Zusammenfassung CySec

1 CONTENTS

2 Concepts 6
2.1 CIA Triade + Non-repudiation, Accountability 6
2.2 Assets 6
2.2.1 Schutz von Intellectual Property 6
2.3 Data Classification 7
2.4 Threat Model 8
2.5 STRIDE 8
2.6 Vulnerabilities, CVE, CWE 8
2.7 Risk Management 9
2.7.1 Exposure Factor 10
2.7.2 Single Loss Expectancy 10
2.7.3 Annualized Rate of Accurance 10
2.7.4 Annualized Loss Expectancy 10
2.7.5 Cost/benefit Analysis 10
2.7.6 Residual Risk 10
2.7.7 Mitigation, Assignment, Acceptance, Deterrence, Avoidance, Rejection 10
2.8 Privacy 11
3 Identity and Access Management 12
3.1 Begriffe 12
3.2 Arten 12
3.3 Kategorien 12
3.4 Ablauf Access Control 12
3.4.1 Identification 12
3.4.2 Authentication 13
3.4.3 Authorization 14
3.4.4 Auditing 14
3.4.5 Accounting 14
3.5 Angriffe 15
3.6 Massnahmen: 15
4 Linux 16
4.1 Red Team vs. Blue Team 16

4.2 SELinux 16
4.3 Sticky Bit, SUID, GUID 16
4.4 Shells 17
4.5 Diverses 17
5 Symmetrische Verschlüsselung 18
5.1 Stream Cipher 21
5.2 Block Cipher 21
5.2.1 Operation Modes von Block Ciphers 22
5.3 Diffie-Hellman 23
5.3.1 Elliptic Curve 24
6 Asymmetric Encryption 25
6.1 Rivest-Shamir-Adleman (RSA) 25
6.1.1 Public und private Key bestimmen 25
6.1.2 Verschlüsselung durch Absender 25
6.1.3 Entschlüsselung durch Empfänger 25
6.1.4 Signatur durch Absender 25
6.1.5 Prüfung der Signatur durch Empfänger 25
6.2 Digital Signature Algorithm (DSA) 26
6.3 Hashing 26
6.3.1 Message Authentication Codes 26
7 Open Web Application Security Project (OWASP) Top 10 27
7.1 Cyber Attack Kill Chain 27
7.1.1 Improved Version 28
7.2 Sichere Anwendungen 28
7.3 Web App Architektur 29
7.3.1 Simpel 29
7.3.2 Mit WAF 29
7.4 Top 10 29
7.5 Attack Types & Common Attacks 30
7.5.1 SQL Injection 30
7.5.2 Cross Site Scripting (XSS) 30
7.5.3 Cross Site Request Forgery (XSRF) 31
7.5.4 Advanced Persistent Threat (APT) 32
7.6 Schutzmechanismen 33
7.6.1 Endpoint detection and response (EDR) 33
7.6.2 Security Information & Event Management (SIEM) 33

7.6.3 Security Orchestration & Automated Response (SOAR) ... 34
7.7 Unterstützung ... 34
8 Ethical Hacking, Pentesting ... 35
8.1 Penetration Testing ... 35
8.2 Hacking vs. Ethical Hacking ... 35
8.3 Penetration Testing ... 35
8.4 Arten von Malware ... 37
8.4.1 Zero-Day Angriff ... 37
8.4.2 Script Kiddie Angriff ... 37
8.4.3 Drive-By Download ... 37
8.4.4 Virus ... 37
8.4.5 Logic Bombs ... 38
8.4.6 Trojan Horses ... 38
8.4.7 Keystroke logging ... 38
8.4.8 Ransomware ... 38
8.4.9 Worms ... 38
8.4.10 Spyware ... 38
8.4.11 Adware ... 38
8.5 Antivirus ... 38
8.5.1 Detection ... 38
8.6 Application Attacks ... 39
8.6.1 Buffer Overflows ... 39
8.6.2 Time of Check to Time of Use ... 39
8.6.3 Backdoor ... 39
8.6.4 Escalation of Privilege / Rootkits ... 39
8.7 Network Security ... 39
8.7.1 Denial of Service (DoS) ... 39
8.7.2 Distributed Denial of Service (DDoS) ... 40
8.7.3 Man-in-the middle ... 41
8.7.4 Man-in-the browser ... 41
8.7.5 Eavesdropping ... 41
8.7.6 Impersonation/Masquerading ... 41
8.7.7 Spoofing ... 41
8.7.8 Replay Attack ... 41
8.7.9 Modification Attack ... 41
8.7.10 Session Hijacking ... 41

9 Complete Cryptographic Systems 42
9.1 Authenticated Encryption 42
9.2 Authenticated Encryption with Associated Data (AEAD) 42
10 Transport Layer Security (TLS) 43
10.1 Versionen 43
10.1.1 TLS 1.3 43
10.2 Headers 43
10.3 Content Types 44
10.4 Handshake 44
10.4.1 TLS 1.2 44
10.4.2 1-RTT Handshake (TLS 1.3) 44
10.4.3 Ohne Perfect Forward Secrecy (unsicher) 45
10.4.4 Mit Perfect Forward Secrecy (sicher) 45
10.4.5 TLS Master Secret und Session Keys aus Premaster Secret berechnen 46
10.4.6 Session Resumption 46
10.5 Cipher Suites 47
10.5.1 TLS 1.3 47
10.6 Heartbeat Protocol 47
11 Public Key Infrastructure 48
11.1 X.509 48
11.1.1 Extensions 49
11.1.2 Revocation 49
11.2 Verifizierungsmöglichkeiten 49
11.3 Trust Service Provider (TSP) 49
11.3.1 Zertifikatsausstellung 50
11.3.2 Chain of Trust 50
11.4 Certificate Pinning 51
12 E-Mail Security 52
12.1 S/MIME 52
12.2 Spam 53
12.2.1 Sender Policy Framework (SPF) 53
12.2.2 Domain Keys Identified Mail (DKIM) 53
12.2.3 Domain-based Message Authentication, Reporting and Conformance (DMARC) 53
13 Authentication & Federation 54
13.1 Authentication Protocols 54
13.1.1 Phasen 54

13.1.2 Authentication Schemes 54

2 CONCEPTS

2.1 CIA TRIADE + NON-REPUDIATION, ACCOUNTABILITY

- **Confidentiality**: kein unautorisierter Zugriff auf Daten at rest, in transit oder in use, z.B. mit Verschlüsselung und Access control
- **Integrity**: Daten sind korrekt und vollständig und wurden nicht unautorisiert verändert, z.B. mit IDS und Hashes
- **Availability**: Daten sind in nützlicher Frist einsehbar, z.B. mit Redundanz, Skalierbarkeit, Backup
- **Non-Repudiation**: Subject einer Aktion kann die Aktion nicht abstreiten, z.B. mit digitalen Zertifikaten, Session-IDs oder Transaction Logs
- **Accountability**: verantwortlich sein für Aktionen und Resultate

2.2 ASSETS

Ein Asset ist alles in einer Umgebung, das geschützt werden sollte. Ein Verlust resultiert in einem Sicherheitsvorfall, Produktivitätsverlust, Profitverlust, zusätzlichen Ausgaben oder sogar Betriebsschliessung.

Es gibt verschieden Arten von Assets:

- **Information**: alle Daten
- **Systeme**: IT-Systeme die Dienste bereitstellen
- **Geräte**: alle mögliche Hardware wie Server, PCs, Laptops, Drucker, usw.
- **Facilities**: physische Gebäude die gekauft oder gemietet sind
- **Personal**: Angestellte
- **Intellectual Property (IP)**: Markennamen, Rezepte, Produktionstechniken, usw.

2.2.1 Schutz von Intellectual Property

- **Copyright**: ermöglicht dem Urheber das Duplizieren seines Werks zu verbieten, tritt automatisch in Kraft und gilt 70 Jahre, geeignet für Literatur, Choreografien, grafische Arbeit, audiovisuelle Inhalte, Musik, Architektur, bei Software kann der Source-Code damit geschützt werden, nicht aber die Idee oder der Prozess hinter der Software
- **Trademark**: Schutz von Wörtern, Slogans oder Logos einer Firma, tritt automatisch in Kraft und kann mit dem TM-Symbol gekennzeichnet werden, nach der Registrierung kann es mit ® gekennzeichnet werden
- **Patent**: schützt IP von Erfindern für 20 Jahre, Erfindung wird öffentlich und muss neu, nützlich und nicht offensichtlich sein, nicht geeignet für Software
- **Trade Secret**: wird von grossen Firmen wie Microsoft benutzt, schützt IP, ohne die Erfindung öffentlich zu machen und auf unbestimmte Zeit

2.3 DATA CLASSIFICATION

Wird benutzt, um zu bestimmen wie viel Aufwand, Geld und Ressourcen benutzt werden sollen, um das Asset zu schützen

Data States:

- **At Rest**: gespeicherte Daten auf Disks, USBs, SANs oder Backup Tapes
- **In Transit/In Motion**: Daten die über ein Netzwerk oder das Internet übertragen werden
- **In Use**: eine Applikation kann verschlüsselte Daten nicht verwenden, sie werden in den Memory oder temporäre Buffers decrypted

Sensitive Daten:

- **Personally Identifiable Information (PII)**: Informationen, die benutzt werden können, um die Identität einer Person festzustellen, z.B. Name, Sozialversicherungsnummer, Geburtsdatum und Geburtsort, Name der Mutter, biometrische Einträge, medizinsche, finanzielle, Ausbildungs- und Anstellungs-Angaben, bei Breaches müssen betroffene Personen informiert werden gesetzlich
- **Protected Health Information (PHI)**: Gesundheitliche Informationen, welche von einer Organisation erstellt oder erhalten wurden, egal ob vergangen, aktuell oder zukünftig, reguliert durch **Health Insurance Portability and Accountability Act (HIPAA)** in den USA, durch Gesetzgebung elektronisches Patientendossiert (EPDG) in der Schweiz
- **Proprietäre Daten**: Daten, die einer Organisation einen Wettbewerbsvorteil verschaffen, wie Source Code, technische Pläne, interne Prozesse, IP, Trade Secret. Kriminelle beachten Copyrights und Patente nicht, daher müssen sie speziell geschützt werden

Regierung/Militär Klassifizierung:

Top secret	Drastischer Schaden der nationalen Sicherheit
Secret	Kritischer Schaden der nationalen Sicherheit
Confidential	Ernster Schaden der nationalen Sicherheit
Sensitive but unclassified	Interne Verwendung
Unclassified	Keine Auswirkungen

Kommerzielle Klassifizierung:

Confidential/Private	Drastische Effekte auf Wettbewerbsvorteil
Sensitive	Klassifizierter als Public
Public	Passt nicht in die anderen beiden Kategorien

Zerstörung von Daten

- **Erasing**: Nur der Link zu den Daten wird gelöscht, mit Undelete können die Daten wiederhergestellt werden
- **Clearing/Overwriting**: Das Medium wird komplett überschrieben mit einem Charakter, dann dem Komplement davon und dann mit Random Bits, kann mit normalen Tools nicht rückgängig gemacht werden
- **Purging**: mehrere Durchgänge von Clearing
- **Degaussing**: Magnetisches Feld löscht Daten komplett, funktioniert nicht bei CDs, DVDs und SSDs
- **Destruction**: Medium wird unreparierbar zerstört, so dass die Daten nicht mehr extrahiert werden können

Verfolung/Verstecken von sensitiven Daten:

