlegen.
- Regeln passend zum Risiko konfigurieren und Ausnahmen eng begrenzen.

4. COMMITS SIGNIEREN



Signaturen helfen, die Zuordnung eines Commits zu einem Schlüsselinhaber zu prüfen.

- GPG- oder SSH-Signierung kann die Authentizität eines Commits überprüfbar machen.
- Öffentlichen Signaturschlüssel dem Konto zuordnen und privaten Schlüssel schützen.
- Eine gültige Signatur beweist nicht, dass der Code sicher oder die Änderung sachlich korrekt ist.

5. ZUGRIFFSTOKEN VERWALTEN



Personal Access Tokens (PATs) sind Zugangsdaten und müssen wie Passwörter geschützt werden.

- Nur benötigte Berechtigungen und passende Laufzeit vergeben.
- Token nicht in Quellcode, Logs oder gemeinsam genutzten Dateien speichern.
- Nicht mehr benötigte oder möglicherweise offengelegte Tokens widerrufen und ersetzen.

6. ABHÄNGIGKEITEN PRÜFEN



Fremde Bibliotheken und Pakete erweitern die Angriffsfläche eines Projekts.

- Dependency Review macht Änderungen an Abhängigkeiten in einem Änderungsvorschlag sichtbar.
- Software Composition Analysis (SCA) vergleicht erkannte Komponenten unter anderem mit bekannten Schwachstelleninformationen.
- Ergebnisse nach Version, Nutzung und Kontext bewerten; Updates kontrolliert testen.

7. AUDIT-LOGS NUTZEN

Protokolle unterstützen die Nachvollziehbarkeit sicherheitsrelevanter Aktivitäten.

- GitHub- oder GitLab-Audit-Logs können je nach Produkt und Tarif Ereignisse zu Konten und Organisationen enthalten.
- Auf ungewöhnliche Anmeldungen, Rechteänderungen, Token-Aktivitäten und Repository-Einstellungen achten.
- Aufbewahrung, Zugriff und Alarmierung passend zu den eigenen Anforderungen festlegen.



8. REPOSITORY POISONING VORBEUGEN



Manipulierte Inhalte oder Prozesse können Entwickler dazu verleiten, unsicheren Code oder Befehle zu übernehmen.

- Änderungen aus unbekannten Quellen und unerwartete Build-Skripte sorgfältig prüfen.
- Abhängigkeiten, Actions und externe CI-Komponenten auf Herkunft und Berechtigungen kontrollieren.
- Bei verdächtigen Änderungen Zusammenführung stoppen, Verantwortliche klären und betroffene Zugangsdaten prüfen.

132 · SERVICE-MESH-SICHERHEIT

SERVICE-MESH-SICHERHEIT

Authentifizierte, kontrollierte Service-zu-Service-Kommunikation in Microservice-Clustern.

1. AUTOMATISIERTES MUTUAL TLS (mTLS)



Sidecars können TLS-Verbindungen zwischen Workloads aufbauen und dabei beide Endpunkte authentifizieren.

- Verschlüsselung schützt den Transportweg.
- Zertifikate belegen die jeweilige Identität.
- mTLS ersetzt keine Autorisierungsprüfung.

2. ENVOY UND TRAFFIC INTERCEPTION



In Istio überniment Envoy häufig die Datenebenen-Funktionen; Linkerd nutzt einen eigenen leichtgewichtigen Proxy. Der Mesh-Datenverkehr wird je nach Installation durch Proxy-Injektion und Netzwerkregeln geleitet.

- Prüfe, welche Pfade tatsächlich den Proxy passieren.
- Ausnahmen und nicht yerwaltete Workloads können Schutzlücken schaffen.
- Proxy- und Steuerungsebene getrennt betrachten.

3. SERVICE-IDENTITÄT: SPIFFE UND SPIRE



SPIFFE definiert ein Format für Workload-Identitäten; SPIRE ist eine Implementierung zur Attestierung und Ausgabe solcher Identitäten.

- Identitäten an Workloads statt nur an IP-Adressen binden.
- Kurzlebige SVIDs begrenzen die Nutzungsdauer kompromittierter Anmeldedaten.
- Mesh-Integration und konkrete Unterstützung hängen von der Plattform ab.

4. PEERAUTHENTICATION UND AUTHORIZATIONPOLICY



Istio PeerAuthentication steuert Anforderungen an die gegenseitige TLS-Authentifizierung. AuthorizationPolicy legt fest, welche identifizierten Guellen welche Aktionen ausführen dürfen.

- Authentifizierung beantwortet: Wer kommuniziert?
- Autorisierung beantwortet: Was darf diese Identität tun?
- Richtlinienumfang und Vererbungsregeln vor dem Rollout prüfen.



Sicher ausrollen
Mit beobachtbarem Ist-Zustand beginnen, Richtlinien testen und Änderungen schrittweise durchsetzen.

5. EGRESS-KONTROLLE UND API-GATEWAYS



Egress-Regeln begrenzen ausgehende Verbindungen aus dem Cluster. Ein API-Gateway kontrolliert typischerweise eingehenden oder externen API-Verkehr; so ersetzt nicht automatisch Egress-Kontrollen.

- Erlaubte Ziele und Protokolle explizit festlegen.
- DNS, direkte IP-Verbindungen und Ausweichpfade berücksishtigen.
- Gateway- und Mesh-Richtlinien auf Überschneidungen prüfen.

6. RATE LIMITING UND DISTRIBUTED TRACING



Rate Limiting begrenzt Anfragen nach festgelegten Schlüsseln und Zeitfenstern. Distributed Tracing verbindet Trace-Kontext über Service-Grenzen hinweg, sofern Instrumentierung und Weitergabe eingerichtet sind.

- Limits pessend zu Kapazität und Fehlerverhalten wählen.
- Tracing erleichtert die Analyse, ist aber keine vollständige Sicherheitsüberwachung.
- Sensible Daten nicht ungeprüft in Trace-Attribute schreiben.

7. SCHLÜSSELROTATION UND MESH-CA



Eine Mesh-Zertifizierungsstelle kann Workload-Zertifikate ausstellen. Rotation erneuert Schlüssel und Zertifikate regelmäßig oder bei Bedarf.

- Vertrauensanker und Aussteller sorgfältig schützen.
- Erneuerung, Ablauf und Widerrufsverhalten überwachen.
- Rotation sicher testen, damit gültige Verbindungen nicht unerwartet abbrechen.

8. ZERO TRUST IM POD-NETZWERK



Zero Trust bedeutet, Zugriffe nicht allein aufgrund der Netzwerklage zu vertrauen: Identität, Kontext und Berechtigung werden geprüft.

- Least Privilege für Workload-Kommunikation anwenden.
- Netzwerksegmentierung und Mesh-Richtlinien ergänzen einander.
- Nicht verwaltete Pods, Admin-Zugänge und Steuerungsebene mit einbeziehen.

BROWSER POLICY ENFORCEMENT

Browser setzen Richtlinien für den Zugriff zwischen Ursprüngen und die Einbettung aktiver Inhalte durch. SOP begrenzt standardmäßig den Zugriff; CORS kann ausgewählte Lesezugriffe erlauben. CSP beschränkt zulässige Ressourcen. COOP und COEP steuern zusätzliche Isolationsbedingungen.

1 · SAME-ORIGIN POLICY (SOP)



Die Same-Origin Policy begrenzt, wie ein Dokument oder Skript Daten anderer Ursprünge lesen kann.

- Ein Ursprung besteht aus Schema, Hostname und Port.
- https://a.example und http://a.example sind verschiedene Ursprünge.
- Auch unterschiedliche Ports oder Hostnamen ergeben verschiedene Ursprünge.
- SOP schützt nicht vor jeder Form ursprungsübergreifender Kommunikation.

2 · CORS-HEADER



CORS ist ein Browsermechanismus, mit dem ein Server bestimmte ursprungsübergreifende Lesezugriffe freigeben kann.

- Access-Control-Allow-Origin nennt einen erlaubten Ursprung oder – in passanden Fällen – *.
- Der Browser prüft die Freigabe anhand der Antwortheader.
- Für credentialierte Anfragen muss der Server Access-Control-Allow-Credentials: true senden.
- Mit Credentials ist * als Allow-Origin nicht zulässig; der konkrete Ursprung muss angegeben werden.
- Der Browser verhindert nicht automatisch, dass eine Anfrage den Server erreicht.

3 · CORS-FEHLKONFIGURATIONEN



Zu großzügige oder falsch erzeugte Freigaben können vertrauliche Antworten für fremde Webseiten lesbar machen.

- Origin-Werte nicht ungeprüft reflektieren oder anhand unsicherer Teilstrings erlauben.
- Der Ursprung null kann in bestimmten sandboxed oder lokalen Kontexten auftreten; ihn pauschal freizugeben ist riskant.
- Wildcard-Allow-Origin und credentialierte Freigabe sind keine gültige Kombination für eine effolgreiche CORS-Freigabe.
- Nur benötigte Ursprünge und Methoden erlauben; Berechtigungen regelmäßig prüfen.

4 · CSP-DIREKTIVEN



Content Security Policy (CSP) legt fest, aus welchen Quellen bestimmte Inhalte geladen oder ausgeführt werden dürfen.

- default-src dient als Standard für viele Ressourcentypen, sofern keine spezifischere Direktive greift.
- script-src steuert zulässige Skriptquellen und Skriptausführung.
- Weitere Direktiven wie img-src oder style-src können Ressourcenarten gezielt begrenzen.
- Eine CSP wird typischerweise per Content-Security-Policy-Antwortheader ausgeliefert.
- Eine restriktive CSP erschwert bestimmte XSS-Auswirkungen, verhindert aber nicht jede Schwachstelle.

5 · CSP-NONCES UND HASHES



Nonces und Hashes erlauben ausgewählte Skripte, ohne pauschal beliebige Inline-Skripte freizugeben.

- Eine Nonce ist ein schwer erratbarer Wert, der pro Antwort neu erzeugt und im CSP-Header sowie am erlaubten Element verwendet wird.
- Ein Hash erlaubt Skriptinhalt, dessen kryptografischer Hash im CSP-Header aufgeführt ist.
- Nonces dürfen nicht statisch wiederverwendet oder für nicht vertrauenswürdige Inhalte freigegeben werden.
- Nonce- und Hash-Regeln müssen exakt zum ausgelieferten Skriptinhalt passen.

6 · CSP-REPORTING



Berichtsfunktionen helfen, CSP-Verstöße zu beobachten und Richtlinien schrittweise zu verbessern.

- report-uri ist eine ältere Direktive; Unterstützung und Verhalten können je nach Browser variieren.
- Reporting API und report-to ermöglichen moderne Berichtswege, abhängig von Browserunterstützung und Konfiguration.
- Berichte können sensible URL- oder Seitendaten enthalten: Endpunkte und Aufbewahrung sorgfältig gestalten.
- Berichte dienen der Diagnose; sie blockieren oder heheben eine Ursache nicht selbst.

7 · CROSS-ORIGIN OPENER POLICY (COOP)



COOP beeinffusst, ob ein Dokument mit geöffneten oder öffnenden Dokumenten anderer Ursprünge dieselbe Browsing-Context-Gruppe teilt.

- Der Header Cross-Origin-Opener-Policy legt die gewünschte Isolation fest.
- Eine passende Einstellung kann bestimmte Referencen zwischen Fenstern trennen.
- Die konkrete Wirkung hängt von Richtlinie und Browsersituation ab.
- COOP ist ein Baustein für Isolation, kein Ersatz für sichere Anwendungslogik.

8 · CROSS-ORIGIN EMBEDDER POLICY (COEP)



COEP steuert, unter welchen Bedingungen ein Dokument Ressourcen anderer Ursprünge einbettet.

- Der Header Cross-Origin-Embedder-Policy legt fest, wie eingebettete Ressourcen behandelt werden.
- Für cross-origin-isolated-Kontexte müssen COEP und COOP zusammen mit den übrigen Browserbedingungen passend konfiguirert sein.
- Eingebettete Ressourcen benötigen je nach Fall eine geeingete CORS- oder CORP-Freigabe.
- Vor Aktivierung alle Drittanbieter-Ressourcen und Browseranforderungen testen.

Merksatz

CORS ist keine allgemeine Zugriffssperre für Serveranfragen: Es steuert vor allem, ob Browser-Skripte Antworten ursprungsübergreifend lesen dürfen.

HTTP-PROTOKOLL-ANOMALIEN

Wenn Webserver, Reverse Proxy und Cache dieselbe HTTP-Anfrage unterschiedlich interpretieren, können Anfragen fehlgeleitet oder Cache-Inhalte manipuliert werden. Die Mindmap ordnet die wichtigsten Ursachen, Angriffsfolgen und Abwehrprinzipien ein.

1. HTTP REQUEST SMUGGLING

Ein Frontend und ein nachgelagerter Server grenzen Anfragen im selben Datenstrom unterschiedlich ab. Dadurch kann ein Teil der Anfrage als zusätzliche, unbeabsichtigte Anfrage interpretiert werden.