- **Steganografie**: Nachrichten in Files verstecken
- **Watermarking**: Verstecktes Merkmal in einem Bild/Papier oder digitaler Marker versteckt im File

2.4 Threat Model

- Threat: Potentielle Gefahr für ein Asset, absichtlich oder versehentlich
- Threat actor: jemand der einen Threat absichtlich ausnutzt, z.B. Script kiddies, organisierte Hackergruppen, State-sponsored Hacker, Hacktivisten, Terroristen
- Threat intelligence: Wissen über einen existierenden oder aufkommenden Threat
- Threat event: absichtliche oder versehentliche Ausnutzung eines Threats

2.5 STRIDE

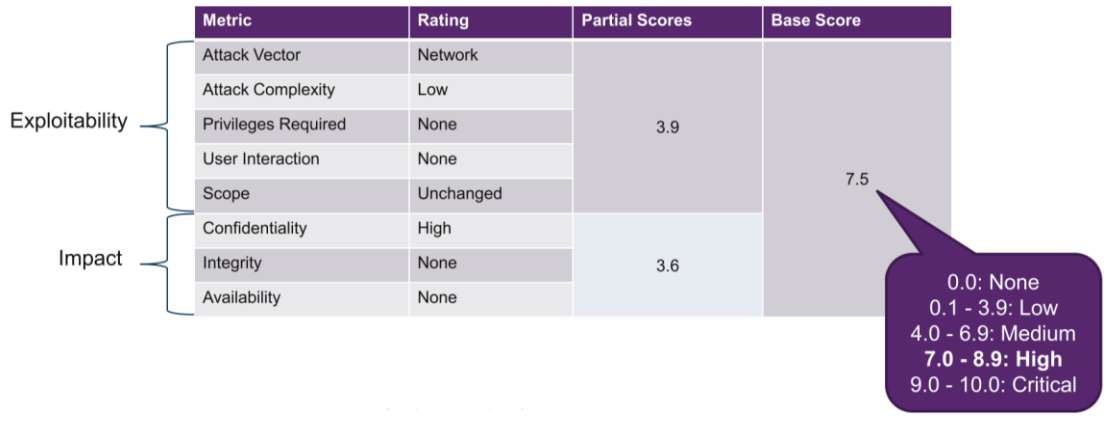
- Spoofing != Authenticity: Zugriff mit einer falschen Identität erhalten
- Tampering != Integrity: Unauthorisierte Manipulation von Daten
- Repudiation != Non-Repudiation: Abstreiten einer Aktion
- Information disclosure != Confidentiality: Unauthorisierter Zugriff
- Denial of Service (DoS) != Availability: Kein oder langsamer authorisierter Zugriff auf Ressourcen
- Elevation of Privilege != Authorization: Beschränkter Account erhält mehr Rechte

2.6 Vulnerabilities, CVE, CWE

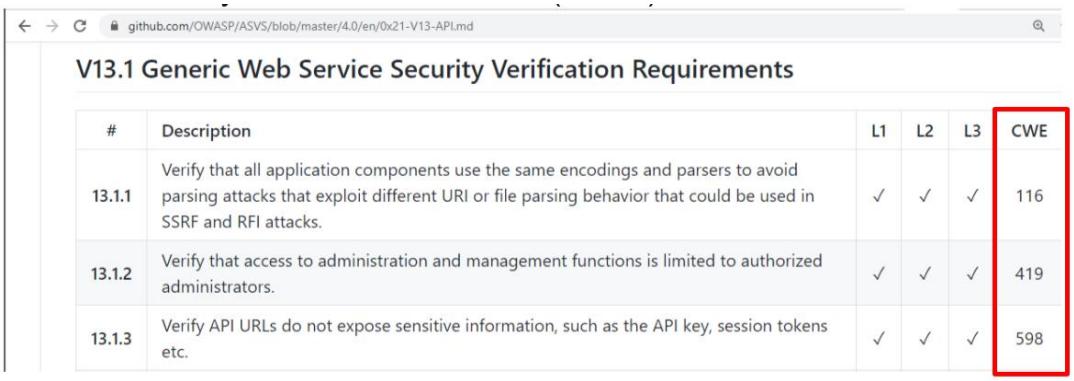
Eine Vulnerability ist eine Schwäche eines Assets, deren Ausnutzung Verlust oder Schaden des Assets zur Folge hat.

Common Vulnerabilities and Exposures (CVE): Industrie-weite ID-Nummer für Vulnerabilities, die Nummer kann von der CVE Numbering Authority (CNA) bestellt werden, oder von MITRE direkt, welche auch CVE Requests für Open Source Software akzeptiert

Common Vulnerability Scoring System (CVSS): einer Vulnerability wird ein Severity Score von 1 – 10 zugewiesen, welcher aufgrund von CIA-Prinzipien berechnet wird



Common Weakness Enumeration (CWE): Von der Community entwickelte Liste von Software- und Hardware-Schwächen für gemeinsame Bezeichnungen, Messlatten für Security Tools und eine Baseline für die Identifikation, Behebung und Verhinderung von Schwächen



github.com/OWASP/ASVS/blob/master/4.0/en/0x21-V13-API.md

V13.1 Generic Web Service Security Verification Requirements

#	Description	L1	L2	L3	CWE
13.1.1	Verify that all application components use the same encodings and parsers to avoid parsing attacks that exploit different URI or file parsing behavior that could be used in SSRF and RFI attacks.	✓	✓	✓	116
13.1.2	Verify that access to administration and management functions is limited to authorized administrators.	✓	✓	✓	419
13.1.3	Verify API URLs do not expose sensitive information, such as the API key, session tokens etc.	✓	✓	✓	598

Exploit: Software oder Command-Abfolge zur Ausnutzung einer Vulnerability

2.7 Risk Management

Das Ziel vom Risk Management ist, IT Security Strategien zu entwickeln und zu implementieren. Die Strategien sollen das Risiko auf ein für die Mission des Unternehmens akzeptables Level zu bringen. Oft können Risiken mit wenig Aufwand stark reduziert werden, eine risikofreie Umgebung ist jedoch unmöglich.

Begriffe:

- **Risk Management**: Faktoren identifizieren, die Daten beschädigen oder offenlegen könnten, die Faktoren bewerten nach Daten-Value und Massnahmen-Kosten, Kosteneffektive Massnahmen implementieren, um die Risiken zu minimieren
- **Countermeasure**: Risiko minimieren durch Aktionen oder Produkte
- **Risk Analysis**: alle Assets der Organisation nach Value evaluieren, die Umgebung auf Risiken untersuchen, jedes Threat Event auf Wahrscheinlichkeit und Schaden evaluieren, die Kosten für Massnahmen dem Management präsentieren
- **Asset Valuation**: Dem Asset einen Geldbetrag für Entwicklung, Betrieb, Administration, Werbung, Support, Reparaturen und Ersatz zuweisen
- **Exposure**: die Möglichkeit für einen Asset Verlust aufgrund eines Threats, quantitative Risk Analysis zur Berechnung des Exposure Factors (EF) durchführen
- **Risk**: die Möglichkeit, dass etwas passieren könnte, das Daten beschädigen, zerstören und offenlegen könnte, je grösser die Wahrscheinlichkeit für ein Threat Event ist desto grösser ist das Risk
- **Realized Risk**: ein Risiko das erkannt wurde
- **Attack**: Exploitation einer Vulnerability durch einen Threat agent
- **Breach**: Umgehung eines Sicherheits-Mechanismus durch einen Threat Agent
- **Penetration**: ein Threat Agent hat Zugriff auf die Infrastruktur durch Umgehung von Kontrollmechanismen

- **Risk Assessment**: Es gibt zwei Methode:
    - **Quantitativ**: Dem Asset-Verlust einen konkreten Geldbetrag zuweisen
      Assign Asset Value (AV), Calculate Exposure Factore (EF), Calculate Single Loss Expectancy (SLE), Assess the annualized rate of occurrence (ARO), Derive the annualized Loss expectancy (ALE), Kosten/Nutzen Analyse von Massnahmen machen
    - **Qualitativ**: Dem Asset-Verlust subjektive und immaterielle Werte zuweisen

2.7.1 Exposure Factor
Der EF ist der Prozentsatz des Werts des Assets, wenn das Asset durch ein identifiziertes Risiko teilweise oder ganz zerstört wird.

2.7.2 Single Loss Expectancy
Der Geldbetrag der verloren geht, wenn ein Asset durch ein identifiziertes Risiko teilweise oder ganz zerstört wird.

SLE = asset value (AV) * exposure factor (EF)

2.7.3 Annualized Rate of Accurance
Die erwarteten Anzahl Male, wo ein spezifischer Threat oder Risiko pro Jahr auftritt.

2.7.4 Annualized Loss Expectancy
Die möglichen jährlichen Kosten aller Instanzen spezifischer erkannter Threats gegen ein spezifisches Asset.

ALE = single loss expectancy (SLE) * annualized rate of occurrence (ARO)

2.7.5 Cost/benefit Analysis
Formel: ALE vor der Sicherheitsmassnahme – ALE nach der Sicherheitsmassnahme – jährliche Kosten der Massnahme

Wenn das Resultat negativ ist, sollte die Massnahme aus finanzieller Sicht nicht implementiert werden, wenn es positiv sollte die Massnahme unbedingt implementiert werden

2.7.6 Residual Risk
Das Residual Risk ist das Restrisiko, das nach der Implementation von Massnahmen übrigbleibt.

2.7.7 Mitigation, Assignment, Acceptance, Deterrence, Avoidance, Rejection
- **Mitigation**: Implementierung von Safeguards und Countermeasures, um Vulnerabilities und Threats zu eliminieren.
- **Assignment**: Ein Risiko einer anderen Entität oder Organisation zuweisen, beispielsweise durch eine Versicherung.
- **Acceptance**: Die Akzeptanz eines Risikos, weil die Kosten der Massnahme die des ALE übersteigen.
- **Deterrence**: Abschreckungsmassnahmen für potenzielle Angreifer implementieren, z.B. Audits, Kameras, Sicherheitspersonal oder Warnschilder
- **Avoidance**: Ein Risiko vermeiden, indem ein anderes Produkt verwendet wird oder es einfach deaktiviert wird.
- **Rejection**: nicht akzeptabel, Risiko ignorieren

2.8 PRIVACY

Privacy ist das Recht einer Person die persönlichen Daten zu kontrollieren.

- Datensammlung sollte beschränkt werden.
- Datenbesitzer sollten Privacy-Prinzipien respektieren und durchsetzen.
- Datenprozesse sollten Privacy und Integrity erzwingen.
- Techniken zur sicheren permanenten Datenlöschung sollten verwendet werden.

Nach 9/11 gab der PATRIOT Act in den USA der Polizei und Geheimdienst weitgehende Berechtigungen zur Überwachung von elektronischer Kommunikation. Personen durften abgehört werden und ISPs mussten grosse Datenmengen sammeln und der Polizei zur Verfügung stellen.

In der EU gibt es seit 2018 die General Data Protection Regulation (GDPR), welche allen EU Bürgern das Recht auf Privatsphäre, das «right to be forgotten» und Einsicht in ihre gesammelten Daten gibt und Datensammler zur Meldung von Breaches verpflichtet.

Pseudonymization: Datenelemente werden mit Pseudonymen ersetzt, somit kann kein Rückschluss auf die Person hinter den Daten gemacht werden, es kann aber rückgängig gemacht werden

Anonymization: Wenn die persönlichen Daten nicht gebraucht werden, können sie anonymisiert werden.

3 IDENTITY AND ACCESS MANAGEMENT

Der Zugriff auf Assets muss kontrolliert werden.

3.1 BEGRIFFE

- **Subject**: Aktive Entität die auf ein Objekt zugreift, z.B. ein User, Programm, Prozess, Service oder Computer
- **Objekt**: passive Entität die einem Subjekt Informationen zur Verfügung stellt, z.B. ein File, DB, Computer, Programm, Prozess, Service, Drucker oder Storage

3.2 ARTEN

- **Preventive**: Probiert das Vorkommen von unautorisierter Aktivität zu stoppen
  Beispiele: Zaun, Schloss, Data Classification, Verschlüsselung, Firewall, IPS
- **Detective**: Probiert unautorisierte Aktivität zu erkennen, nachdem sie passiert ist
  Beispiele: Sicherheitspersonal, Überwachungskameras, Audit Trails, IDS
- **Corrective**: Probiert die Umgebung zu ändern, damit eine erkannte unautorisierte Aktivität nicht mehr passieren kann
  Beispiele: Reboot, Antivirus der Virus in Quarantäne steckt, Backup-Restore
- **Deterrent:** Ermutigt potenzielle Täter, einen Angriff nicht zu tun
  Beispiele: Warnschilder, Security-Trainings, Kameras
- **Compensating:** Massnahme wird installiert, um andere Massnahmen zu unterstützen oder durchzusetzen
  Beispiel: Policy besagt, dass alle PII verschlüsselt sein müssen, jedoch werden sie im Klartext übers Netzwerk übertragen, eine Compensating control wird implementiert, damit sie bei der Übertragung auch geschützt sind
- **Directive/Authoritative:** Soll die Subjekte ermutigen, die Sicherheitsrichtlinien einzuhalten.
  Beispiele: Monitoring, Prozesse, Überwachung durch Vorgesetzten
- **Recovery:** Corrective controls mit mehr Fähigkeiten
  Beispiele: Cluster, RAID, Backup-Restore, Shadowing, Multisite

3.3 KATEGORIEN

- Physikalisch (Hardware)
- Technisch/Logisch (Software)
- Administrativ (Policies und Prozesse)

3.4 ABLAUF ACCESS CONTROL

3.4.1 Identification

Dabei beansprucht das Subjekt eine Identität. Alle Subjekte müssen eine unique Identity haben, welche public ist. Die IT-Systeme arbeiten ausschliesslich mit Identities. Dies kann beispielsweise die Eingabe eines Usernames, eine Smartcard, ein Token-Gerät oder der Fingerabdruck sein.

3.4.2 Authentication

Das Subjekt muss beweisen, dass ihm die Identität gehört. Meist passiert das durch ein Passwort. Die Authentication Daten sind private und müssen geschützt werden.

Kategorien und Beispiele (Type 1 am schwächsten, Type 3 am stärksten):

- Type 1: Something you know
  - **Passworte** sind eine schwache Form der Authentication, sie sind oft einfach und schnell geknackt, werden aufgeschrieben oder geteilt. Passworte sollen Klein-/Gross-Buchstaben, Zahlen und Sonderzeichen enthalten. Sie sollen keine persönlichen Informationen und Wörter ausdem Wörterbuch enthalten.
  - **Passwordphrases** sind eine gute Möglichkeit für starke Passwörter, dabei wird ein Satz geschrieben und einige Zeichen durch Sonderzeichen ersetzt.
  - **Kognitive Passwörter**: Dem User werden Fragen gestellt werden mit Antworten, die nur das Subjekt wissen sollte, beispielsweise «Wie hiess dein erster Chef?»
  - **PIN**
- Type 2: Something you have
  - **Smartcard**: Badge in Kreditkartengrösse mit Microchip und einem oder mehreren Zertifikaten für asymmetrische Verschlüsselung, sind tamper-resistant und sind eine einfache Option für User, um komplexe Verschlüsselungskeys zu transportieren
  - **Token**: Passwort-generierendes Gerät, Server weiss, was das Gerät gerade für einen Token anzeigt, es gibt zwei Arten:
    - **Synchronous (Time-based One Time Password TOTP)**: Alle 30s gibt es einen neuen Token, bedingt genaue Zeit auf Gerät und Server
    - **Asynchronous (HMAC-based One Time Password HOTP)**: Dynamische Token werden generiert basierend auf einem Algorithmus und Counter, der Token bleibt der gleiche, bis er benutzt wird
  - Memory Card
  - USB drive
- Type 3: Something you are/do
  - Fingerprint
  - Stimme
  - Retina
  - Iris
  - Gesicht
  - Hand
  - Handvenen
  - Puls
  - Keyboard Pattern

Für noch mehr Sicherheit wird Multifaktor mit 2 oder mehr Types eingesetzt.

Viele IAMs setzen auch Geolocation Techniken mit der IP ein, um zu prüfen, ob das Login von einem realistischen Ort kommt.

3.4.3 Authorization

Dabei wird geprüft, ob die Identität berechtigt ist für die angeforderte Aktion.

Arten:

- **Discretionary Access Control (DAC)**: Jedes Object hat einen Owner und der Owner kann anderen Subjects Zugriff gewähren oder verweigern, z.B. NTFS
- **Role Based Access Control (RBAC)**: Berechtigungen werden an Rollen anstatt Subjects zugewiesen. Das Subject bekommt dann eine oder mehrere Rollen zugewiesen, von welchem es die Berechtigungen erbt, z.B. Kubernetes
- **Rule Based Access Control**: Globale Rules werden allen Subjects zugewiesen, z.B. Firewall.
- **Attribute Based Access Control**: Berechtigungen werden an Attribute anstatt Subjects zugewiesen. Das Subject bekommt dann eine oder mehrere Attribute zugewiesen, von welchem es die Berechtigungen erbt
- **Mandatory Access Control**: Subjects und Objects bekommen Labels und wenn beide das Label haben, wird der Zugriff gewährt

Begriffe und Prinzipien:

- **Implicit deny**: Grundsätzlich gibt es keinen Zugriff, ausser man ist berechtigt, wird von den meisten Mechanismen verwendet.
- **Constrained interface**: Teile des UIs werden versteckt, wenn der User nicht berechtigt ist.
- **Access Control Matrix**: Tabelle mit Subjects, Objects und zugewiesenen Berechtigungen, System checkt diese Matrix vor der Aktion
- **Capability Table**: Subject-focused Tabelle mit einer Liste der Objects bei jedem Subject
- **Content-dependent control**: Bei einer DB kann man beispielsweise nur bestimmte Daten ausgeben, es gibt eine View für intern und eine für den Kunden mit weniger Infos.
- **Context-dependent control**: Spezifische Aktvität vorher wird benötigt bevor Zugriff gewährt wird. Beispielsweise kann man in einem Software-Onlineshop erst herunterladen nachdem man bezahlt hat.
- **Need to know**: Es gibt nur Zugriff auf die Informationen, die man für seine Job Funktion benötigt.
- **Least privilege**: Es gibt nur die Privilegien, die man für seine Job Funktion benötigt.
- **Separation of Duties and Responsibilities**: Dabei wird sichergestellt, damit nicht eine einzelne Person eine kritische Funktion benutzen kann. Beispielsweise muss der CFO eine Überweisung über einem gewissen Betrag absegnen.

3.4.4 Auditing

Die Aktionen eines Subjects aufzeichnen.

3.4.5 Accounting

Die Fähigkeit, die Person hinter einem Subject verantwortlich zu machen.

3.5 ANGRIFFE

- **Access Aggregation Attacks (passive attack)**: Nicht-sensitive Informationen sammeln und sie zusammenfügen, um sensitive Informationen zu erlernen.
- **Password Attacks (Brute-force attack)**: Online accounts angreifen oder DBs stehlen
- **Dictionary Attacks (Brute-force attack)**: Ebenfalls Online accounts angreifen oder DBs stehlen, jedoch wird für das Cracken ein Dictionary mit Wörtern und bekannten Passwörtern verwendet
- **Birthday Attack (Brute-force attack)**: möchte Kollisionen finden aufgrund des Birthday paradox, kann mit sicheren Hash-Methoden verhindert werden
- **Rainbow Table Attacks**: es wird eine vorgefertigte Liste von Passworten und Hashes genommen fürs Brute-Force, das ist viel schneller, kann aber mit Salted Hashes verhindert werden
- **Sniffer Attack (Eavesdrop)**: Die Pakete im Netzwerk werden aufgezeichnet und analysiert nach Passworten, kann verhindert werden mit Verschlüsselung, OTPs und physikalischen Schutzmassnahmen
- **Spoofing**: man gibt eine falsche Identität (IP, Mail, Tel-Nr.) vor
- **Social Engineering**: Ein Angreifen möchte das Vertrauen eines Users gewinnen, damit er ihm sensitive Informationen sendet
- **Shoulder surfing**: Sensitive Informationen vom Bildschirm ablesen
- **Phishing**: Form des Social Engineerings, man möchte dass der User ein Anhang öffnet oder auf einen Link klickt, um ihm sensitive Daten zu entlocken oder einen Virus zu installieren
- **Spear Phishing**: gezieltes Phising auf eine spezifische Gruppe von Personen
- **Whaling**: Phishing gegen höhere Manager
- **Vishing**: Phishing über Instant Messaging oder VoIP anstatt Mail

3.6 MASSNAHMEN:

- **Layering**: verschiedene Controls in Serie verwenden, damit mehrere überwunden werden müssen
- **Abstraction**: vereinfacht die Security-Administration, indem man Gruppen von Funktionen oder Typen macht und diese dann zuweist
- **Data Hiding**: absichtlich Daten so platzieren, damit unautorisierte Subjects sie nicht sehen
- **Security through obscurity**: etwas verstecken und das Subject nicht informieren, hoffen dass er es nicht findet
- **Encryption**: Einsicht durch unautorisierte Subjects verhindern

4 LINUX

4.1 RED TEAM VS. BLUE TEAM

- **Red Team**: Offensive Security, Penetration Testing der Systeme, Rolle des Hackers wird angenommen, suchen kreative Lösungen und testen Angriffe aus der Praxis, das Ziel ist der angegriffenen Firma Lücken aufzuzeigen, meist sind das externe Personen, welche die internen Sicherheitsmassnahmen nicht kennen, benötigte Skills:
    - Outside the box thinking (ständig neue Tools und Techniken probieren)
    - Tiefes Wissen über die Systeme (Protokolle, Libraries, Networking, Server, DBs)
    - Softwareentwicklung (eigene Tools entwickeln können)
    - Penetration Testing (Lücken und Bedrohungen finden)
    - Social Engineering
- **Blue Team**: Defensive Security, Netzwerksicherheit wird untersucht, um Lücken zu finden, Wege zur besseren Incident Response werden gesucht, müssen die Taktiken der Angreifer kennen, aktuelle Aktivität wird mit einem IDS überwacht, benötigte Skills:
    - Organisiert und Detail-orientiert (Mindset, um Lücken nicht zu vergessen)
    - Security Analyse und Threat Profil (Security Assessment, Risk Profil, Threat Profil)
    - Hardening (technische Hardening Techniken kennen um die Angriffsfläche zu reduzieren)
    - IDS-Knowhow (Software kennen die Netzwerktraffic überwacht)
    - Security Information and Event Management (SIEM, Echtzeitanalyse von Events)

4.2 SELINUX

Linux verwendet standardmässig Discretionary Access Control (DAC) für Files, SELinux wurde von der NSA entwickelt, um es mit Mandatory Access Control (MAC) zu erweitern.