- CL.TE: Frontend nutzt Content-Length, Backend Transfer-Encoding
- TE.CL: Frontend nutzt Transfer-Encoding, Backend Content-Length
- TE.TE: unterschiedliche Behandlung mehrdeutiger Transfer-Encoding-Angaben
- Folgen: Anfragen können sich verschieben oder dem falschen Kontext zgeordnet werden.



2. HTTP/2-DOWNGRADE

Bei einer Übersetzung von HTTP/2 zu HTTP/1.1 kann eine zusätzliche Interpretationsgrenze entstehen. Sicherheitsprobleme hängen von der konkreten Implementierung und der Weitergabe von Headern ab.

- HTTP/2 verwendet ein anderes Nachrichtenformat als HTTP/1.1
- Ein Gateway muss Nachrichten korrekt und eindeutig umsetzen
- Unstimmige Übersetzung oder Header-Normalisierung erhöht das Risiko
- Protokollübergänge und beteiligte Parser gezielt prüßen.



3. CACHE KEYS UND NICHT BERÜCKSICHTIGTE EINGABEN

Ein Cache Key bestimmt, unter welchem Schlüssel eine Antwort gespeichert und wiedergefunden wird. Eingaben, die die Antwort beeinflussen, aber nicht in den Schlüssel einfließen, können zu einer falschen Wiederverwendung fühnen.

- Typische Bestandteile: Host, Pfad und gegebenenfalls ausgewählte Parameter oder Header
- Unkeyed Input: beeinflusst die Antwort, wird aber nicht im Cache Key abgebildet
- Variierende Antworten müssen zur Cache-Konfiguration passen
- Personalisierte oder sensible Antworten nicht unbedacht gemeinsam cachen.



4. WEB CACHE POISONING

Beim Cache Poisoning wird eine Antwort, die der Cache für weitere Abrufe speichert, durch eine nicht vorgesehene Eingabe oder fehlerhafte Vertrauensgrezne beeinflusst.

- Voraussetzung: eine Eingabe beeinflusst die Antwort, ohne angemessen berücksichtigt zu werden
- Ein Cache speichert die veränderte Antwort unter einem gemeinsam genutzten Schlüssel
- Mögliche Folge: andere Nutzer erhalten den unerwünschten Inhalt
- Abwehr: Eingaben normalisieren, Schlüssel und Cache-Regeln gemeinsam gestalten.



5. CACHE DECEPTION

Cache Deception bezeichnet Konstellationen, in denen eine dynamische oder vertrauliche Antwort irrtümlich als cachefähiger Inhalt behandelt wird. Entscheidend sind Routing, Cache-Regeln und die tatsächliche Antwort.

- Nicht allein auf Dateiendungen oder Pfadmuster vertruuen
- Cache-Verhalten für dynamische und private Inhalte ausdrücklich festlegen
- Cache-Control und weitere relevante Antwortmerkmale prüfen
- Mit getrennten Testkonten und nicht sensiblen Beispieldaten testen.



6. CACHE-ÜBERLASTUNG UND VERFÜGBARKEIT

Viele unterschiedliche oder aufwendige Anfragen können Cache und Ursprungsserver belasten. Ein Cache verhindert nicht automatisch Überlastung; bei vielen Cäihren, dass Schichten eine zusätzliche Arbeit am Backend entstehen.

- Hohe Vielfalt an Cache Keys kann die Trefferquote senken
- Aufwendige dynamische Antworten können Backend-Ressourcen beanspruchen
- Beobachten: Anfragevolumen, Cache-Treffer, Latenz und Backend-Auslastung
- Begrenzen und sklarbeiren; Schutzmaßnahmen auf mögliche Fehlalarme prüfen.



7. REVERSE-PROXY-ROUTING

Reverse Proxies nehmen Client-Anfragen entgegen und leiten sie an interne Dienste weiter. Unterschiede bei Host, Pfad, Normalisierung oder Header-Behandlung können dazu führen, dass Schichten eine Anfrage verschieden zuordnen.

- Routingregeln an Proxy, Gateway und Anwendung aufeinander abstimmen
- Mehrdeutige Pfade und widersprüchliche Host-Angaben zurückweisen
- Vertrauenswürdige und vom Client kontrollierte Header klar unterschieden
- Logs über die Verarbeitungsschichten hinweg korrelieren.



8. GEGENMASSNAHMEN

Ziel ist, mehrdeutige Anfragen früh zu verwerfen und für alle beteiligten Systeme eine einheitliche Interpretation sicherzustellen.

- Strikte, konsistente HTTP-Parser verwenden und fehlerhafte Nachrichten ablehnen
- HTTP-Spezifikationen konform implementieren; Protokoll-Downgrades kontrollieren
- Content-Length und Transfer-Encoding eindeutig behandeln; Konflikte zurückweisen
- Cache Keys, private Antworten und Cache-Control-Regeln gezielt konfigurieren
- Proxy- und Anwendungskonfiguration gemeinsam testen und aktuell halten.



SICHERHEITSRAHMEN

Die Beispiele beschreiben Risiken und Abwehrmaßnahmen. Tests an realen Systemen nur mit ausdrücklicher Berechtigung durchführen.

GRAPHQL-SICHERHEIT

Zugriff, Anfragekosten und Backend-Ausführung absichern.

1. SCHEMA-INTROSPEKTION-RISIKEN



Introspection ermöglicht Clients, Schema-Informationen abzufragen. Das kann Entwicklung und Werkzeuge unterstützen, aber auch Typen und verfügbare Felder offenlegen.

- Bewerte die Sichtbarkeit nach Umgebung und Bedrohungsmodell; Introspection allein ist nicht automatisch eine Sicherheitslücke.
- Begrenze sensible Daten durch Schema- und Zugriffskontrollen, nicht nur durch das Verbergen von Schema-Details.
- Verlasse dich nicht darauf, dass Angreifer unbekannte Felder nicht erraten können.

2. QUERY-TIEFE UND KOMPLEXITÄT



Verschachtelte Abfragen und teure Felder können hohe Rechenlast auslösen. Eine reine Tiefengrenze erkennt nicht alle kostspieligen Abfragen.

- Setze eine maximale Query-Tiefe passend zum Schema.
- Bewerte zusätzlich die erwartete Feld- oder Resolver-Kosten; begrenze die Gesamtkomplexität.
- Berücksichtige Listenargumente und mögliche Ergebnismengen bei der Kostenschätzung.
- Begrenze Laufzeit und Ressourcen serverseitig; teste legitime Abfragen gegen die Limits.

3. DENIAL OF SERVICE DURCH BATCH-REQUESTS



Batching kann mehrere Operationen in einer Anfrage bündeln. Ohne Grenzen kann das die Arbeit des Servers stark erhöhen oder Schutzmechanismen umgehen, die nur einzelne HTTP-Anfragen zählen.

- Begrenze Anzahl und Gesamtaufwand gebündelter Operationen.
- Wende Limits auf die tatsächliche Arbeit an, nicht nur auf die Zahl der HTTP-Anfragen.
- Kombiniere Kontrollen für Batch-Größe, Query-Kosten, Zeit und Ressourcen.
- Beobachte Fehlerraten und Lastspitzen, ohne vertrauliche Inhalte unnötig zu protokollieren.

4. BOLA / IDOR IN RESOLVERN



Broken Object Level Authorization entsteht, wenn ein Nutzer auf ein Objekt zugreifen kann, für das ihm die Berechtigung fehlt. Ein Resolver darf nicht allein deshalb Daten liefern, weil eine ID bekannt oder die Anfrage authentifiziert ist.

- Prüfe bei jedem Objektzugriff die Berechtigung für genau dieses Objekt und die angeforderte Aktion.
- Leite den Nutzerkontext aus einer vertrauenswürdigen Authentifizierung ab, nicht aus frei übermittelten IDs oder Rollen.
- Nutze zentrale Autorisierungsfunktionen und teste Zugriffe über altennative Abfragepfade.
- Vermeide Antworten, die unbefugten Zugriff durch unterschiedliche Fehlermeldungen unnötig bestätigen.

5. RATE LIMITING AUF RESOLVER-EBENE



Ein Limit nur auf HTTP-Ebene kann zu grob sein: Eine einzelee GraphQL-Anfrage kann zahlreiche Resolver-Aufrufe auslösen. Zusätzliche Kontrollen können die tatsächliche Arbeit besser abbilden.

- Kombiniere gegebenenfalls Limits pro Identität, Client oder IP mit Kosten- und Resolver-Kontrollen.
- Berücksichtige Missbrauch durch viele IDs, verschachtelte Felder und Batching.
- Definiere klare Antworten bei Überschreitung und überwache legitime Fehlblockierungen.
- Verwende keine IP-Adresse als einziges Identitätsmerkmal; geteilte Netze und Proxys können viele Nutzer bündeln.

6. FIELD SUGGESTIONS UND INFORMATIONSABFLUSS



Manche GraphQL-Implementierungen schlagen bei unbekannten Feldern ähnliche gültige Feldnamen vor. Solche Hinweise können beim Entwickeln helfen, aber Details des Schemas verraten.

- Pröfe, ob Fehlermeldungen Feldnamen oder interne Details offenlegen.
- Passe Vorschläge und Fehlermeldungen an die jeweilige Umgebung und das Bedrohungsmodell an.
- Behnadle diese Maßnahme als ergänzende Informationskontrolle, nicht als Ersatz für Autorisierung.

7. EINGABETYPEN VALIDIEREN



GraphQL-Typen und Argumentdefinitionen prüfen grundlegende Form und Typverträglichkeit. Fachliche Regeln müssen zusätzlich serverseitig geprüft werden.

- Validiere Wertebereiche, Längen, erlaubte Zustände und fachliche Beziehungen.
- Behnadle Client-Eingaben grundsätzlich als nicht vertrauenswürdig, auch wenn sie typisiert sind.
- Begrenze Listen- und Seitengrößen und weise unerwartete Werte sicher zurück.
- Unterscheide Eingabevalidierung von Berechtigungsprüfung: Beides ist erforderlich.

8. BACKEND-INJECTION VERHINDERM



Resolver übergeben Eingaben häufig an Datenbanken, Suchsysteme oder Betriebssystemfunktionen. Unsichere Verkettung kann Injection ermöglichen.

- Nutze parametrisierte Abfragen oder sichere APIs des jeweiligen Backends.
- Setze Eingaben nicht ungeprüft in SQL-, Shell- oder Suchausdrücke ein.
- Validiere und normalisiere nach den Anforderungen des konkreten Datenziels; Escaping allein ist kein universeller Schutz.
- Verwende minimale Datenbank- und Dienstberechtigungen und behnadle Fehler sicher.

VOIP-SICHERHEIT

SIP steuert Sitzungen;
SRTP schützt Medien;
SBC, Segmentierung und Überwachung ergänzen den Schutz.

1. SIP-ARCHITEKTUR

- Signalisierung;
- INVITE, BYE;
- Proxy, Registrar;
- Medien separat.



2. SIP-DIGEST

- Challenge-Response;
- Nonce;
- starke Passwörter;
- Fehlversuche begrenzen.



3. TLS & SRTP

- TLS schützt SIP-Transport;
- SRTP schützt Medien;
- Schlüssel sicher verwalten;
- kein automatischer Ende-zu-Ende-Schutz.



4. SBC

- Zugangskontrolle;
- Normalisierung;
- Topologieschutz;
- Updates und Kapazität.



5. TOLL FRAUD

- Kontenmissbrauch;
- ungewöhnliche Ziele;
- Berechtigungen;
- Kostenalarme.



6. VISHING & CALLER ID

- Täuschung;
- Rufnummer kein Identitätsbeweis;
- Rückruf verifizieren;
- melden.



7. PCAP &RTP

- nur autorisierte Mitschnitte;
- RTP von SIP unterscheiden;
- SRTP schützt Medien;
- Datenzugriff begrenzen.



8. VOICE-VLAN, QoS & DDoS

- Segmentieren und filtern;
- QoS priorisiert, verschlüsselt nicht;
- Überlastung beobachten;
- Reaktionsweg.



MINDMAP 137: ZTNA-CONTROLLER UND ENKLAVENARCHITEKTUR

ZTNA-ENKLAVEN

Zero Trust Network Access (ZTNA) vermittelt den Zugriff auf einzelne Anwendungen anhand von Identität, Gerätezustand und Kontext. Eine Enklave ist dabei ein logisch abgegrenzter Bereich oder Zugriffspfad zu geschützten Ressourcen – kein pauschal vertrauenswürdiges internes Netz. Ein Controller koordiniert Richtlinien und Zugriffskomponenten; die konkrete Aufteilung unterscheidet sich je nach Implementierung.

- Kein implizites Vertrauen allein aufgrund des Netzwerkstandorts.
- Zugriff wird auf benötigte Ressourcen begrenzt.
- Identität und Kontext fließen in die Entscheidung ein.

MERKSATZ
Erst Identität und Richtlinie prüfen, dann den eng begrenzten Zugriff ermöglichen.