Es werden Labels an Prozesse, Files, Sockets usw. zugewiesen

Es können Klassifizierungen erstellt werden, ein tiefer eingestufter Prozess kann dann beispielsweise keine Files einer höheren Klasse nutzen, egal ob die Berechtigungen das zulassen würden.

4.3 STICKY BIT, SUID, GUID

Wenn das Sticky Bit gesetzt ist auf einem Ordner, kann nur der Owner des Files es löschen oder umbenennen, auch wenn die Berechtigungen es zulassen würden.

Bei Files hat es keine Bedeutung.

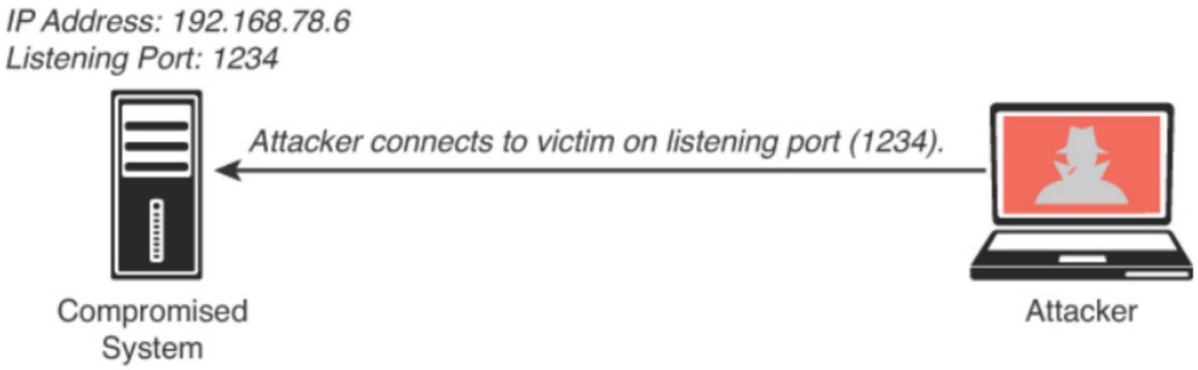
Aktivieren: chmod +t dir/

SUID: File wird immer als Owner ausgeführt (chmod u+s file)

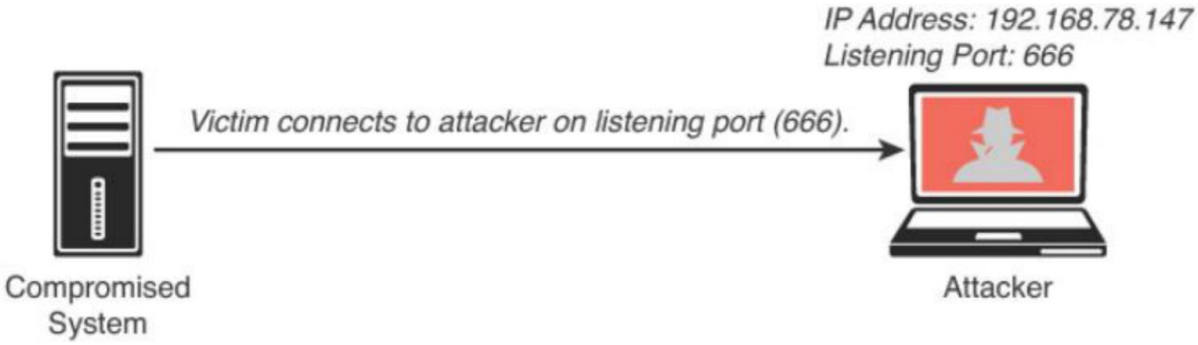
GUID: File wird immer mit der Gruppe ausgeführt (chmod g+s file)

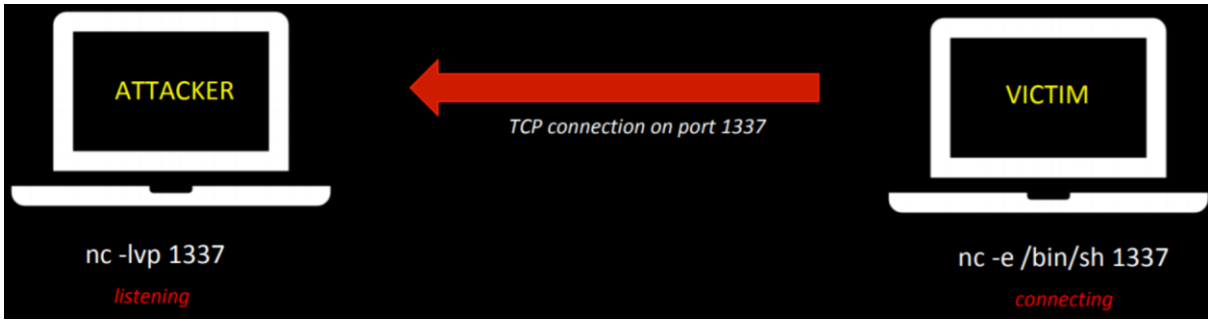
4.4 SHELLS

Bind Shell (normal):



Reverse Shell:





Die Reverse Shell funktioniert auch durch Firewalls, wenn beispielsweise Server ins Internet dürfen.

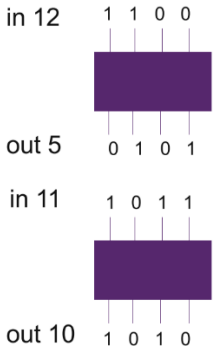
4.5 DIVERSES

- Als Firewall kann iptables verwendet werden. Der Nachfolger davon ist nftables.
- IDS für Linux: Snort
- IPS für Linux: Security Onion oder RedHunt Linux
- Mit Scapy können IP-Pakete gesendet, gesnifft oder manipuliert werden. Es kann alle möglichen Pakete erstellen, somit ist es gut geeignet fürs Testen von Netzwerkattacken und ob IDS/IPS sie erkennen.

5 SYMMETRISCHE VERSCHLÜSSELUNG

Begriffe:

- Message/Plaintext: Daten im Klartext
- Ciphertext: Mithilfe eines kryptografischen Algorithmus verschlüsselte Daten
- Cipher: der kryptografische Algorithmus
- Cryptographic key: der Schlüssel mit dem verschlüsselt wird
- One-Way Function: Mathematische Operation, die aus einem Input einen Output generiert, der nicht zurück zum Input transformiert werden kann
- Reversability: Eigenschaft der Kryptografie die Verschlüsselung rückgängig machen zu können
- Nonce: eine unique public Zahl die bei jeder Verwendung anders sein muss, z.B. Counter
- Initialization Vector (IV): zufällige Bits mit der gleichen Länge wie die Message, es wird dann zusammen XORed damit bei jeder Verschlüsselung ein anderer Ciphertext entsteht
- Confusion: Beziehung zwischen Plaintext und Ciphertext ist so kompliziert, dass ein Angreifer nicht den Plaintext abändern und das Resultat analysieren kann, um den Key rauszufinden, wird durch Substitution hinzugefügt
- Diffusion: wenn eine kleine Änderung am Plaintext viele Änderung im ganzen Ciphertext verteilt bewirkt, wird durch Permutationen hinzugefügt
- Kerckhoff: Ein kryptografisches System sollte sicher sein, auch wenn alles über das System ausser der Key public ist. «The enemy knows the system.» Somit kann jeder das System testen und Schwächen werden schneller aufgedeckt, die allermeisten Kryptografen stimmen zu.
- Substitution: Bytes mit anderen ersetzen

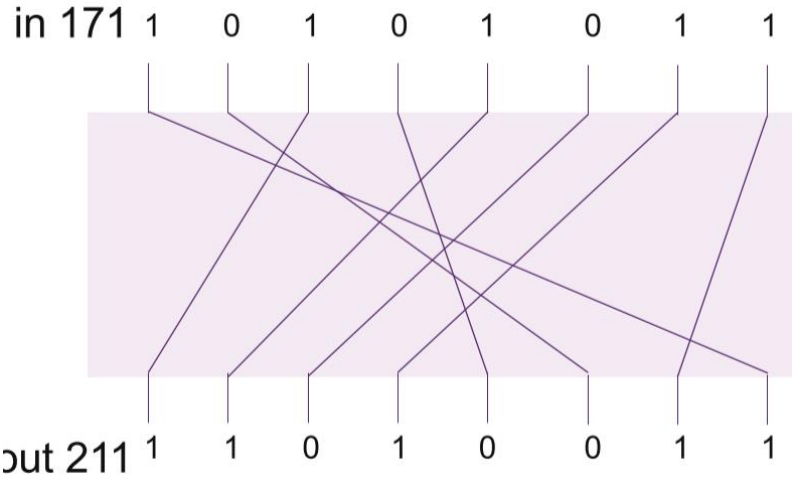


0	1	2	3	4	5
6	13	1	15	7	12

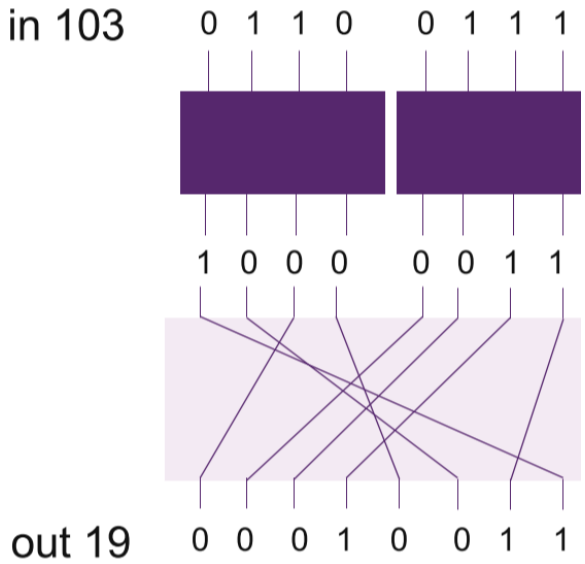
6	7	8	9	10	11
8	3	2	0	14	10

12	13	14	15
5	9	11	4

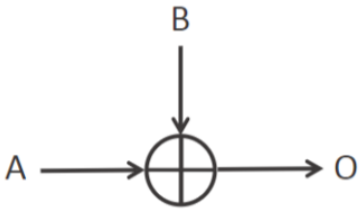
- Permutation: Bytes mit anderen vertauschen



- Substitution-Permutation (SP) Network: Verschiedene Substitutionen und Permutationen werden wiederholt und kombiniert



- XOR ist ein binärer Operator zwischen zwei Values, der sehr nützlich in der Kryptografie ist.



A	B	O
0	0	0
0	1	1
1	0	1
1	1	0

Wenn XOR 2x ausgeführt wird, wird der Effekt rückgängig gemacht (A=Key,B=Message)

$$A \oplus B \oplus A = B$$

- One Time Pad: Wenn man mit XOR eine Message mit einem gleich langen Key verschlüsselt, hat man eine perfekte unknackbare Verschlüsselung, wenn man den Key vernichtet. Es gibt keine statistischen Zusammenhänge und man nennt es Perfect Secrecy. Jedoch ist unpraktisch wegen dem langen Key und die Keys müssen ja sicher transportiert werden können und dürfen nicht 2x benutzt werden.

- Caesar Cipher: Für die Verschlüsselung werden die Buchstaben um eine vorher vereinbarte Anzahl Buchstaben im Alphabet verschoben, z.B. wird A zu D.

Bei der symmetrischen Verschlüsselung wird ein Shared Key verwendet, welcher sowohl vom Sender für die Verschlüsselung als auch vom Empfänger zur Entschlüsselung verwendet wird.