1. SOFTWARE-DEFINED PERIMETER (SDP)

Ein Software-Defined Perimeter verbirgt oder beschränkt Ressourcen, bis ein berechtigter Zugriff angefordert wird. Typisch ist die Trennung zwischen Steuerungsebene, die Zugriffsentscheidungen koordiniert, und Datenebene, die den erlaubten Verkehr vermittelt.

- Ressourcen sind nicht automatisch für alle erreichbar.
- Steuerung und Weiterleitung können getrennte Komponenten sein.
- SDP ist ein Architekturansatz, keine einzelne universelle Produkttechnik.

2. IDENTITÄTSZENTRIERTE ZUGRIFF-PROXIES

Ein Proxy vermittelt die Verbindung zwischen Nutzer oder Gerät und einer Anwendung. Die Autorisierung richtet sich vorrangig nach Identität und Richtlinie statt nach einem pauschalen Vertrauen in das Quellnetz.

- Authentifizierung bestätigt eine Identität; Autorisierung legt erlaubte Aktionen fest.
- Der Proxy kann den Zugriff auf bestimmte Anwendungen begrenzen.
- Verschlüsselung und sichere Konfiguration bleiben erforderlich.

3. GERÄTEZUSTAND UND POSTURE ATTESTATION

Vor oder während eines Zugriffs kann die Lösung Gerätesignale auswerten, etwa registrierten Status, Betriebssystemversion oder vorhandene Schutzmechanismen. Attestation bezeichnet eine überprüfbare Aussage über Eigenschaften eines Geräts oder seiner Umgebung; Umfang und Vertrauenswürdigkeit hängen von der Umsetzung ab.

- Signale sind keine Garantie für ein risikofreies Gerät.
- Aktualität und Herkunft der Messwerte berücksichtigen.
- Bei fehlender oder veralteter Attestation Zugriff einschränken oder erneut prüfen.

WICHTIG
Ein einzelner Gerätescheck ersetzt weder laufende Überwachung noch weitere Schutzmaßnahmen.

4. KURZLEBIGER ZUGRIFF UND JUST-IN-TIME-BASTIONS

Zeitlich begrenzte Berechtigungen reduzieren die Dauer unnötig offener Zugänge. Eine Bastion beziehungsweise ein Jump Host kann administrativen Zugriff über einen kontrollierten Zwischenpunkt führen.

- Rechte nur für den benötigten Zweck und Zeitraum vergeben.
- Zugriffe nachvollziehbar protokollieren und regelmäßig überprüfen.
- Kurzlebige Berechtigungen sicher widerrufen oder auslaufen lassen.

5. MIKROSEGMENTIERUNG UND ZERO-TRUST-PODS

Mikrosegmentierung teilt Systeme und Arbeitslasten in kleinere Sicherheitsbereiche. Ein „Zero-Trust-Pod" ist eine mögliche Bezeichnung für eine eng begrenzte Gruppe von Ressourcen; der Begriff beschreibt keine einheitlich standardisierte technische Einheit.

- Kommunikation zwischen Segmenten nur nach konkreter Richtlinie erlauben.
- Seitliche Bewegungen eines Angreifers erschweren.
- Regeln an Anwendungen und Arbeitslasten ausrichten, nicht nur an IP-Bereichen.

6. KONTINUIERLICHE RISIKOBEWERTUNG

Das Risiko kann sich während einer Sitzung ändern, etwa durch neue Gerätesignale, ungewöhnliches Verhalten oder geänderte Bedrohungslage. Eine kontinuierliche Bewertung kann deshalb eine erneute Prüfung oder Anpassung des Zugriffs auslösen.

- Kontextsignale regelmäßig und verhältnismäßig auswerten.
- Bei erhöhtem Risiko Sitzung begrenzen, erneut authentifizieren oder beenden.
- Fehlalarme oder Auswirkungen auf legitime Arbeit berücksichtigen.

7. API-GESTEUERTE ZUGRIFFSPOLICIES

APIs ermöglichen es, Richtlinien programmatisch zu verwalten oder ZTNA-Komponenten mit Identitäts-, Geräte- und Überwachungssystemen zu verbinden. Änderungen an Policies sind sicherheitsrelevant und müssen kontrolliert erfolgen.

- Zugriff auf APIs mit minimal nötigen Rechten absichern.
- Änderungen versionieren, protokollieren und nachvollziehbar freigeben.
- Fehlerhafte oder zu weit gefasste Regeln testen, bevor sie wirksam werden.

8. GEGENÜBERSTELLUNG ZU IPSEC- UND SSL-VPNS

ZTNA und VPNs lösen verwandte, aber nicht identische Zugriffsprobleme. Klassische VPNs stellen häufig eine Netzwerkverbindung zu einem entfernten Netz oder Segment her; ZTNA zielt typischerweise auf anwendungsbezogene, richtliniengesteuerte Zugriffe. Es gibt Überschneidungen und unterschiedliche Produktmodelle – ein pauschaler Ersatz ist nicht immer passend.

- VPNs können weiterhin für bestimmte Netze und Anwendungsfälle geeignet sein.
- ZTNA kann den Zugriff stärker auf einzelne Anwendungen begrenzen.
- Auswahl nach Bedrohungsmodell, Ressourcen, Identitätsintegration und Betrieb treffen.

ZUSAMMENFASSUNG
ZTNA-Enklaven verbinden Identitätsprüfung, Gerätekontext, eng gefasste Richtlinien und kontrollierte Vermittlung. Sie ergänzen eine Sicherheitsarchitektur, ersetzen aber nicht automatisch alle VPN- oder Netzwerkschutzfunktionen.

INCIDENT POST-MORTEM

Strukturierte, sachliche Nachbereitung eines Sicherheitsvorfalls: verstehen, was geschah, wirksam verbessern und Erkenntnisse in die Organisation zurückführen.

- Ziel: Lernen und künftige Reaktion stärken – nicht Schuldige suchen.

1. BLAMELESS POST-MORTEM-KULTUR



Eine faire Auswertung trennt Lernfragen von persönlicher Schuldzuweisung. Sie betrachtet Entscheidungen im damaligen Kontext und schafft Raum, Unsicherheiten und Fehler offen zu benennen.

- Respektvoll und faktenorientiert moderieren
- Systembedingungen und Informationslage berücksichtigen
- Vertraulichkeit und Beteiligung klar regeln

2. URSACHENANALYSE: 5-WHY UND ISHIKAWA



Ursachenanalyse fragt nach beitragenden Bedingungen, nicht nur nach dem letzten sichtbaren Ereignis. Die 5-Why-Methode verfolgt eine Ursache-Kette; ein Ishikawa-Diagramm ordnet mögliche Einflussfaktoren in Kategorien.

- Beobachtungen von Annahmen trennen
- Mehrere Ursachen können zusammenwirken
- Ergebnisse anhand von Belegen prüfen

3. CHRONOLOGISCHE TIMELINE REKONSTRUIEREN



Eine gemeinsame Ereignisfolge macht Zusammenhänge und offene Fragen sichtbar. Zeitstempel können aus unterschiedlicden Systemen stammen und müssen deshalb mit Zeitzone und Quelle eingeordnet werden.

- Erkennung, Meldung, Eindämmung und Wiederherstellung erfassen
- Zeitquelle und Zeitzone notieren Unsichere Zeitpunkte kenntlich machen

4. DETECTION- UND RESPONSE-GAPS ERKENNEN



Prüfen, wo vorhandene Kontrollen oder Abläufe den Vorfall nicht rechtzeitig erkannt oder wirksam bearbeitet haben. Ein Gap ist eine konkrete Verbesserungschance, nicht automatisch ein individualdes Versagen.

- Welche Signale waren verfügbar?
- Wo entstanden Verzögerungen oder Informationslücken?
- Welche Abhängigkeiten behinderten die Reaktion?

5. PRIORISIERTE KORREKTUR-MASSNAHMEN (CAPA)



Korrektur- und Vorbeugemaßnahmen sollen Ursachen und Risiken adressieren. Priorisierung verbindet erwartete Risikominderung mit Aufwand, Abhängigkeiten und Verantwortlichkeit.

- Maßnahme, Verantwortliche und Termin festhalten
- Erfolgskriterium und Nachweis definieren
- Fortschritt nachverfolgen und bei Bedarf neu priorisieren

6. DOKUMENTATION UND STAKEHOLDER-BERICHT



Ein klarer Bericht hält Umfang, Auswirkungen, belegte Erkenntnisse, Unsicherheiten und beschlossene Maßnahmen fest. Inhalt und Detailtiefe richten sich nach Rolle und berechtigtem Informationsbedarf.

- Fakten und Hypothesen sichtbar unterscheiden
- Sensible Informationen angemessen schützen
- Entscheidungen und offene Punkte dokumentieren

7. METRIKEN VERBESSERN: MTTD UND MTTR



MTTD bezeichnet die mittlere Zeit bis zur Erkennung; MTTR wird häufig als mittlere Zeit bis zur Wiederherstellung oder Behebung verwendet. Da Definitionen variieren, müssen Start- und Endpunkte festgelegt werden, bevor Kennzahlen verglichen werden.

- Einheitliche Ereignisdefinitionen verwenden
- Zeitraum und Datenqualität mitberichten
- Kennzahlen als Hinweise verstehen, nicht isoliert

Wichtig

MTTR ist mehrdeutig. Im Bericht ausdrücklich angeben, ob damit Reaktion, Behebung oder Wiederherstellung gemeint ist.

8. ERKENNTNISSE IN INCIDENT-PLAYBOOKS INTEGRIEREN



Ein Playbook wird nützlicher, wenn konkrete Erkenntnisse in überprüfbare Abläufe, Rollen und Kommunikationswege übersetzt werden. Änderungen sollten nachvollziehbar versioniert und geübt werden.

- Schritte und Eskalationswege aktualisieren
- Zuständigkeiten und benötigte Kontakte prüfen
- Änderungen durch Übungen oder Simulationen erproben

TABLETOP-ÜBUNGEN

Eine Tabletop-Übung ist eine moderierte, meist diskussionsbasierte Simulation eines Sicherheitsvorfalls. Das Team prüft Entscheidungen, Zuständigkeiten und Kommunikationswege anhand eines Szenarios – ohne produktive Systeme anzugreifen oder tatsächlich zu verändern.

- Ziel: Abläufe und Zusammenarbeit testen, nicht einzelne Personen bewerten.
- Die Übung macht Lücken sichtbar; sie ersetzt keinen technischen Wiederherstellungstest.

SICHER ÜBEN

Nur autorisierte Übungsdaten und klar abgegrenzte Szenarien verwenden. Reale Systeme, Konten oder Personen nicht ohne ausdrückliche Freigabe einbeziehen.

1. SZENARIODESIGN



Wähle einen plausiblen Vorfall, der zu den Risiken und Abläufen der Organisation passt.

- Ransomware: Systeme sind nicht verfügbar; Wiederherstellbarkeit ist unklar.
- Supply Chain: Ein Dienstleister oder Softwarelieferant meldet einen Sicherheitsvorfall.
- Datenleck: Hinweise deuten auf unbefugten Zugriff oder Offenlegung hin.
- Lege Umfang, Übungsziel, Startlage und Grenzen fest.

2. INJEKTIONEN UND ESKALATIONSPFADE



Injektionen sind vorbereitete Ereignisse oder Informationen, die den Übungsverlauf weiterentwickeln. Sie sollen Entscheidungen anregen und können je nach Reaktion unterschiedliche Anschlussfragen auslösen.

- Beipsiele: Alarmmeldung, Medienanfrage, ausgefallener Dienst oder neue Erkenntnis zur Datenlage.
- Eskalationen schrittweise und nachvollziehbar einführen.
- Die Moderation hält fest, welche Information wann verfügbar war.

3. ROLLENVERTEILUNG



Beteiligte handeln innerhalb ihrer tatsächlichen Zuständigkeiten. Die Übungsleitung erklärt Rollen und Grenzen vor Beginn.

- C-Level: Prioritäten und wesentliche Geschäftsentscheidungen.
- Recht: rechtliche Fragen und mögliche Meldepflichten prüfen.
- IT und Technik: Lagebild, Eindämmungsoptionen und Wiederherstellung einordnen.
- PR: abgestimmte Kommunikation mit Mitarbeitenden, Kundschaft und Öffentlichkeit unterstützen.

Keine Rechtsberatung simulieren
Die Übung kann Entscheidungswege für Rechtsfragen testen; sie ersetzt keine konkrete rechtliche Prüfung.

4. KRISENSTABS-KOMMUNIKATION



Der Krisenstab braucht klare Wege, um Informationen zu teilen, Entscheidungen festzuhalten und Rückfragen zu bündeln.

- Bestimme Haupt- und Ersatzkanäle sowie Kontaktwege.
- Trenne bestätigte Fakten, Annahmen und offene Fragen.
- Dokumentiere Entscheidung, verantwortliche Person und Zeitpunkt.
- Prüfe, wie das Team bei Ausfall eines bevorzugten Kanals weiterarbeitet.

5. NOTFALLHANDBUCH-VALIDIERUNG



Die Übung prüft, ob das Notfallhandbuch in der Situation auffindbar, verständlich und handlungsleitend ist.

- Sind Kontakte und Zuständigkeiten aktuell auffindbar?
- Sind Eskalationskriterien und Freigaben verständlich?
- Sind Abhängigkeiten und Alternativwege beschrieben?
- Erfasst Verbesserungsbedarf, statt die Übung als vollständigen Techniktest zu behandeln.

6. BEOBACHTUNG UND SCHIEDSRICHTERWESEN



Beobachtende erfassen den Ablauf möglichst neutral. Die Übungsleitung steuert Zeit und Informationsfluss, Schiedsrichter klären Regeln und Auslegung des Szenarios.

- Beobachte Entscheidungen, Übergaben, Verzögerungen und Informationsbedarf.
- Notiere konkrete Situationen statt pauschaler Urteile.
- Trenne Beobachtung von Interpretation.
- Greife nur ein, wenn Sicherheit, Übungsgrenzen oder Verständlichkeit es erfordern.

7. DEBRIEFING UND HEISSAUSWERTUNG



Direkt nach der Übung sammelt das Team Eindrücke, solange Entscheidungen und offene Fragen noch präsent sind.

- Was lief wie vorgesehen?
- Wo fehlten Informationen, Zuständigkeiten oder Entscheidungsbefugnisse?
- Welche Annahmen erwiesen sich als unsicher?
- Formuliere konkrete Verbesserungsmaßnahmen mit Verantwortlichen und Zielterminen.

8. REIFEGRAD DER REAKTIONSBEREITSCHAFT



Bewerte die Reaktionsbereitschaft anhand beobachtbarer Fähigkeiten und vereinbarter Kriterien – nicht anhand eines unbegründeten Gesamturteils.

- Beispiele: Erreichbarkeit, Lagebild, Entscheidungsweg, Kommunikationsübergabe und Nachverfolgung von Maßnahmen.
- Halte Stärken, Lücken und Unsicherheiten getrennt fest.
- Vergleiche spätere Übungen mit denselben Kriterien, um Fortschritt sichtbar zu machen.
- Ein Ergebnis beschreibt den Übungsstand, nicht die Garantie für den Ernstfall.

MINDMAP 140 · EXTORTION RESPONSE

EXTORTION RESPONSE

Geordnet auf Erpressung reagieren: Menschen schützen, Lage belegen, rechtliche Pflichten beachten und Entscheidungen nachvollziehbar treffen.

- Sicherheit und Eindämmung zuerst
- Fakten von Forderungen trennen
- Entscheidungen dokumentieren



1 · TRIAGE BEI MEHRFACH-ERPRESSUNG



Angreifer können mehrere Druckmittel gleichzeitig einsetzen. Jede Behauptung ist zunächst eine zu prüfende Information.

- Verschlüsselung: betroffene Systeme und Wiederherstellbarkeit erfassen
- Datenabfluss: Umfang und mögliche Bertoffene vorsichitg prüfen
- DDoS: Verfügbarkeit und betroffene Dienste beobachten
- Zeitpunkte, Quellen und Auswirkungen protokollieren

2 · EXTERNE BERATUNG UND VERHANDLUNG



Eine koordinierte Ansprechstruktur verhindert widersprüchliche Aussagen und unbedachte Zusagen.

- Incident-Response- und Forensik-Fachleute nach Bedarf einbinden
- Rechtsberatung und zuständige Leitung hinzuziehen
- Verhandler nur mit klarer Rolle und Freigabe einsetzen
- Keine Wiederherstellung oder Datenlöschung versprechen, bevor sie geprüft ist

3 · RECHTLICHE RAHMENBEDINGUNGEN



Zahlungen, Daten und Kommunikation können rechtliche Fragen auslösen. Zuständigkeit und anwendbare Regeln hängen vom konkreten Sachverhalt ab.

- Rechtsberatung zu Melde-, Datenschutz- und Aufbewahrungspflichten einholen
- Sanktions- und Zahlungsrisiken prüfen lassen
- OFAC ist die US-Behörde Office of Foreign Assets Control; ihre Regeln können je nach Verbindung und Sachverhalt relevant sein
- Keine rechtliche Freigabe aus einer allgemeinen Checkliste ableiten

4 · PROOF-OF-LIFE UND DECRYPTOR-TESTS



Behauptungen zu gestohlenen Daten oder Entschlüsselung nicht ungeprüft übernehmen. Tests dürfen Beweise und produktive Systeme nicht gefährden.

- Nachweise auf Echtheit und Aussagekraft prüfen
- Decryptor nur in isolierter, kontrollierter Umgebung testen
- Kopien und Integrität relevanter Belege sichern
- Ergebnisse, Einschränkungen und Fehlversuche dokumentieren

5 · KRYPTOWÄHRUNG UND WALLET-COMPLIANCE



Blockchain-Transaktionen können öffentlich nachvollziehbar sein; eine Adresse identifiziert jedoch nicht automatisch eine Person oder Organisation.

- Wallet-Adressen, Transaktionskennungen und Zeitpunkte sichern
- Analyseergebnisse als Hinweise, nicht als sichere Identitätszuordnung behandeln
- Compliance- und Sanktionsfragen vor jeder Zahlung fachlich prüfen lassen
- Keine eigenmächtige Überweisung oder Weitergabe von Zugangsdaten

6 · KOMMUNIKATION AUF ERPRESSER-PORTALEN

Wenn ein Portal genutzt wird, sollte Kommunikation kontrolliert, knapp und beweiswahrend erfolgen.

- Zugriff nur über genehmigte, sichere Verfahren
- Keine unnötigen Angaben zu Personen, Systemen oder Abwehrmaßnahmen teilen
- Nachrichten und Zeitstempel rechtmäßig und sicher dokumentieren
- Keine Links oder Dateien unkritisch öffnen



8 · ENTSCHEIDUNGSMATRIX ZUR LÖSEGELDZAHLUNG



Eine Zahlung ist keine Garantie für Entschlüsselung, Löschung gestohlener Daten oder das Ende weiterer Angriffe. Entscheidungen gehören in die zuständige Leitung und müssen rechtlich geprüft werden.

- Sichere Wiederherstellungsmöglichkeiten und Ausfallfolgen bewerten
- Risiken für Menschen, Daten und Betrieb abwägen
- Rechtliche, regulatorische und versicherungsvertragliche Fragen klären
- Entscheidung, Begründung, Freigaben und Alternativen dokumentieren

Entscheidungshilfe

Bei unklarer Rechtslage, möglichem Sanktionsbezug oder fehlender Freigabe: Zahlung nicht veranlassen; zuständige Rechtsberatung und Leitung einschalten.

Sind Sicherheit, Rechtslage, Zahlungsrisiken und Freigaben ausreichend geklärt?

Nein oder unklar

Keine Zahlung auslösen. Lage sichern, zuständige Leitung und qualifizierte Rechts- sowie Incident-Response-Beratung einbeziehen.

Ja, geprüft und freigegeben

Entscheidung samt Annahmen, Alternativen und Zuständigkeiten dokumentieren; Umsetzung nur nach genehmigtem Verfahren.

WICHTIG

Diese Mindmap ist eine Lernhilfe, keine Rechtsberatung oder technische Einsatzanweisung. Bei einem realen Vorfall sofort den organisationsinternen Krisenprozess und qualifizierte Fachstellen einbeziehen.

CYBER-VERSICHERUNG

Finanzieller Risikotransfer – kein Ersatz für Prävention.

1. ZWECK UND RISIKOTRANSFER



Risikotransfer verlagert vertraglich definierte finanzielle Folgen teilweise auf einen Versicherer. Das zugrunde liegende Risiko bleibt bestehen.

- Mögliche Ziele: Liquidität sichern und Kostenfolgen planbarer machen.
- Selbstbehalt und Sublimits verbleiben beim versicherten Unternehmen.
- Prävention und betriebliche Resilienz bleiben eigene Aufgaben.

2. TYPISCHE DECKUNGSBAUSTEINE



Policen kombinieren je nach Angebot unterschiedliche Bausteine. Umfang und Auslöser müssen im konkreten Wortlaut geprüft werden.

- Eigenschäden: etwa Wiederherstellung oder bestimmte Betriebsunterbrechungskosten.
- Drittschäden: mögliche Haftpflichtansprüche und vereinbarte Abwehrkosten.
- Incident Response: gegebenenfalls Forensik, Rechtsberatung oder Krisenkommunikation.
- Zusatzleistungen können eingeschränkt, limitiert oder an Bedingungen geknüpft sein.

3. UNDERWRITING: RISIKOPRÜFUNG



Beim Underwriting bewertet der Versicherer Angaben zum Unternehmen, zu Risiken und bestehenden Kontrollen. Angaben sollten vollständig, nachvollziehbar und auf den beantragten Schutz bezogen sein.

- Branche, Umsatz, Standorte und Abhängigkeiten beschreiben.
- Datenarten, Systeme und externe Dienstleister erfassen.
- Vorhandene Schutzmaßnahmen und bekannte Vorfälle konkret angeben.
- Zusatzleistungen in Unterlagen nicht als gesicherte Tatsache darstellen.

4. SICHERHEITSANFORDERUNGEN



Versicherer können Sicherheitskontrollen verlangen oder bei der Risikobewertung berücksichtigen. Es gibt keinen universellen Satz an Kontrollen, der für jede Police gleichermaßen gilt.

- Mehr-Faktor-Authentifizierung für besonders kritische Zugänge einsetzen.
- Backups getrennt sichern und Wiederherstellung regelmäßig testen.
- Systeme zeitnah aktualisieren und Schutz vor Schadsoftware betreiben.
- Zugriffe nach Aufgaben begrenzen; Notfall- und Meldewege festlegen.
- Konkrete Anforderungen und Nachweispflichten der Police prüfen.



WICHTIG

Eine genannte Sicherheitsmaßnahme ist nicht automatisch eine Garantie für Deckung. Entscheidend bleiben Vertragswortlaut und Umstände des Vorfalls.

5. DECKUNGSGRENZEN UND KOSTEN



Die Entschädigung kann durch Höchstbeträge, Teilgrenzen, Selbstbehalte, Wartezeiten und weitere Bedingungen begrenzt sein.

- Versicherungssumme: vereinbarte Obergrenze nach Vertragsregeln.
- Sublimit: niedrigere Grenze für bestimmte Leistungen oder Ereignisse.
- Selbstbehalt: vereinbarter Anteil, den der Versicherungsnehmer trägt.
- Wartezeit oder Karenz kann insbesondere bei Betriebsunterbrechung relevant sein.
- Kostenarten, Bewertungsgrundlagen und Anrechnung anderer Leistungen beachten.

6. SCHADENSFALL UND MELDEWEG



Bei einem Verdacht zählen schnelle Eindämmung, sichere Beweissicherung und die Einhaltung der vertraglichen Melde- und Abstimmungspflichten. Notfallmaßnahmen dürfen nicht durch die Suche nach Deckungsgewissheit verzögert werden.

- Incident-Response-Plan aktivieren und Zuständigkeiten klären.
- Vorfall und Zeitverlauf dokumentieren; relevante Belege geschützt sichern.
- Versicherer oder benannte Meldestelle fristgerecht nach Police informieren.
- Vorgesehene Dienstleister und Freigabeprozesse beachten.
- Behördliche, gesetzliche und vertragliche Pflichten getrennt prüfen.



IM ERNSTFALL

Akute Gefahren eindämmen und interne Notfallwege nutzen. Vertragsbedingungen geben vor, welche Meldungen und Abstimmungen gegenüber dem Versicherer erforderlich sind.

7. AUSSCHLÜSSE UND STREITPUNKTE



Ausschlüsse und Einschränkungen sind vertragsabhängig. Ein Cybervorfall allein bedeutet nicht automatisch, dass jede Kostenposition versichert ist.

- Ausgeschlossene Ereignisse oder Schadensarten im Wortlaut nachlesen.
- Vorsatz, bekannte Umstände oder Pflichtverletzungen können relevant sein; genaue Wirkung ist Vertragsfrage.
- Krieg, staatliche Akteure oder Infrastruktor-Ausfälle können besondere Klauseln betreffen.
- Sanktionsregeln, Kausalität und Nachweisfragen können die Prüfung beeinflussen.
- Begriffe und Ausnahmen niemals aus einer anderen Police übertragen.

8. RISIKOTRANSFER SINNVOLL EINORDNEN



Ein belastbarer Versicherungsentscheid verbindet Risikobild, Sicherheitsniveau, finanzielle Tragfähigkeit und Vertragsdetails. Versicherbarkeit ist kein Ersatz für Risikosteuerung.

- Kritische Ausfall- und Wiederherstellungskosten abschätzen.
- Deckungsbausteine, Ausschlüsse, Limits und Selbstbehalt vergleichen.
- Sicherheitskontrollen und Nachweise laufend pflegen.
- Notfallübungen durchführen und Erkenntnisse umsetzen.
- Police bei wesentlichen Änderungen erneut auf Passung prüfen.