Vorteile:

- Geschwindigkeit: 1000x – 10000x schneller als Asymmetrisch
- Prozessoren haben oft ein AES Instruction Set

Nachteile:

- Man braucht einen Weg das Secret sicher auszutauschen
- Non-Repudiation wird nicht erfüllt, man weiss nicht woher ein Ciphertext kommt
- Integrity wird nicht erfüllt, die Integrity der Message aus dem Ciphertext wird nicht geprüft

Es gibt 2 Arten:

5.1 Stream Cipher

Funktionieren mit Messages von jeder Länge bzw. fortlaufenden Streams.

Vorteile:

- Man ist nicht an eine bestimmte Länge der Message gebunden
- Cipher ist sehr schnell und braucht wenig Memory, läuft somit auch auf schwacher Hardware
- Wenn er gut konzipiert ist, kann er jede beliebige Stelle des Streams aufsuchen.

Nacheile:

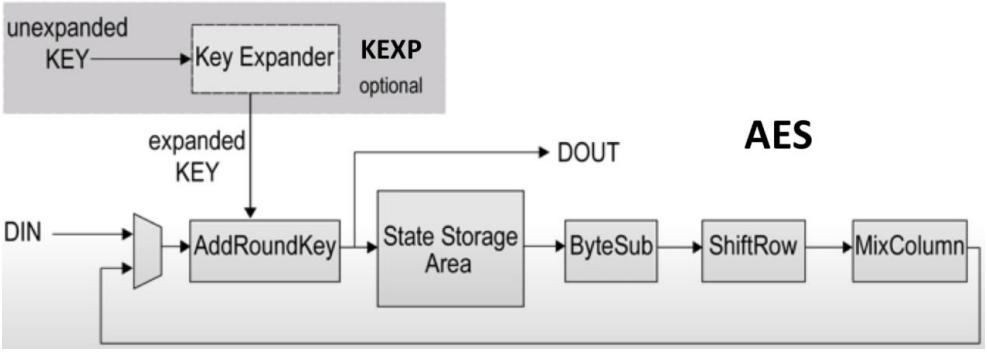
- Keystream muss statistisch zufällig aussehen
- Ein Key und die Nonce darf nie nochmals verwendet werden
- Es gibt keine garantierte Integrität für den Plaintext

5.2 Block Cipher

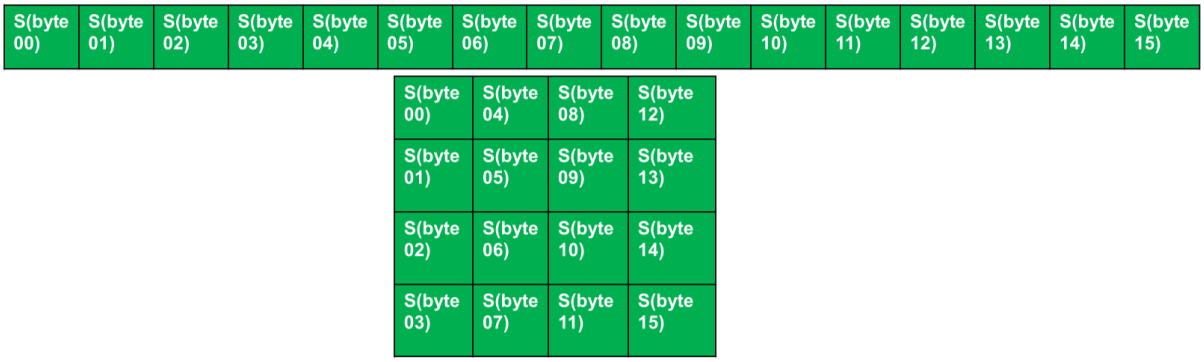
Benötigen einen Input einer bestimmten Grösse und liefern einen Output der gleichen Grösse. Die Verschlüsselung erfolgt durch Confusion und Diffusion in SP-Networks.

Mit Abstand am meisten verwendet wird der Advanced Encryption Standard (AES) es gibt aber auch z.B. noch Feistel Ciphers oder Chacha20, welches oft in Smartphones eingesetzt wird. AES basiert auf dem Rijndael Algorithmus und ist der Nachfolger von DES.

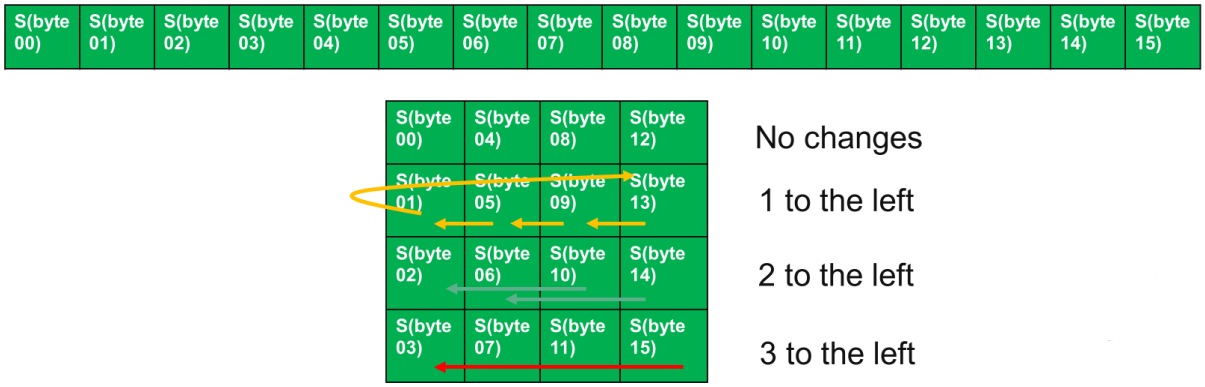
	Key Length (Nk words) (bits)	Block Size (Nb words) (bits)	Number of Rounds (Nr)
AES-128	4 (128)	4 (128)	10
AES-192	6 (192)		12
AES-256	8 (256)		14



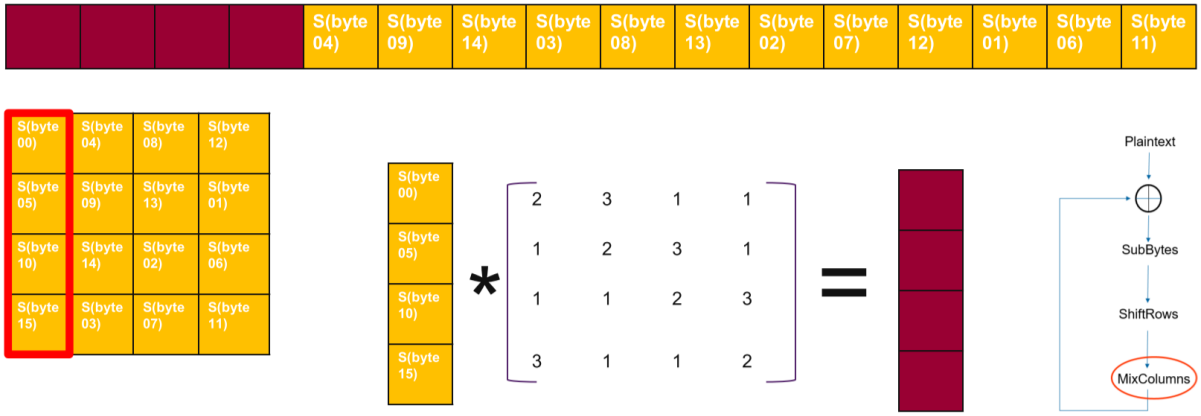
Beim ByteSub werden mithilfe eines Lookup Tables alle Bytes zu einem anderen Byte geändert. Die Bytes müssen nicht am gleichen Ort rauskommen.



Beim ShiftRow werden die Rows rotiert um eine immer 1 grössere Zahl.



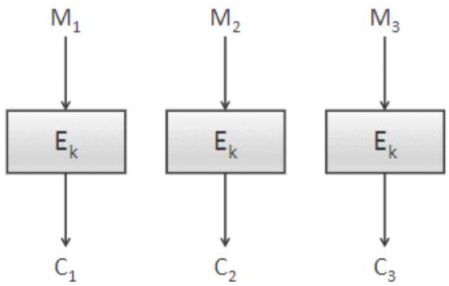
Beim MixColumns werden noch die Kolonnen untereinander durch Matrixmultiplikation verändert.



5.2.1 Operation Modes von Block Ciphers

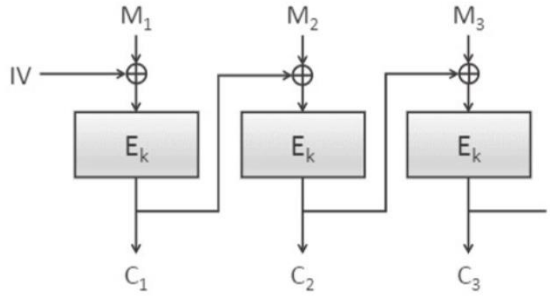
5.2.1.1 Electronic Code Book (ECB)

Ein Block nach dem anderen wird verschlüsselt, daher ist es schwach wenn sich Blocks wiederholen.



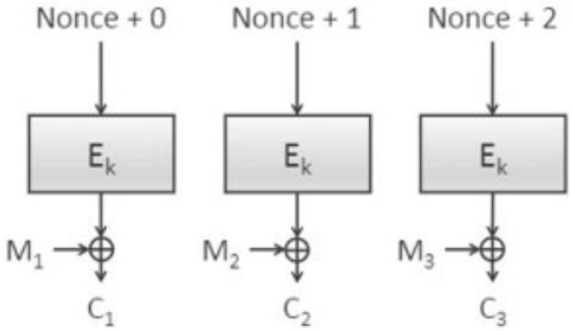
5.2.1.2 Cipher Block Chaining (CBC)

Der Output vom Block wird mit dem Output des vorherigen Blocks XORed, daher ist das nicht parallelisierbar und somit nicht performant. Sicherheitsmässig ist es besser als ECB.



5.2.1.3 Counter Mode (CTR)

Standardmodus von AES. Anstatt der Message wird einen Nonce verschlüsselt, welche dann mit dem Block XORed wird. Das ist parallelisierbar und sicher.



5.3 DIFFIE-HELLMAN

Mit Diffie-Hellman können zwei Parteien über einen unsicheren Kanel einen gemeinsamen Key vereinbaren.

1. Alice und Bob vereinbaren eine 2048/4096 Bit Primzahl p und einen Generator g, welcher ebenfalls eine Primzahl ist und eine primitive Wurzel von p ist

Is 2 a primitive root of prime number 5? 2 mod 5 = 2 2^2 mod 5 = 4 2^3 mod 5 = 3 2^4 mod 5 = 1	Is 2 a primitive root of prime number 7? 2 mod 7 = 2 2^2 mod 7 = 4 2^3 mod 7 = 1 2^4 mod 7 = 2 2^5 mod 7 = 4 2^6 mod 7 = 1
Ja, 1 – 4 ist abgedeckt	Nein, 1 – 6 ist nicht abgedeckt

2. Alice generiert eine zufällige private Zahl a und Bob geneneriert eine zufällige private Zahl b, beide liegen zwischen $1 - p$
3. Alice berechnet $A = g^a \ mod \ p$ und Bob berechnet $B = g^b \ mod \ p$
4. Die beiden Ergebnisse werden ausgetauscht
5. Alice berechnet $B^a \ mod \ p$ und Bob berechnet $A^b \ mod \ p$, beide erhalten dasselbe

Das Ergebnis ist meist das pre-master secret, mit dem dann session keys erstellt werden z.B. mit SHA-256, einer Hashed-key derivation function (HKDF), weil es zu gross (mind. 2048 Bit) ist.