142 · NIS-2-RICHTLINIE UND REGULATORISCHE COMPLIANCE

1. NIS-2: ÜBERBLICK



Die NIS-2-Richtlinie (EU) 2022/2555 stärkt die Cybersicherheit in der EU. Sie verpflichtet bestimmte Einrichtungen zu Risikomanagement und zur Meldung erheblicher Sicherheitsvorfälle. Ob eine Organisation erfasst ist und welche nationalen Pflichten gelten, hängt unter anderem von Sektor, Größe und Umsetzung im jeweiligen Mitgliedstaat ab.

- Ziel: ein gemeinsames hohes Cybersicherheitsniveau
- Betroffene Sektoren umfassen unter anderem Energie, Gesundheit, digitale Infrastruktur und öffentliche Verwaltung
- Nationale Umsetzung und zuständige Behörden beachten



EINORDNUNG
Diese Mindmap ist eine Lernübersicht, keine Rechtsberatung. Prüfe für konkrete Pflichten die geltenden nationalen Vorschriften und behördlichen Hinweise.

2. KRITIKALITÄTSKATEGORIEN



NIS-2 unterscheidet grundsätzlich zwischen wesentlichen und wichtigen Einrichtungen. Die Einordnung richtet sich nach den Kriterien der Richtlinie und ihrer nationalen Umsetzung; die Kategorie beeinflusst insbesondere Aufsicht und Durchsetzung.

- Wesentliche Einrichtungen: typischerweise besonders kritische Sektoren oder größe Einrichtungen nach den Kriterien
- Wichtige Einrichtungen: weitere erfasste Einrichtungen in den festgelegten Sektoren
- Sektor, Größe, Sonderfälle und nationale Regeln gemeinsam prüfen

3. RISIKOMANAGEMENTMAßNAHMEN



Erfasste Einrichtungen müssen geeignete und verhältnismäßige technische, operative und organisatorische Maßnahmen umsetzen. Der Ansatz berücksichtigt Risiken, Eintrittswahrscheinlichkeit und mögliche Auswirkungen.

- Konzepte für Risikoanalyse und Informationssystemsicherheit
- Bewältigung von Vorfällen sowie Aufrechterhaltung des Betriebs
- Sicherheit der Lieferkette und Umgang mit Schwachstellen
- Wirksamkeit der Maßnahmen regelmäßig überprüfen

4. MELDEPFLICHTEN BEI VORFÄLLEN



Für erhebliche Sicherheitsvorfälle sieht NIS-2 eine gestufte Meldung an die zuständige Stelle oder das CSIRT vor. Die Fristen laufen ab Kenntnis des erheblichen Vorfalls; maßgeblich sind die anwendbaren Vorschriften und Meldewege.

- Frühwarnung: möglichst binnen 24 Stunden
- Vorfallmeldung: binnen 72 Stunden
- Abschlussbericht: grundsätzlich spätsetens einen Monat nach der Vorfallmeldung
- Bei fortdauerndem Vorfall kann ein Fortschrittsbericht erforderlich sein; Meldeweg der zuständigen Stelle nutzen



FRISTEN RICHTIG LESEN
Die 24- und 72-Stunden-Fristen beziehen sich auf unterschiedliche Meldestufen. Nicht jeder Sicherheitsvorfall erfüllt automatisch die Kriterien eines erheblichen Vorfalls.

5. VERANTWORTUNG DER LEITUNG



Leitungsorgane müssen die Cybersicherheitsmaßnahmen billigen und ihre Umsetzung überwachen. NIS-2 verlangt außerdem, dass Leitungsmitglieder zu Cybersicherheitsrisiken geschult werden.

- Maßnahmen nicht nur delegieren: Billigung und Aufsicht bleiben zentral
- Schulung unterstützt fundierte Entscheidungen und Risikoverständnis
- Nationale Umsetzung besitmmt konkrete Verantwortlichkeiten und Verfahren

6. AOFSICHT UND SANKTIONEN



Die Richtlinie sieht Aufsichts- und Durchsetzungsbefugnisse sowie mögliche Geldbußen vor. Für wesentliche Einrichtungen gelten höhere unionsrechtliche Höchstgrenzen als für wichtige Einrichtungen; die konkrete Anwendung erfolgt nach nationalem Recht.

- Wesentliche Einrichtungen: Höchstgrenze mindestens 10 Mio. Euro oder 2 % des weltweiten Jahresumsatzes, je nachdem, welcher Betrag höher ist
- Wichtige Einrichtungen: Höchstgrenze mindestens 7 Mio, Euro oder 1,4 % des weltweiten Jahersumsatzes, je nachdem, welcher Betrag höher ist
- Das sind Obergrenzen, keine automatisch festgesetzten Bußgelder
- Leitungsrerantwortung und persönliche Folgen richten sich nach nationalem Recht



KEINE PAUSCHALAUSSAGE
Ob und in welcher Höhe eine Sanktion verhängt wird, hängt vom Einzelfall, den zuständigen Behörden und der nationalen Umsetzung ab.

7. VORBEREITUNG



Eine strukturierte Bestandsaufnahme hilft, mögliche Pflichten und Lücken zu erkennen. Beginne mit Zuständigkeit, Anlagen und Meldewegen; dokumentiere Entscheidungen und Verbesserungen.

- Einrichtung, Tätigkeiten und mögliche Sektorzuurdnung erfassen
- Kritische Systeme, Abhängigkeiten und Lieferanten dokumentieren
- Risiken, bestehende Maßnahmen und Verantwortliche abglelichen
- Meldekontakte, Eskalationsweg und Übungsszenarien festlegen

8. MERKSÄTZE



NIS-2 verbindet Risikomanagement, Vorfallmeldung, Aufsicht und Verantwortung der Leitung. Entscheidend ist nicht allein ein einzelnes Sicherheitswerkzeug, sondern ein nachvollziehbarer und laufend überprüfter Organisationsansatz.

- Erst Anwendungsbereich und nationale Umsetzung klären
- Maßnahmen am Risiko ausrichten und ihre Wirksamkeit prüfen
- Erhebliche Vorfälle über den vorgesehenen Meldeweg und fristgerecht melden
- Leitung einbeziehen; Sanktionen nicht mit automatischen Bußgeldern verwechseln

143 DORA IM FINANZSEKTOR

IKT-Risiken · Vorfälle · Resilienztests · Drittparteien

1. ZIEL UND ANWENDUNGSBEREICH



DORA schafft unionsweit einen Rahmen, damit Finanzunternehmen IKT-Störungen bewältigen und ihre kritischen Tätigkeiten fortführen oder geordnet wiederherstellen können.

- Erfasst zahlreiche Finanzunternehmen und bestimmte IKT-Drittdienstleister; konkrete Einordnung hängt von Rechtsform und Regelung ab.
- Proportionalität berücksichtigt Größe, Risikoprofil und Komplexität.
- Das Management trägt Verantwortung für den IKT-Risikomanagementrahmen.
- Resilienz umfasst Vorbereitung, Reaktion, Wiederherstellung und Lernen.

2. IKT-RISIKOMANAGEMENT



Ein strukturierter Rahmen verbindet Governance, Schutzmaßnahmen und laufende Verbesserung.

- Rollen, Verantwortlichkeiten und Eskalationswege festlegen.
- IKT-Systeme, Daten, Abhängigkeiten und Risiken erfassen.
- Schutz und Prävention risikobasiert planen.
- Erkennung, Reaktion und Wiederherstellung vorbereiten.
- Kontrollen, Richtlinien und Pläne regelmäßig überprüfen.

2. IKT-VORFÄLLE KLASSIFIZIEREN UND MELDEN



Finanzunternehmen müssen IKT-Vorfälle erkennen, dokumentieren, klassifizieren und – sofern meldepflichtig – über die vorgesehenen Kanäle melden. Kriterien und Fristen richten sich nach den einschlägigen DORA-Regelungen und technischen Standards.

- Mögliche Kriterien: betroffene Kunden oder Transaktionen, Dauer, geografische Ausbreitung, Datenverlust, Kritikalität betroffener Dienste und wirtschaftliche Auswirkungen.
- Schwellenwerte und Meldefristen nicht pauschal annehmen: maßgeblich sind geltende Vorgaben und der konkrete Fall.
- Beweismittel sichern und Entscheidungen nachvollziehbar dokumentieren.
- Nach dem Vorfall Ursachen analysieren und Abhilfemaßnahmen verfolgen.



EINORDNUNG

Nicht jede technische Störung ist automatisch ein meldepflichtiger schwerwiegender IKT-Vorfall. Die Klassifizierung erfolgt anhand der anwendbaren Kriterien.

8. MERKSEHEMA



DORA verbindet Gevernance, Risikomanagement, Vorfallbearbeitung, Tests und die Kontrolle externer IKT-Abhängigkeiten zu einem fortlaufenden Resilienzzyklus.

- Erfassen und bewerten: Systeme, Dienste, Daten und Abhängigkeiten.
- Schützen und erkennen: Kontrollen, Überwachung und Zuständigkeiten.
- Reagieren und melden: Vorfall klassifizieren, dokumentieren und erforderlichenfalls melden.
- Wiederherstellen und lernen: Dienste zurückführen, Ursachen beheben, Maßnahmen prüfen.
- Dritte und Tests: Verträge, Abhängigkeiten und angemessene Resilienztests laufend berücksichtigen.

4. RESILIENZTESTS



Tests prüfen, ob Schutz- und Wiederherstellungsmaßnahmen unter geeigneten Bedingungen funktionieren. Umfang und Häufigkeit sind risikobasiert festzulegen.

- Tests können etwa Schwachstellenanalysen, Szenarioübungen, Wiederherstellungstests oder technische Sicherheitstests umfassen.
- Feststellungen priorisieren, Verantwortliche benennen und Bebebung nachverfolgen.
- Tests kontrolliert planen: Auswirkungen auf produktive Systeme und Kunden berücksichtigen.
- Ergebnisse dokumentieren und in Verbesserungen überführen.

7. INFORMATIONSAUSTAUSCH UND ZUSAMMENARBEIT



DORA ermöglicht Finanzunternehmen unter bestimmten Bedingungen den freiwilligen Austausch von Cyberbedrohungsinformationen und Erkenntnissen innerhalb vertrauenswürdiger Vereinbarungen.

- Austausch kann Erkennung und Prävention unterstützen.
- Vertraulichkeit, Datenschutz, Wettbewerbsrecht und Informationssicherheit beachten.
- Informationen zweckgebunden teilen und Empfänger sowie Schutzbedarf berücksichtigen.

BEGRIFFE



DORA steht für Digital Operational Resilience Act. TLPT bedeutet Threat-Led Penetration Testing, also bedrohungsgeleitete Penetrationstests.

5. TLPT



Threat-Led Penetration Testing ist ein fortgeschrittener, bedrohungsgeleiteter Test für bestimmte Finanzunternehmen nach den DORA-Vorgaben. Er orientiert sich an relevanten Bedrohungsinformationen und simuliert gezielte Angriffe auf ausgewählte kritische Funktionen.

- Nicht jedes Unternehmen ist automatisch TLPT-pflichtig; die zuständige Aufsicht bestimmt den erfassten Kreis nach den geltenden Kriterien.
- Der Test wird kontrolliert geelant und von qualifizierten Testern durchgeführt.
- Umfang, Regeln, sichere Grenzen und Kommunikationswege vorab festlegen.
- Tests dürfen nicht mit unautorisierten Angriffen gleichgesetzt werden.
- Ergebnisse und Abhilfemaßnahmen werden nach dem vorgesehenen Verfahren behandelt.



SICHERHEIT ZUERST

Penetrationstests nur mit ausdrücklicher Autorisierung, vereinbartem Umfang und geeigneten Schutzmaßnahmen durchführen.

6. IKT-DRITTPARTEIENRISIKO



Die Nutzung externer IKT-Dienste hebt die Verantwortung des Finanzunternehmens für sein Risikomanagement nicht auf.

- Vor Beauftragung Risiken, Konzentrationen und Abhängigkeiten bewerten.
- Vertragliche Regelungen zu Leistung, Sicherheit, Vorfällen, Prüfungen und Beendigung berücksichtigen.
- Kritische oder wichtige Funktionen besonders bedarfs sorgfältig überwachen.
- Auslagerungs- und Exit-Szenarien sowie Datenrückgabe oder -übertragbarkeit planen.
- DORA sieht zudem einen Aufsichtsrahmen für als kritisch eingestufte IKT-Drittdienstleister vor.

KI-SICHERHEIT FÜR LLMs

Ein Sprachmodell verarbeitet Eingaben und erzeugt Ausgaben. Sicherheitsrisiken entstehen im Modell, in den Daten und in der Anwendung, die das Modell einbindet. Die Mindmap ordnet zentrale Gefahren und Schutzmaßnahmen ein.

- Modell, Daten, Schnittstellen und Nutzungskontext betrachten.
- Schutzmaßnahmen als mehrschichtige Kontrollen verstehen.
- Keine einzelne Abwehr verhindert alle Angriffe.



1. PROMPT INJECTION

Eine manipulierte Eingabe versucht, Anweisungen oder Kontext des Modells zu beeinflussen. Sie kann direkt vom Nutzer kommen oder indirekt aus abgerufenen Dokumenten und anderen externen Inhalten.

- Direkt: Nutzereingabe fordert zum Ignorieren vorheriger Regeln auf.
- Indirekt: ein Dokument enthält versteckte oder feindliche Anweisungen.
- Trenne nicht vertrauenswürdige Inhalte von Systemanweisungen.
- Prüfe Tool-Aufrufe und begrenze deren Berechtigungen.



2. UNSICHERE AUSGABEN

Modellausgaben sind nicht automatisch sicher, korrekt oder für eine Weiterverarbeitung geeignet. Ein Fehler entsteht, wenn Ausgaben ungeprüft als Code, HTML, SQL oder Steuerbefehl verwendet werden.

- Ausgaben vor Anzeige oder Ausführung kontextgerecht validieren.
- HTML und andere aktive Inhalte sicher kodieren oder bereinigen.
- Keine Modellausgabe direkt als privilegierten Befehl ausführen.
- Fehlerfälle und unerwartete Formate abfangen.



3. DATENVERGIFTUNG

Vergiftete Trainings- oder Feinabstimmungsdaten können Verhalten oder Ausgaben eines Modells gezielt beeinflussen. Auch manipulierte Wissensquellen können Antworten verfälschen, ohne das Modelltraining selbst zu verändern.

- Herkunft und Integrität von Datensätzen dokumentieren.
- Datenquellen prüfen und Zugriffe begrenzen.
- Anomalien und unerwartete Verhaltensänderungen untersuchen.
- Abrufquellen versionieren und Änderungen nachvollziehbar machen.



4. OFFENLEGUNG SENSIBLER INFORMATIONEN

Ein System kann vertrauliche Informationen aus Prompts, Kontexten, Trainingsdaten oder angebundenen Quellen preisgeben. Minimierung und Zugriffskontrolle verringern die Folgen einer unerwarteten Ausgabe.

- Keine unnötigen Geheimnisse in Prompts oder Kontexten speichern.
- Zugriff auf Daten nach dem Prinzip der geringsten Rechte begrenzen.
- Ausgaben auf sensible Inhalte prüfen, wo es angemessen ist.
- Protokolle so gestalten, dass sie keine Geheimnisse offenlegen.



5. MODELL- UND LIEFERKETTENRISIKEN

Modelle, Bibliotheken, Gewichte und Dienste stammen oft aus unterschiedlichen Quellen. Manipulierte oder verwundbare Komponenten können Risiken in die Anwendung bringen.

- Modell- und Paketquellen nachvollziehbar auswählen.
- Versionen und Abhängigkeiten inventarisieren.
- Änderungen an Modellen und Konfigurationen kontrolliert ausrollen.
- Drittanbieterzugriffe und Datenverarbeitung bewerten.



6. MODELLDIEBSTAHL UND MISSBRAUCH

Angreifer können versuchen, Modellverhalten nachzubilden, proprietäre Informationen zu gewinnnen oder bereitgestellte Funktionen zu missbrauchen. Häufige Abfragen und übermäßige Auskunft können das Risiko erhöhen.

- Zugriff authentifizieren und angemessen begrenzen.
- Abfragevolumen überwachen und auffällige Muster untersuchen.
- Interne Prompts und Konfigurationen nicht unnötig offenlegen.
- Missbrauchsgrenzen und Reaktionswege definieren.



7. GUARDRAILS UND MEHRSCHICHTIGE ABWEHR

Guardrails sind Kontrollen rund um ein KI-System, etwa Eingabeprüfungen, Richtlinien, Berechtigungen und Ausgabevalidierung. Sie ergänzen einander, sind aber keine Garantie gegen alle Umgehungen.

- Eingaben und abgerufene Inhalte als nicht vertrauenswürdig behandeln.
- Werkzeuge mit minimal nötigen Rechten ausstatten.
- Ausgaben prüfen, bevor sie Aktionen auslösen.
- Sicherheitskontrollen regelmäßig mit realistischen Fällen testen.



8. SICHERHEITSBETRIEB

Sichere LLM-Anwendungen benötigen Überwachung, Vorfallbehandlung und kontinuierliche Anpassung. Protokollierung sollte hlifreich sein, ohne unnötig personenbezogene oder vertrauliche Inhalte zu sammeln.

- Sicherheitsrelevante Ereignisse mit geeigneten Schutzmaßnahmen erfassen.
- Fehlverhalten und Richtlinienverstöße untersuchen.
- Vorfallreaktion und Eskalationswege festlegen.
- Erkenntnisse in Tests und Verbesserungen überführen.



Merksatz

LLM-Sicherheit umfasst Daten, Modell, Anwendung und Betrieb. Prüfe jede Ebene und hire Übergänge.

SHADOW IT & CASB

Shadow IT sind digitale Dienste und Anwendungen, die Mitarbeitende ohne Freigabe oder ausreichende Abstimmung mit der zuständigen Organisation nutzen.

Ein Cloud Access Security Broker (CASB) unterstützt dabei, Cloud-Dienste sichtbar zu machen und Sicherheitsrichtlinien anzuwenden. Er ergänzt Governance und technische Schutzmaßnahmen, ersetzt sie aber nicht.

- **Ziel:** Risiken erkennen und angemessen behandeln.
- **Leitidee:** erst sichtbar machen, dann bewerten und steuern.

WICHTIGE EINORDNUNG

Nicht jede unbekannte Cloud-Nutzung ist automatisch schädlich. Prüfe Zweck, Datenarten, Risiken und geltende Organisationsregeln, bevor du Maßnahmen festlegst.

1. SHADOW IT ERKENNEN



Unbekannte SaaS-Nutzung kann über mehrere Signale auffallen. Einzelne Signale beweisen jedoch weder eine Richtlinienverletzung noch einen Datenabfluss.

- Cloud-Dienst-Namen in DNS-- Proxy- oder Firewall-Protokollen
- Anmeldungen oder OAuth-Freigaben für bislang unbekannte Anwendungen
- Ungewöhnliche Uploads oder wiederkehrende Datenübertragungen
- Hinweise aus Inventaren, Kostenabrechnungen und Meldungen der Beschäftigten

2. SAAS-RISIKEN BEWERTEN



Bewerte einen Dienst im Kontext: Welche Daten werden verarbeitet, wer erhält Zugriff und welche Kontrollen sind vorhanden? Verwende nachvollziehbare Kriterien statt einer pauschalen Einstufung.

- Datenklassifizierung und möglicher Schutzbedarf
- Anbieterfunktionen für Identität, Protokollierung und Verschlüsselung
- Zugriffsrechte, Freigaben und externe Zusammenarbeit
- Datenstandort, Aufbewahrung und Löschmöglichkeiten gemäß geltenden Vorgaben
- Geschäftlicher Nutzen und verfügbare genehmigte Alternativen

3. DATENABFLUSS KONTROLLIEREN



CASB-Funktionen können helfen, risikoreiche Aktivitäten zu erkennen und Richtlinien durchzusetzen. Welche Sichtbarkeit möglich ist, hängt von Integration, Konfiguration und Verschlüsselung ab.

- Aktivitäten und Übertragungen protokollieren, soweit technisch verfügbar
- Richtlinien an Datenart, Nutzerrolle und Dienstkontext ausrichten
- Warnen, zusätzliche Authentifizierung verlangen oder Aktionen begrenzen
- Blockieren nur dort, wo es angemessen und organisatorisch freigegeben ist
- Ausnahmen und Entscheidungen nachvollziehbar dokumentieren

4. PROXY-MODUS



Im Proxy-Modus wird der Datenverkehr über einen vermittelnden Dienst geleitet. Je nach Architektur erfolgt die Einbindung beispielsweise explizit über Proxy-Einstellungen oder transparent über die Netzwerkinfrastruktur.

- Kann laufenden Verkehr untersuchen und Richtlinien zeitnah anwenden
- Wirksamkeit hängt davon ab, ob Nutzer und Geräte tatsächlich über den Proxy zugreifen
- TLS-Inspektion kann zusätzliche Vertrauens-, Datenschutz- und Betriebsfragen aufwerfen
- Nicht jeder Dienst oder jede App lässt sich ohne Einschränkungen vermitteln

5. API-MODUS



Im API-Modus verbindet sich der CASB mit den Schnittstellen eines Cloud-Anbieters. Damit lassen sich je nach Anbieter und Berechtigungen gespeicherte Inhalte, Konfigurationen oder Aktivitäten untersuchen.

- Kann Daten im Cloud-Dienst auch nach dem Upload prüfen
- Erfordert eine autorisierte Integration und passende API-Berechtigungen
- Abdeckung und Aktualität richten sich nach den Fähigkeiten des Anbieters
- Erfasst nicht automatisch jeden Zugriff oder jede Echtzeit-Aktion

6. RICHTLINIEN DURCHSETZEN



Eine gute Richtlinie ist verständlich, risikobasiert und überprüfbar. Berücksichtige den Geschäftszweck und informiere zuständige Stellen, bevor Kontrollen eingeführt oder verschärft werden.

- Beispiele erlaubter, eingeschränkt erlaubter und nicht genehmigter Dienste festlegen
- Datenklassen mit passenden Aktionen verknüpfen
- Rollen, Gerätevertrauen und Authentifizierungsstatus einbeziehen
- Ausnahmen mit Begründung, Zuständigkeit und Überprüfungstermin versehen
- Alarme auf klare Bearbeitungswege und verantwortliche Teams ausrichten

7. VORGEHEN BEI EINEM FUND



Ein Fund sollte strukturiert geprüft werden. Vermeide vorschnelle Sperren, die Arbeitsabläufe unterbrechen oder wichtige Belege verändern könnten.

- Signal und betroffenen Dienst erfassen
- Nutzung, Datenarten und Umfang anhand verfügbarer Protokolle prüfen
- Risiko und mögliche Auswirkungen mit zuständigen Verantwortlichen bewerten
- Angemessene Reaktion wählen: informieren, absichern, begrenzen oder genehmigte Alternative anbieten
- Entscheidung, Zuständigkeit und Nachkontrolle dokumentieren

8. KURZÜBERBLICK



CASB unterstützt die Sichtbarkeit und Steuerung von Cloud-Nutzung, doch kein einzelner Modus deckt alle Situationen ab. Kombiniere passende Kontrollen und überprüfe ihre Wirkung regelmäßig.

- Erkennen: Signale aus Netzwerk, Identität und Cloud-Diensten zusammenführen
- Bewerten: Kontext und Datenrisiko berücksichtigen
- Steuern: Proxy- und API-Funktionen gezielt einsetzen
- Schützen: Grenzen der Sichtbarkeit, insbossndere bei Verschlüsselung, einplanen
- Verbessern: Regeln, Ausnahmen und Reaktionswege regelmäßig überprüfen

MINDMAP 146 · INSIDER THREAT MANAGEMENT & UEBA

146 · INSIDER THREAT MANAGEMENT & UEBA

Verhaltenssignale im Kontext prüfen – nicht als Schuldbeweis behandeln.

1. BEGRIFF UND BEDROHUNGSBILD



Ein Insider-Risiko betrifft eine Person oder ein Konto mit berechtigtem Zugriff, das diesen absichtlich missbraucht oder unbeabsichtigt unsicher handelt. Auch kompromittierte Zugangsdaten können wie ein Insider-Konto wirken.

- Böswillig: Daten stehlen, verändern oder Zugänge missbrauchen.
- Unachtsam: Daten fehlversenden, Schutzregeln umgehen oder Geräte verlieren.
- Kompromittiert: Ein Angreifer nutzt legitime Zugangsdaten.
- Risiko hängt von Zugriff, Gelegenheit, Auswirkungen und Schutzmaßnahmen ab.

2. INSIDER THREAT MANAGEMENT



Ein tragfähiges Programm verbindet Prävention, Erkennung, Reaktion und Lernen. Zuständigkeiten und zulässige Datennutzung werden vorab festgelegt; Vorfälle werden koordiniert und nachvollziehbar bearbeitet.

- Rollen, Meldewege und Eskalationskriterien definieren.
- Zugriffe nach Aufgabe begrenzen und regelmäßig überprüfen.
- Mitarbeitende über sichere Abläufe und Meldemöglichkeiten informieren.
- Security, IT, Datenschutz und Personalwesen nur im erforderlichen Rahmen einbinden.

3. UEBA: VERHALTENSBASELINE



Eine Baseline beschreibt erwartbare Aktivitätsmuster für einen passenden Vergleichskontext. Sie kann individuelle, rollenbezogene oder systembezogene Muster berücksichtigen. Eine Baseline ist nicht unveränderlich: Aufgaben, Saisonalität und Systemänderungen beeinflussen Verhalten.

- Beipsiele: übliche Anmeldezeiten, genutzte Anwendungen und Datenzugriffe.
- Vergleichsgruppen müssen sinnvoll und ausreichend geschützt sein.
- Zeitraum, Datenquellen und bekannte Änderungen dokumentieren.
- Zu venig oder verzerrte Daten schwächen die Aussagekraft.