5.3.1 Elliptic Curve

Elliptic Curve Diffie Hellman Encryption (ECDHE) ist ein sichererer Ersatz für die Mathematik unter Diffie Hellman. Es ist eine zweidimensionale Kurve, wobei der private key eine Nummer und der public Key zwei Nummern sind. Das elliptic curve discrete logarithm problem (ECDLP) ist etwas schwieriger zur lösen als das discrete logarithm problem vom normalen Diffie Hellman. Es werden viel kürzere Keys für die gleiche Sicherheit benötigt.

Symmetric	Diffie-Hellman and RSA	Elliptic Curve
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Bei den meisten Protokollen wird für jede Session ein neuer DH Key Exchange gemacht (Perfect Forward Secrecy). Wenn somit ein Key gestohlen wird, können frühere oder spätere aufgezeichnete Messages nicht geknackt werden.

6 ASYMMETRIC ENCRYPTION

Asymmetric Encryption: Separater Public Key zur Verschlüsselung und Private Key zur Entschlüsselung (jeder kann den Public Key abrufen und somit eine Nachricht senden, die nur der bestimmte Empfänger mit dem Private Key entschlüsseln kann)

Signatur: Separater Private Key zur Verschlüsselung und Public Key zur Entschlüsselung (jeder kann prüfen, ob der Absender Zugriff auf den Private Key hat)

6.1 RIVEST-SHAMIR-ADLEMAN (RSA)

Ist ein weit verbreitets System zur sicheren Datenübertragung. Der Vorteil gegenüber der symmetrischen Verschlüsselung ist, dass man nicht den Shared Key kompliziert mit allen teilen muss.

Die Sicherheit von RSA besteht darin, dass es extrem schwer ist, p_1 und p_2 aus n zu bestimmen (time complexity). Es ist somit eine One-way Funktion. RSA ist jedoch schwach bei sehr kurzen messages, daher wird Optimal Assymetric Encryption padding (OAEP) als Standard Padding verwendet, es ist ein Pseudo Random Padding, welches einen IV dem Prozess hinzufügt. RSA ist 1000x langsamer als symmetrische Verschlüsselungsverfahren und wird daher nicht für die Verschlüsselung selbst sondern für den Schlüsselaustausch (früher bei TLS) oder Signatur verwendet.

RSA hat eine message integrity Verifikation. Zuerst wird die message hashed und danach der hash signiert (mit dem private key verschlüsselt). Der Empfänger hasht die message ebenfalls, entschlüsselt die Signatur mit dem public key und vergleicht die Hashes.

Im Rahmen von TLS sendet der Client dem Server eine Challenge die der Server signieren muss. So kann der Client prüfen, ob der Server wirklich der ist den er angibt zu sein, weil er Zugriff auf den Private Key braucht.

6.1.1 Public und private Key bestimmen

1. Zwei beliebige Primzahlen, p_1 und p_2 bestimmen
2. $n = p_1 \cdot p_2$ bestimmen
3. $\varphi(n) = (p_1 - 1) * (p_2 - 1)$ bestimmen ($\varphi(n)$ = Anzahl teilerfremde Zahlen von n in $\mathbb{Z}_n$)
4. Eine beliebige Zahl d bestimmen, $1 < d < \varphi(n)$ & $ggT\big(d, \varphi(n)\big) = 1$
5. Multiplikatives Inverses e von d bestimmen: $d \cdot e \equiv 1 \; mod \; \varphi(n)$
6. Öffentliche Schlüssel e und n veröffentlichen
7. Der private Schlüssel d bleibt geheim

6.1.2 Verschlüsselung durch Absender

1. $(\text{Buchstabe mit Buchstabentabelle in Zahl übersetzt})^e \; mod \; n$

6.1.3 Entschlüsselung durch Empfänger

1. $(Ergebnis \; der \; Verschlüsselung)^d \; mod \; n$

6.1.4 Signatur durch Absender

1. $(Ergebnis \; der \; Verschlüsselung)^d \; mod \; n$

6.1.5 Prüfung der Signatur durch Empfänger

1. $(\text{Buchstabe mit Buchstabentabelle in Zahl übersetzt})^e \; mod \; n$

Ergebnis der Entschlüsselung mithilfe der Buchstabentablle wieder in Text umwandeln

6.2 Digital Signature Algorithm (DSA)

In ein paar Jahren wird RSA zu langsam werden, weil die Keys immer grösser werden müssen. Daher gibt es DSA, welches viel schneller ist. DSA kann aber nicht zur Verschlüsselung, sondern lediglich zum Signieren verwendet werden. Es ist ähnlich wie RSA, jedoch wird die Mathematik von Diffie Hellman (auch mit Elliptic Curves) verwendet.

Es gibt z.B. Ed448 oder Ed25519, welche verschiedene Curves verwenden.

6.3 Hashing

Ein Hash nimmt eine beliebige Eingabe und wandelt sie in einen Pseudo Random Bit String um, der immer die bestimmte Länge vom Hash-Algorithmus hat. Es gibt keine Möglichkeit, aus dem Hash wieder die originalen Daten wiederherzustellen.

Die meisten Hash-Funktionen wandern iterativ von Block zu Block, nach jedem Block ist der Hash wieder anders. Diffusion sollte vorhanden sein, also eine kleine Änderung erzeugt Änderungen über den ganzen Hash hinweg. Sie sollte schnell, aber nicht zu schnell sein, da sonst Brute-Forcing schneller ist.

Ein Hash Algorithmus wird als kaputt angesehen, wenn es möglich ist, mit anderen Inputs den gleichen Hash zu erzeugen.

Algorithmen:

- MD5: broken
- SHA-1: schwach, aber nicht ganz broken
- SHA-2: gleich wie SHA-1 aber längerer Output daher momentan sicher
- SHA-3: nicht besser oder schlechter als SHA-2, anderer Algorithmus
- Password-Based Key Derivation Function 2 (PBKDF2): 5000x SHA-2, daher 5000x langsamer und gut geeignet für Passwörter, Anzahl Iterationen können konfiguriert werden
- Bcrypt: Alternative zu PBKDF2 für Passwörter, basiert auf Blowfish Cipher, ist langsam auf GPUs und kann somit nicht so gut parallelisiert angegriffen werden wie PBKDF2

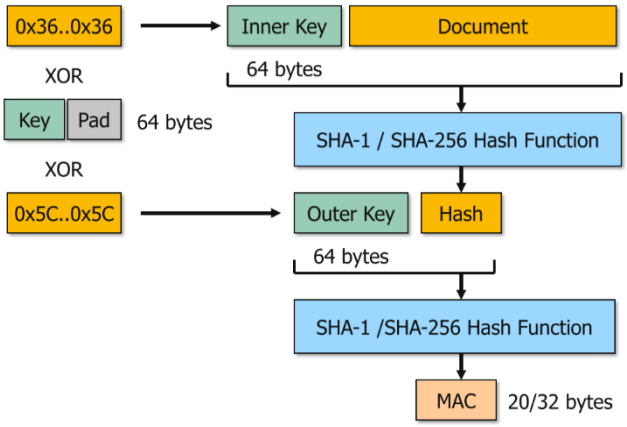
6.3.1 Message Authentication Codes

Symmetrische Verschlüsselung ist anfällig auf Tampering (Integrity-Verletzung). Mit einem Message Authentication Code kann das behoben werden:

1. Der Absender hängt den Key dem Ciphertext an und hasht das Ganze
2. Der Ciphertext und der Hash werden versendet
3. Der Empfänger hangt den Key dem Ciphertext an und hasht das Ganze
4. Nun wird verglichen

Standard MACs haben Sicherheitsprobleme aufgrund der Struktur von Hash-Funktionen wie SHA-256, SHA-1 oder SHA-2 (Length Extension Attack).

Daher gibt es Keyed-Hash MAC (HMAC), dabei wird der Key gesplittet und mit beiden Teilen gehasht, somit ist man nicht mehr gegen einen Length Extension Attack verwundbar.



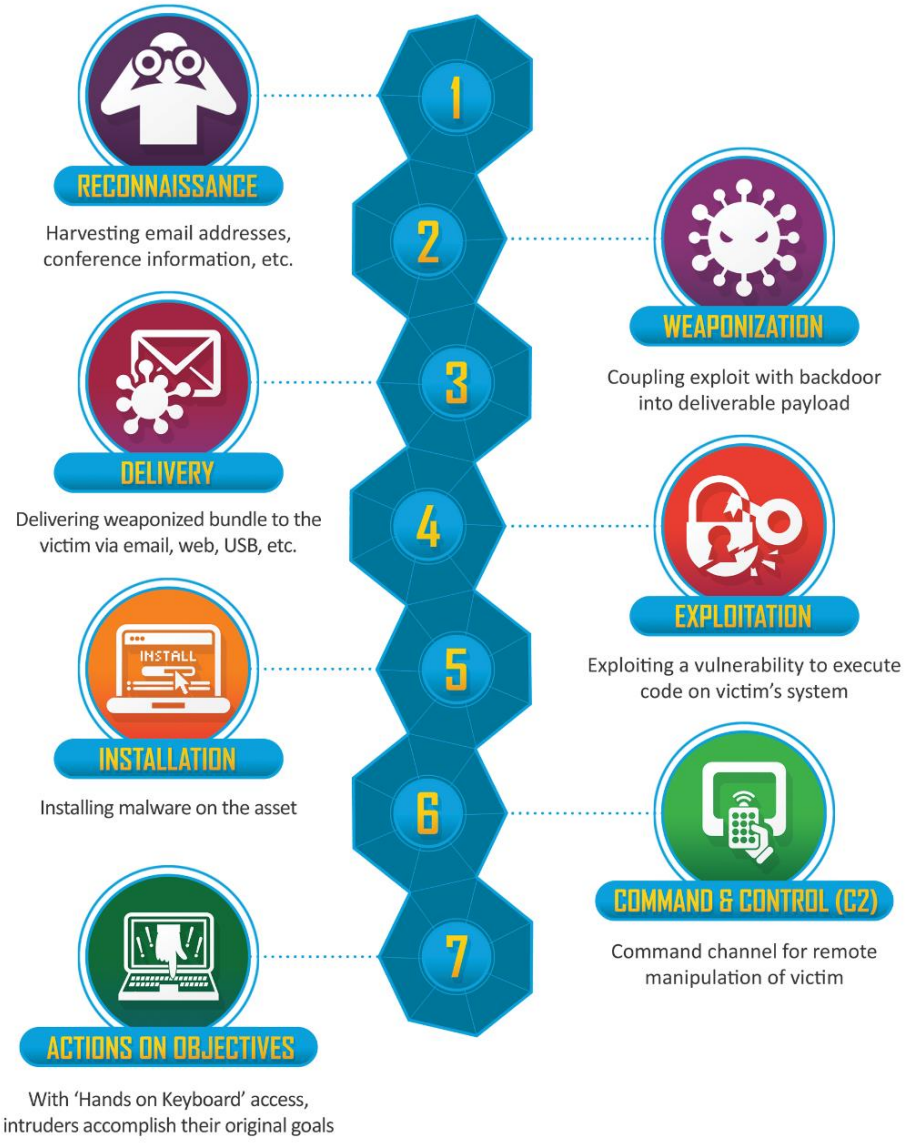
7 Open Web Application Security Project (OWASP) Top 10

Die OWASP Top 10 sollen Awareness schaffen für die am meisten auftretenden Sicherheitslücken in Webapps. Es reicht jedoch nicht für Softwareentwickler und ist nicht einfach testbar.