4. ANOMALIE-SCORING



Ein Anomalie-Score fasst Merkmale zusammen, die von einem gewählten Vergleichsmuster abweichen. Er priorisiert Prüfungen, misst aber nicht unmittelbar Absicht oder Schuld. Schwellenwerte und Regeln müssen zum Einsatzkontext passen.

- Signale kombinieren: Neuartigkeit, Umfang, Zeitpunkt und betroffene Systeme.
- Kontext ergänzen: Rolle, genehmigte Änderungen, Arbeitsabläufe und Ausnahmen.
- Trefferqualität anhand nachvollziehbarer Fälle bewerten.
- Fehlalarme und übersehene Ereignisse regelmäßig untersuchen.

SCORE IST KEIN URTEIL

Ein hoher Score begründet eine sorgfältige, abgestufte Prüfung. Entscheidungen über Personen benötigen geeignete Verfahren und menschliche Verantwortung.

5. DATA HOARDING ERKENNEN



Data Hoarding bezeichnet hier auffälliges Ansammeln oder Zusammenführen von Daten, das vom üblichen Arbeitsbedarf abweicht. Einzelne große Downloads sind nicht automatisch verdächtig: Sicherungen, Migrationen oder Projektarbeit können sie erklären.

- Beobachten: ungewöhnliche Massendownloads oder viele Exporte.
- Beachten: Zugriff auf neue Datenbestände oder ungewöhnliche Archivierung.
- Kontext prüfen: Auftrag, Genehmigung, Zielsystem und Zeitpunkt.
- Datenminimierung beachten; keine unnötigen Inhalte überwachen.

6. HR-IT-VERKNÜPFUNG



Eine abgestimmte Verbindung relevanter Personal- und IT-Prozesse kann helfen, Zugänge bei Rollenwechsel oder Austritt zeitgerecht anzupassen. Sie erfordert klare Zuständigkeiten, Zweckbindung und begrenzten Zugriff auf personenbezogene Informationen.

- Rollenwechsel: Berechtigungen an neue Aufgaben anpassen.
- Austritt: Konten, Sitzungen, Geräte und Schlüssel nach festgelegtem Ablauf behandeln.
- Freigaben und Zeitpunkte zwischen zuständigen Stellen koordinieren.
- Nur erforderliche Informationen teilen und Zugriffe protokollieren.

PRIVATSPHÄRE WAHREN

Personaldaten nicht pauschal mit Verhaltenssignalen zusammenführen. Zweck, Zugriff, Aufbewahrung und zulässige Nutzung müssen klar geregelt sein.

7. PRÜFUNG UND REAKTION

Ein Signal wird zunächst validiert und anschließend nach Schwere und möglicher Auswirkung eskaliert. Reaktionen sollen Belege sichern, Schaden begrenzen und berechtigte Abläufe nicht unnötig beeinträchtigen.

- Quelle, Zeitstempel und betroffene Identität oder Entität verifizieren.
- Mit unabhängigen Protokollen und autorisierten Verantwortlichen abgleichen.
- Bei akuter Gefahr abgestufte Schutzmaßnahmen nach Verfahren ergreifen.
- Entscheidungen, Begründung und Abschluss nachvollziehbar dokumentieren.



8. LEITGEDANKE



Ein Insider-Erkennung verbindet technische Hinweise mit organisatorischem Kontext und Schutz der Privatsphäre. Ziel ist eine faire, überprüfbare Risikobehandlung – nicht die lückenlose Beobachtung oder automatische Bewertung von Menschen.

- Baseline beschreibt Vergleich, nicht Norm für jede Person.
- Anomalie zeigt Abweichung, nicht Absicht.
- Score priorisiert, entscheidet aber nicht allein.
- Daten sparsam nutzen; Maßnahmen begründen und überprüfen.
- Kommunikation nachvollziehbar und verständlich gestalten.

MINDMAP 147 · SATELLITEN- UND RAUMFAHRT-CYBERSICHERHEIT

1 · SYSTEMSEGMENTE



Die Sicherheitsarchitektur beginnt mit einer Übersicht der Komponenten und ihrer Vertrauensgrenzen.

- Weltraumsegment: Satellitenbus, Nutzlast, Bordcomputer und Software.
- Bodensegment: Kontrollzentrum, Bodenstationen, Telemetrie- und Missionssysteme.
- Verbindungen: Uplink zur Übertragung zum Satelliten; Downlink zur Übertragung zur Erde.
- Nutzersegment: Terminals, Dienste und Schnittstellen zu externen Netzen.

2 · TT&C UND TELEMETRIE



TT&C steht für Telemetrie, Tracking und Command: Zustandsdaten empfangen, Satelliten verfolgen und Befehle übermitteln.

- Telemetrie beschreibt gemessene Zustände; sie ist nicht automatisch vertrauenswürdig.
- Kommandos können kritische Funktionen auslösen und brauchen strenge Zugriffskontrol kontrollen.
- Befehls- und Telemetriepfade inventarisieren und voneinander abgrenzen.
- Authentizität und Integrität von Befehlen prüfen; Verfügbarkeit der Kommunikation mitdenken.

BEGRIFFE TRENNEN

Uplink und Downlink beeeichnen die Übertragungs-richtung, Sie sind keine Verschhisselungsverfahnen und garantieren für sich allein keine Sicherheit.

8 · ÜBERWACHUNG UND REAKTION



Space Cyber verbindat technische Schutzmaßnaahmen mit vorbereiteten Betriebs- und Wiederherstellungsabläufen.

- Zeitstempel, Befehle, Anmeldungen und Systemzustände nachvollziehbar protokollieren.
- Abweichungen wie ungewöhnliche Kommandos, Zugriffsversuche oder Telemetrieänderungen untersuchen.
- Meldewege, Zuständigkeiten und sichere Betriebszustände vorab festlegen.
- Nach einem Ereignis Ursachen, Auswirkungen und Verbesserungen dokumentieren.

3 · BEDROHUNGEN DER FUNKSTRECKE



Funkverbindungen können gestört, abgehört oder durch falsche Signale beeinflusst werden.

- Abhören kann Vertraulichkeit gefährden, wenn Inhalte nicht angemessen geschützt sind.
- Manipulierte oder wiederholte Nachrichten können Integrität und Steuerung beeinträchtigen.
- Jamming stört die Funkvefügbarkeit; Verschlüsselung verhindert es nicht.
- Replay-Schutz, Protokollierung und Erkemung ungewöhnlicher Befehle unterstützen die Abwehr.

ZENTRUM · SPACE CYBER

Schutz vernetzter Weltraumsysteme: Satellit, Funkverbindungen und Bodensegment gemeinsam betrachten.

- Ziel: Vertraulichkeit, Integrität und Verfügbarkeit missionsrelevanter Daten und Funktionen sichern.



7 · KRYPTOGRAFIE UND SCHLÜSSEL



Kryptografische Schutzmaßnahmen müssen zum Kommunikationsprotokoll und zum Bedrohungsmodell passen.

- Verschlüsselung schützt Vertraulichkeit, sofern Shlüssiel und Verfahren korrekt verwaltet werden.
- Authentisierung prüft Identitäten; Integritätsschutz hilft, unbemerkte Änderungen zu erkennen.
- Schlüssel sicher erzeugen, speichern, verteilen, erneuern und bei Bedarf widerrufen.
- Kryptografie verhindert weder Funkstörung noch kompromittierte Endgeräte.

SICHERHEITSPRINZIP

Keine eigene Kryptografie entwickeln. Bewährte Verfahren und peprüfte Implementierungen entspreccheend den geltenden Systemanforderungen einsetzen.

4 · GPS-SPOOFING-RESILIENZ



Spoofing beeeichnet das Vortäuschen gefälschter GNSS-Signale, um einen Empfänger zu falschen Positions- oder Zeitangaben zu verleiten.

- Nicht jede GNSS-Störung ist Spoofing: Jamming unterdrückt oder stört Signale.
- Plausibilitätsprüfungen mit unabhängigen Sensoren oder Quellen können Abweichungen aufdecken.
- Signalqualität, Zeitverhalten und Positionssprünge überwachen.
- Bei widersprüchlichen Messungen Unsicherheit anzeigen und sichere Betriebsverfahren nutzen.

KEINE EINZELMAßNAHME GENÜGT

GNSS-Resilienz entsteht durch abgestimmte Erkennung, alternative Datenquellen und betriebliche Reaktion; eine einzelne Prüfung beweist nicht sicher die Echtheit eines Signals.

SICHERHEITSGRENZE

Weltraumsysteme umfassen mehr als den Satelliten: Kommunikationsstrecken, Bodenstationen, Steuerzentren, Lieferkette und Nutzerterminals gehören zum Schutzbereich.

6 · BODENSEGMENT HÄRTEN



Bodenstationen und Kontrollsysteme sind IT-Systeme mit besonders schutzbedürftigen Steuerungsfunktionen.

- Netze segmentieren: Missionssteuerung nicht unnötig direkt dem Internet aussetzen.
- Mehrfaktor-Authentisierung und rollenbasierte, minimale Berechtigungen einsetzen.
- Systeme und Anwendungen aktuell halten; sichere Konfigurationen und Endpunktschutz nutzen.
- Fernzugriffe absichern, überwachen und auf notwendige Quellen begrenzen.

5 · COTS UND LIEFERKETTE



COTS-Komponenten sind kommerziell verfügbare Produkte, die in Raumfahrtsystemen eingesetzt werden können.

- Komponenten, Versionen, Herkunft und Abhängigkeiten dokumentieren.
- Risiken durch unsichere Standardkonfigurationen und nicht benötigte Funktionen reduizeren.
- Updates und Schwachstellenbehandlung an lange Einsatz- und Wartungszyklen anpassen.
- Zuliefererzugänge, Build-Prozesse und Software-bestandteile in die Risikobetrachtung aufnehmen.

148 · DECEPTION TECHNOLOGY

Kontrollierte Lockmittel machen verdächtige Interaktionen sichtbar.

1. ZWECK UND PRINZIP



- Ein Täuschungsziel soll Angreifer anziehen oder ihre Aktivitäten erkennbar machen.
- Normale Nutzer sollten es im Regelfall nicht benötigen.
- Erkennt Interaktionen, die bei produktiven Ressourcen ungewöhnlich wären.
- Liefert Kontext für Untersuchung und Priorisierung.
- Kann Angreifer von echten Anlagen ablenken, garantiert dies aber nicht.

2. HONEYPOT



- Ein Honeypot ist ein einzelnes absichtlich exponiertes System oder ein Dienst, der Angriffe beobachtbar macht.
- Kann etwa einen scheinbaren Webdienst, Server oder Login-Endpunkt nachbilden.
- Geringe legitime Nutzung erleichtert die Bewertung eines Treffers.
- Täuschungsgrad und Interaktionsumfang bestimmen, welche Aktivitäten sichtbar werden.
- Ein Honeypot ist kein Ersatz für Patchen, Zugriffsschutz oder Backups.

3. HONEYNET



- Ein Honeynet verbindet mehrere Honeypots zu einer kontrollierten Täuschungsumgebung. Es kann Beziehungen zwischen Systemen und Aktivitäten sichtbar machen.
- Segmentierung begrenzt Verbindungen zu Produktionsnetzen.
- Kontrollierte Egress-Regeln verhindern, dass kompromittierte Systeme Dritte angreifen.
- Protokollierung erfasst relevante Netzwerk- und Systemereignisse.
- Mehr Komponenten bedeuten mehr Aufwand für Betrieb und Überwachung.

4. CANARY TOKENS



- Ein Canary Token ist ein gezielt platzierter Köder, dessen unerwartete Nutzung einen Hinweis auslöst. Beispiele sind scheinbare Zugangsdaten, Dokumente oder URLs.
- Ein Alarm kann auf Zugriff, Öffnen oder Auflösen eines Tokens hinweisen.
- Token-Typ und Auslösebedingungen bestimmen die Aussagekraft.
- Ein Treffer ist ein Signal zur Prüfung, kein automatischer Beweis für einen erfolgreichen Einbruch.
- Keine echten Geheimnisse oder personenbezogenen Daten als Köder verwenden.

5. PLATZIERUNG UND ISOLATION

- Platzierung richtet sich nach dem Bedrohungsmodell und den zu schützenden Ressourcen. Täuschungsressourcen müssen klar kontrolliert und von produktiven Systemen abgegrenzt sein.
- Nur erforderliche Dienste und Ports exponieren.
- Netzwerksegment und ausgehenden Verkehr streng begrenzen.
- Administrationszugänge separat schützen.
- Verantwortliche, Zweck und Eskalationsweg festlegen.