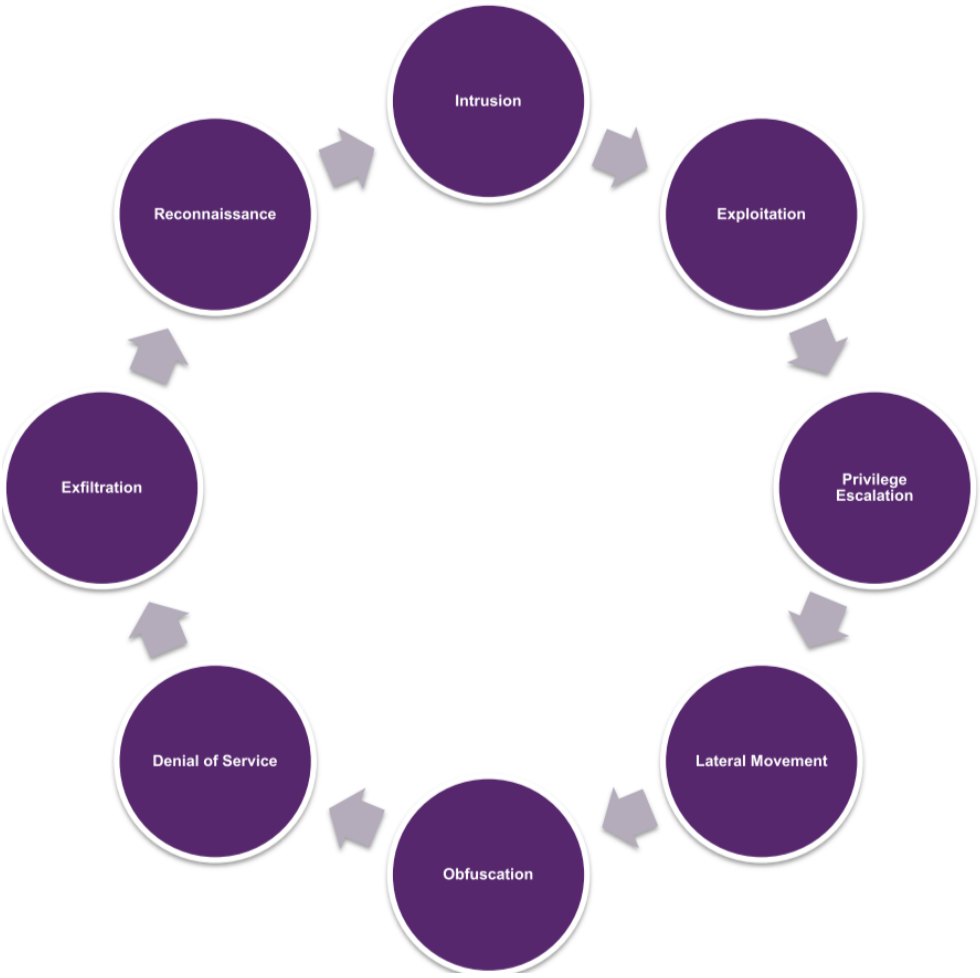
Das OWASP **Application Security Verification Standard (ASVS)** Projekt stellt eine Basis zum Testen von Sicherheitsmassnahmen in Web Applikations und eine Sicherheits-Anforderungsliste zur Verfügung. Es gibt 3 Levels:

- Level 1: 136 Kontrollen, nur die Basics, besser als Top 10, einfach automatisierbar, ergibt keine Secure App
- Level 2: 267 Kontrollen, One-Time Aktivitäten wie Code Revision Control, defect Tracker oder wiederholbares Deployment, meiste Kontrollen können Unit- oder Integration-tested weden
- Level 3: 286 Kontrollen, geeignet für extrem kritische Software, z.B. selbstfahrende Autos, Atomkraftwerke, Finanzwesen

7.1 Cyber Attack Kill Chain



7.1.1 Improved Version

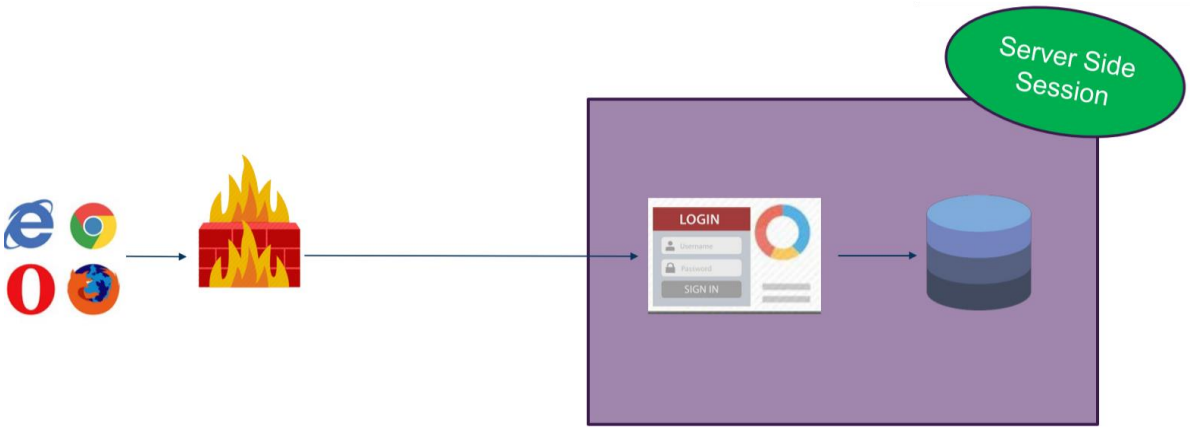


7.2 SICHERE ANWENDUNGEN

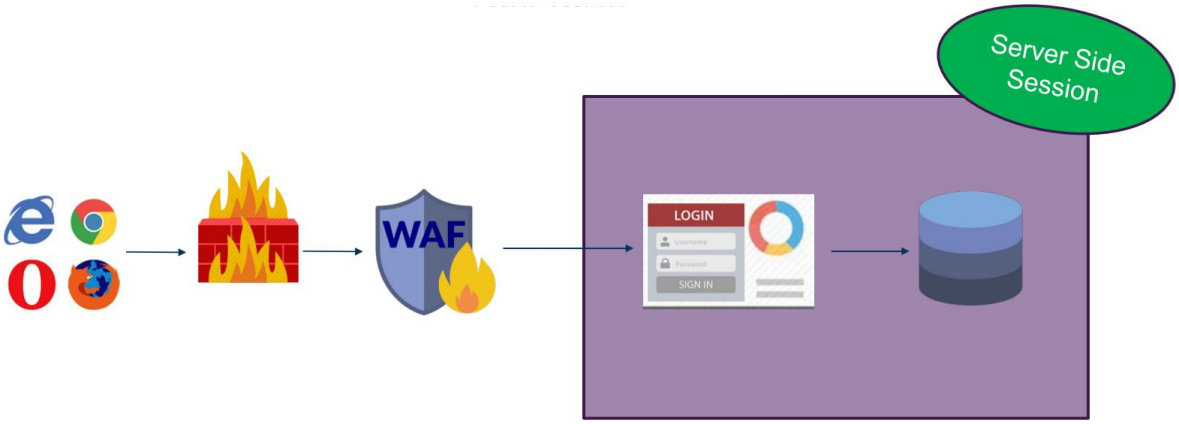
Schwachstelle im Monitoring, Überwachung	Forensic Readiness, Fraud Detection
Schwachstelle in der Anwendung	Sichere Programmierung, Schulung Entwickler
Schwachstellen bei eingesetzten Bibliotheken (Libraries)	Patching, Updating von Libraries, Bibliotheken
Schlecht konfigurierte Anwendung (z.B. SSL/TLS)	Hardening (muss man selbst machen)
Schwachstelle des Application Services (Web, DNS, FTP, SSH)	Patching (Hersteller)
Schwachstelle auf TCP/IP (Netzwerk Ebene)	Firewall & Patching OS (Produkt) durch Hersteller

7.3 Web App Architektur

7.3.1 Simpel

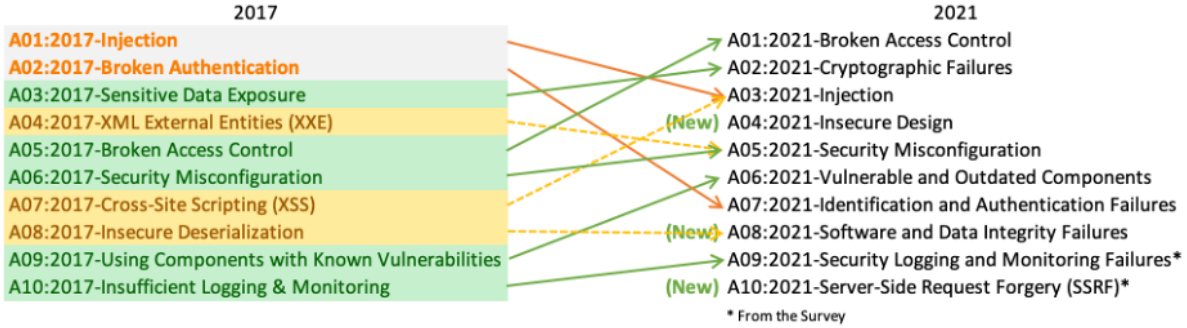


7.3.2 Mit WAF



Beim PCI-DSS Standard für Kreditkartenverabeitende Firmen ist eine WAF obligatorisch. Die WAF filtert sowohl Requests als auch Responses. Schweizer WAF-Brands sind SES, Airlock oder Nevisweb. Die WAF kümmert sich auch um TLS, HSTS, Security Headers, Private Key, Cipher, TLS-Protokolle und Reverse Proxying.

7.4 Top 10



7.5 Attack Types & Common Attacks

7.5.1 SQL Injection

Eine SQL Injection ist möglich bei einem Formular. Dabei gibt der Angreifer einen Teil eines SQL Query Strings ein anstatt einer normalen Eingabe.

Beispiele wird im Hintergrund folgender SQL Query String generiert: SELECT Username FROM Users WHERE Username=' + inputUser + ' AND Password=' + inputPassword + '

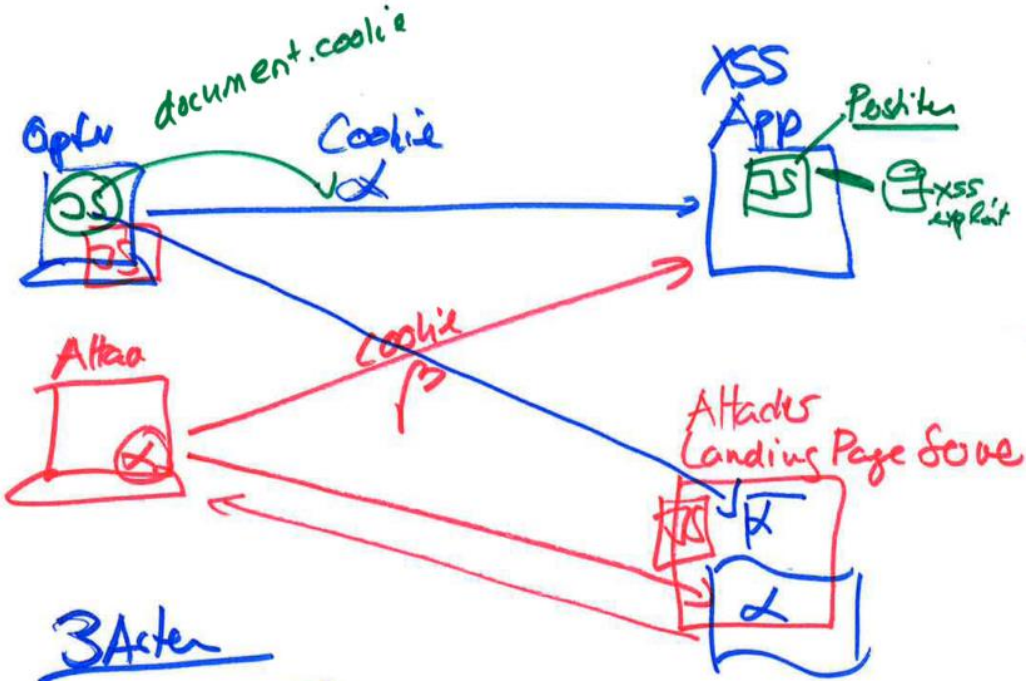
Nun gibt der Angreifer inputUser=test und inputPassword=' OR ''='' ein. Somit ist das Statement true und der User wird eingeloggt ohne das Passwort.