6. TELEMETRIE UND ALARMQUALITÄT

- Ein guter Alarm ist zeitnah, nachvollziehbar und mit genügend Kontext versehen, um eine sichere Untersuchung zu ermöglichen.
- Zeitstempel, Ziel, Ereignistyp und relevante Quellinformationen erfassen.
- Uhren synchronisieren und Protokolle gegen unbefugte Änderung schützen.
- Alarme an zentrale Überwachung oder ein SIEM weitereieten.
- Fehlalarme und technische Ausfälle berücksichtigen; Signale nicht ungeprüft gleichsetzen.



7. REAKTION UND GRENZEN

- Bei einem Treffer beginnt eine strukturierte Prüfung. Das Signal kann auf Fehlkonfiguration, interne Tests oder einen Angriff hindeuten.
- Alarm verifizieren und betroffene Täuschungsressource identifizieren.
- Ereignisse sichern und mit anderen Telemetriedaten korrelieren.
- Nach festgelegtem Incident-Response-Prozess eskalieren.
- Täuschung kann erkannt, umgangen oder selbst als Angriffsfläche missbraucht werden.



Merke
Täuschung verbessert Sichtbarkeit, nicht automatisch Sicherheit. Isolation, Zugriffskontrollen und ein klarer Reaktionsprozess bleiben wesentlich.

8. KURZÜBERBLICK



- Honeypots täuschen einzelee Dienste oder Systeme vor, Honeynets verknüpfen mehrere Täuschungsziele und Canary Tokens melden die Nutzung gezielt platzierter Köder.
- Ziel: verdächtige Interaktionen sichtbar machen und untersuchen.
- Voraussetzungen: Autorisierung, Isolation, begrenzter Egress und geschützte Protokolle.
- Ergebnis richtig deuten: Ein Alarm ist ein Untersuchungsanlass, keine Gewissheit.

Nur autorisiert und isoliert einsetzen.

149 · DISASTER RECOVERY & BUSINESS CONTINUITY

Einordnung und Planung für die Wiederherstellung von IT-Diensten und die Fortführung kritischer Geschäftsprozesse. ISO 22301 beschreibt Anforderungen an ein Business-Continuity-Managementsystem; konkrete Wiederanlaufziele legt die Organisation anhand ihrer Auswirkungen und Risiken fest.



- Ziel: nach Störungen priorisiert handlungsfähig bleiben.
- DR (Disaster Recovery) fokussiert die Wiederherstellung von IT und Daten.
- BCM (Business Continuity Management) umfasst die Fortführung wichtiger Prozesse.
- Die acht Knoten zeigen zentrale Bausteine und ihre Beziehungen.

BEGRIFFE UNTERSCHEIDEN

RPO und RTO sind Planungsziele, keine Garantie für eine störungsfreie Wiederherstellung.

1 · AUSWIRKUNGEN UND PRIORITÄTEN

Die Planung beginnt mit der Frage, welche Unterbrechungen welche Folgen haben. Eine Business-Impact-Analyse (BIA) hilft, wichtige Aktivitäten, Abhängigkeiten und zeitliche Dringlichkeit nachvollziehbar zu ordnen.



- Erfasse kritische Prozesse und benötigte IT-Dienste.
- Notiere Abhängigkeiten: Personal, Standorte, Daten, Anbieter und Netze.
- Vergleiche Folgen eines Ausfalls über die Zeit.
- Leite Wiederherstellungsprioritäten aus den Folgen ab.

2 · RPO UND RTO

RPO (Recovery Point Objective) beschreibt den maximal tolerierten Datenverlust in Zeit, also wie weit zurück Daten nach einer Wiederherstellung höchstens liegen sollen. RTO (Recovery Time Objective) bezeichnet die angestrebte maximale Zeit bis zur Wiederaufnahme eines Dienstes oder Prozesses.



- RPO 4 Stunden: Wiederherstellungspunkt soll höchstens vier Stunden vor dem Störungszeitpunkt liegen.
- RTO 8 Stunden: Wiederaufnahme wird innerhalb von acht Stunden angestrebt.
- Niedrigere Zielwerte können zusätzliche Technik, Tests und Kosten erfordern.
- Ziele müssen zu Prozessbedarf und Wiederherstellungsverfahren passen.

NICHT VERWECHSELN

RPO betrifft den Datenstand; RTO betrifft die Wiederanlaufzeit. Beide Werte werden pro Dienst oder Prozess festgelegt.

3 · SICHERUNG UND DATENWIEDERHERSTELLUNG

Backups sind nur dann hilfreich, wenn sie verfügbar, intakt und wiederherstellbar sind. Sicherungsstrategie und Wiederherstellungstests sollten sich an RPO, Datenmenge, Abhängigkeiten und Bedrohungen orientieren.



- Sicherungen regelmäßig erstellen und den Erfolg überwachen.
- Kopien gegen unbeabsichtigte Änderung und unbefugten Zugriff schützen.
- Aufbewahrung und Wiederherstellungsreihenfolge dokumentieren.
- Restore-Tests durchführen und Ergebnisse festhalten.

WICHTIG

Ein erfolgreicher Backup-Job beweist allein nicht, dass Anwendungen und Daten im Notfall wiederhergestellt werden können.

4 · FAILOVER UND WIEDERANLAUF

Failover schaltet einen Dienst bei Ausfall auf eine alternative Komponente oder Umgebung um. Automatisches Failover kann Ausfallzeiten reduzieren, benötigt aber geeignete Erkennung, getestete Abläufe und klare Zuständigkeiten.



- Hot Standby: Ersatzsystem läuft und kann schnell übernehmen.
- Warm Standby: Ersatzumgebung ist vorbereitet, benötigt aber meist weitere Aktivierung oder Synchronisierung.
- Cold Standby: Ressourcen werden erst bei Bedarf bereitgestellt; der Start dauert typischerweise länger.
- Nach dem Failover Datenkonsistenz, Funktion und Rückschaltplan Risiken wie Anbieter- oder Netzwerkausfällen.

ABHÄNGIGKEITEN BEACHTEN

Failover ist keine vollständige Wiederherstellung, wenn etwa Identitätsdienste, DNS, Netzwerk oder Datenbanken fehlen.

5 · DR-SITES

Eine Disaster-Recovery-Site ist ein alternativer Standort oder eine alternative Umgebung, die zur Wiederherstellung von IT-Diensten genutzt werden kann. Ihre Eignung hängt von Ausstattung, Datenreplikation, Erreichbarkeit und Abhängigkeiten ab.



- Hot Site: weitgehend betriebsbereit und auf schnelle Übernahme ausgelegt.
- Warm Site: teilweise ausgestattet; Aktivierung und Ergänzungen sind erforderlich.
- Cold Site: Standort und Grundinfrastruktur, aber wenig unmittelbar nutzbare IT-Ausstattung.
- Standorttrennung allein schützt nicht vor gemeinsamen Risiken wie Anbieter- oder Netzwerkausfällen.

6 · NOTFALLPLÄNE FÜR IT-AUSFÄLLE

Ein Notfallplan macht Rollen, Entscheidungen und Abläufe im Ereignisfall auffindbar. Er sollte auch Szenarien berücksichtigen, in denen zentrale IT, Kommunikation oder Identitätsdienste nicht verfügbar sind.



- Auslösekriterien und Eskalationswege festlegen.
- Rollen, Stellvertretungen und Kontaktwege offline verfügbar halten.
- Manuelle Ersatzverfahren für priorisierte Prozesse beschreiben.
- Wiederherstellungsschritte, Reihenfolge und Rückkehr zum Normalbetrieb dokumentieren.

BEI TOTALEM IT-AUSFALL

Papier- oder anderweitig offline verfügbare Kontaktlisten und Ablaufpläne können helfen, wenn elektronische Dokumentation nicht erreichbar ist.

7 · KOMMUNIKATION UND ÜBUNG

Geübte Abläufe verringern Unsicherheit und machen Lücken sichtbar. Übungen sollten zum Risiko und zur Größe der Organisation passen und Erkenntnisse in die Planung zurückführen.



- Interne und externe Ansprechpartner samt Ersatzkontakt festlegen.
- Übungen durchführen: etwa Planspiel, Wiederherstellungstest oder Umschaltung.
- Zeitbedarf und beobachtete Hindernisse dokumentieren.
- Maßnahmen mit Verantwortlichen und Termin nachverfolgen.

8 · PRÜFEN UND VERBESSERN

Business Continuity ist kein einmaliges Dokument. Veränderungen an Systemen, Prozessen, Standorten und Dienstleistern können Wiederherstellungsannahmen ungültig machen; regelmäßige Überprüfung hält den Plan nutzbar.



- RPO- und RTO-Ziele gegen Testergebnisse und Prozessbedarf abgleichen.
- Pläne bei wesentlichen Änderungen aktualisieren.
- Backup-, Failover- und Kommunikationsverfahren erneut testen.
- Erkenntnisse in Zuständigkeiten, Ressourcen und Maßnahmen übersetzen.

MERKSATZ

Prioritäten bestimmen, Ziele festlegen, Wiederherstellung testen und den Plan anhand der Ergebnisse verbesssern.

149 · DISASTER RECOVERY & BUSINESS CONTINUITY

IT-Dienste wiederherstellen - kritische Prozesse fortführen

CYBER WARFARE, GEOPOLITIK UND STAATLICHE APT-AKTEURE

Abschließende Mindmap: Wie staatlich gelenkte Cyber-Operationen in geopolitische Konflikte eingebettet sind – und wie Attribution, Schutz kritischer Infrastrukturen und Resilienz zusammenhängen.

- Cyber-Operationen können Spionage, Einflussnahme oder Störung unterstützen; ihre Wirkung hängt vom politischen und technischen Kontext ab.
- Technische Indikatoren allein beweisen weder die Identität noch die staatliche Steuerung eines Akteurs.

EINORDNUNG

Diese Übersicht vermittelt Grundlagen. Sie ist keine Anleitung zu Angriffen und ersetzt keine Lagebeurteilung durch zuständige Stellen.

1. BEGRIFF UND ZIELE



- Cyber Warfare: politisch oder militärisch eingebetteter Einsatz digitaler Mittel; der Begriff ist nicht einheitlich definiert.
- Ziele können Informationsgewinn, Einfluss, Störung oder Unterstützung anderer Operationen sein.
- Cyber-Aktivität ist nicht automatisch ein bewaffneter Angriff.

2. HYBRIDE KRIEGSFÜHRUNG



- Kombination militärischer und nichtmilitärischer Mittel, etwa Cyber-Aktivitäten, Desinformation und wirtschaftlichem Druck.
- Digitale Operationen können andere Maßnahmen vorbereiten, begleiten oder verstärken.
- Wirkung entsteht häufig durch Zusammenspiel und zeitliche Abstimmung.

3. STAATLICHE AKTEURE UND APT



- APT bezeichnet eine anhaltende, zielgerichtete Bedrohung; es ist kein Beweis für staatliche Urheberschaft.
- Staatliche Stellen, beauftragte Gruppen und kriminelle Akteure können sich überschneidende Werkzeuge nutzen.
- Motive umfassen Spionage, strategischen Vorteil und Störung.

4. ATTRIBUTION: HINWEISE STATT GEWISSHEIT

- Untersucht werden etwa Infrastruktur, Malware, Arbeitsweise, Zeitmuster und Opferauswahl.
- Indikatoren können kopiert, manipuliert oder über Dritte genutzt werden.
- Eine belastbare Bewertung verbindet technische, operative und geopolitische Informationen und benennt Unsicherheit.



5. KRITISCHE INFRASTRUKTUREN

- KRITis umfasst lebenswichtige Dienste wie Energie, Wasser, Gesundheit, Kommunikation und Verkehr.
- IT- und Betriebstechnik (OT) können gekoppelt sein: Auswirkungen reichen daher über Datenverlust hinaus.
- Abhängigkeiten von Lieferanten, Netzen und Personal können Störungen verstärken.



6. MÖGLICHE FOLGEN

- Vertraulichkeitsverlust: sensible Informationen werden offengelegt.
- Integritätsverlust: Daten oder Steuerungsinformationen werden verändert.
- Verfügbarkeitsverlust: Dienste oder Anlagen sind zeitweise nicht nutzbar.
- Physische und gesellschaftliche Folgen hängen von System, Schutzmaßnahmen und Einsatzbedingungen ab.



7. RESILIENZ UND REAKTION

- Vorbereiten: kritische Anlagen und Abhängigkeiten kennen, Zugänge begrenzen, sichere Backups vorhalten.
- Erkennen: Protokolle, ungewöhnliche Aktivitäten und Meldungen zusammen betrachten.
- Reagieren: Zuständigkeiten, Kommunikationswege und Wiederanlaufpläne üben.
- Wiederherstellen: Systeme kontrolliert zurückführen und Erkenntnisse in Schutzmaßnahmen übernehmen.



8. VERANTWORTUNGSVOLLE EINORDNUNG



- Fakten, Hypothesen und offene Fragen getrennt dokumentieren.
- Politische Zuschreibungen nicht aus einem einzelnen technischen Indikator ableiten.
- Bei konkreten Vorfällen zuständige Sicherheits- und Behördenstellen einbeziehen.
- Kernaussage: Abschreckung und Schutz profitieren von Kooperation, Vorsorge und verlässlicher Wiederherstellung.