Massnahmen:

- Sichere Programmierung: Prepared Statements in Java, Parameters Collection in .NET, Stored Procedures in SQL
- Least Privileges des DB-Users der Applikation
- Request Filter auf der WAF welche bekannte SQL Injection Muster aussortiert
- Keine SQL-Fehler dem User zurückgeben sonderen anonyme Fehler

7.5.2 Cross Site Scripting (XSS)

XSS ist möglich, wenn die Webseite User-Eingaben mit JavaScript Code akzeptiert, welche reflected werden. Der User befindet sich somit auf der originalen, richtigen Website, jedoch wurde diese durch Schadcode erweitert.



Ziele können sein:

- Session Cookie über die Variable «document.cookie» auf welche JavaScript Zugriff hat an den Angreifer senden, sodass er Session Hijacking machen kann
- Die Logindaten an den Angreifer senden

Es gibt 3 Arten von XSS:

- Stored XSS: Die Eingabe des Angreifers wird dauerhaft auf dem Webserver abgelegt und bei jedem Aufruf ausgeführt.
- Reflected XSS: Die Eingabe des Angreifers wird der URL angehängt (z.B. /search?term=gift)
- DOM-Based XSS: Der Schadcode wird als DOM-Objekt in der Browserumgebung des Clients gespeichert, beim Laden der Website wird der Code lokal ausgeführt.

Gegenmassnahmen:

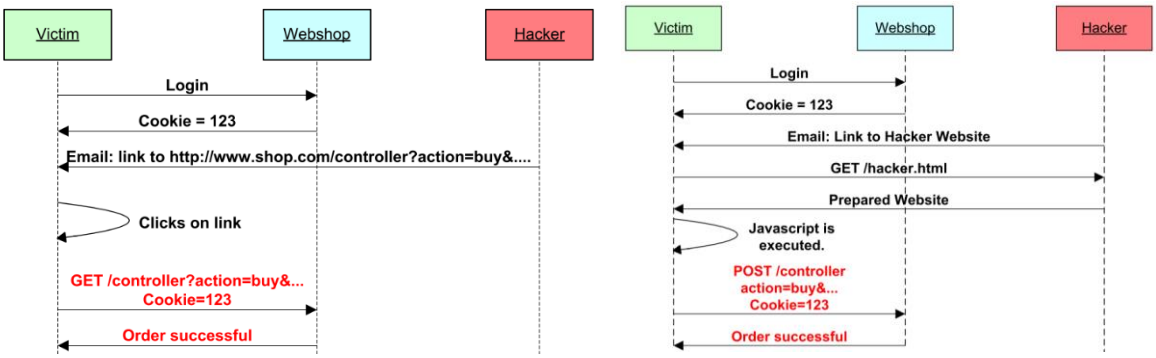
- Die App muss die Eingabe von gefährlichen Zeichen wie <, >, « und ’ in HTML Entities konvertieren &lt, &gt, &quot und &apos.
- Content Security Policy (CSP, script-src self)
- X-XSS-Protection Header (Browser-Feature)
- Cookie mit HttpOnly, Secure, SameSite=Strict (JavaScript kann document.cookie nicht auslesen)
- WAF mit Request Filter gegen gefährliche Zeichen

7.5.3 Cross Site Request Forgery (XSRF)

Wenn ein User sich beim E-Banking einloggt, wird ein Session Token in seinem Browser gespeichert. Beim E-Banking gibt es ein Überweisungsformular, man kann mit einem Inspection Proxy herausfinden, was für ein Request gesendet wird nach dem Klick auf Absenden.

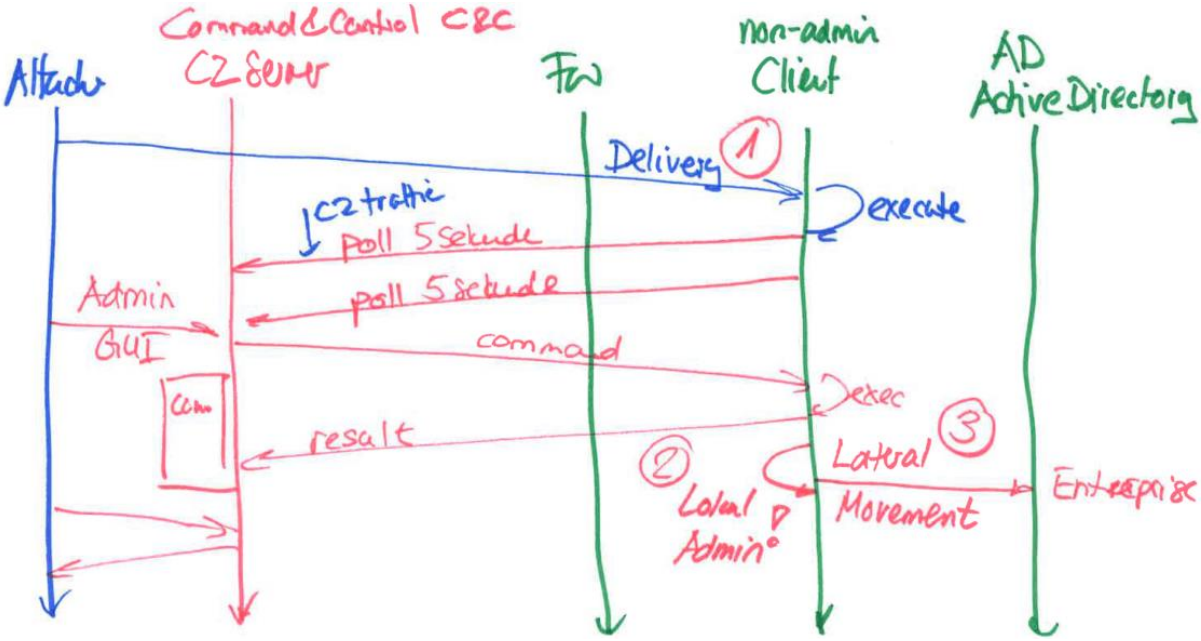
Nun kann ein Hacker dem User ein Mail mit einem Link senden, der einen Request an die Bank sendet. Der gleiche Link könnte auch auf einer bösartigen Website des Hackers sein. Der Request wird angenommen, weil es den Session Token verwendet, der noch im Browser gespeichert ist. Es spielt keine Rolle, ob der Request ein HTTP Get oder Put Request ist, in beiden kann der Session Token hinterlegt werden.

Eigentlich wäre es nicht erlaubt, dass die bösartige Seite auf das Cookie mit dem Session Key zugreifen kann. Der Angreifer kann das umgehen, indem er auf seiner Seite die wirkliche Seite der Bank einbindet.



Als Gegenmassnahme sendet der Server dem Client eine random Nonce, der Client fügt die Nonce dann seinem Request hinzu. Der Angreifer bräuchte auch eine gültige Nonce, die hat er jedoch nicht, weil der Request nicht über das dafür gedachte Formular gesendet wurde.

7.5.4 Advanced Persistent Threat (APT)

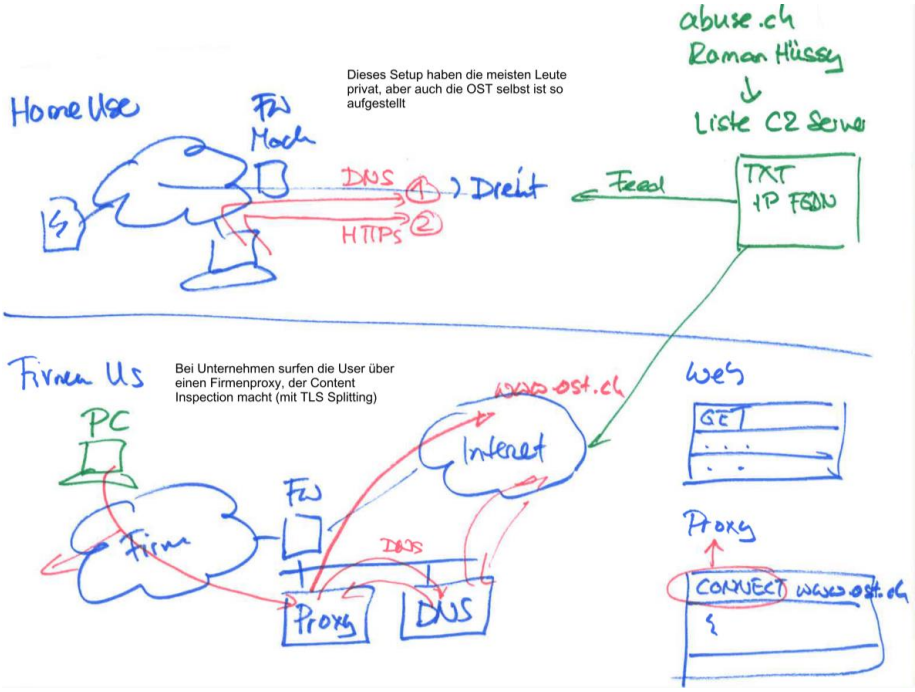


1. Angreifer sendet einem Client einen Virus, z.B. über E-Mail (Angriff auf den Menschen)
   Wenn der Mensch das File ausführt, wird die Malware auf dem Client installiert (ohne Admin Rechte)
   Der Hacker kann ja nicht vom Internet aus auf den Client zugreifen, daher sendet die Malware in regelmässigen Abständen einen Poll (z.B. einen HTTPS Get über Port 443, der fast immer offen ist) an den Command & Control (C&C oder C2) Server des Angreifers, der Server antwortet dann darauf mit beispielsweise einem Script, das dann auf dem Client ausgeführt wird
2. Im Schnitt geht es 6 Tage, bis ein Exploit für eine Privilege Escalation Vulnerability in Windows entwickelt, es dauert aber im Schnitt 54 Tage, bis der Patch installiert ist, der Hacker hat also oft genug Zeit, um die bekannte Vulnerability auszunutzen und Admin-Rechte auf dem Client zu erlangen
3. Mit den Admin-Rechten kann der Angreifer dann Lateral Movement machen und nach und nach das ganze Active Directory übernehmen

Gegenmassnahmen:

- Auslieferung
  - E-Mail-Content Filter
  - Awareness Kampagnen
  - USB-Ports blockieren, wenn möglich
- Ausführung
  - Anti-Virus Check
  - Verhaltensgesteuerte Ansätze
  - Microsoft: Sentinel & Defender
- Netzwerk Monitoring
  - Blockieren von bekannten C2 IPs

- Proxy mit TLS Splitting und Content Inspection



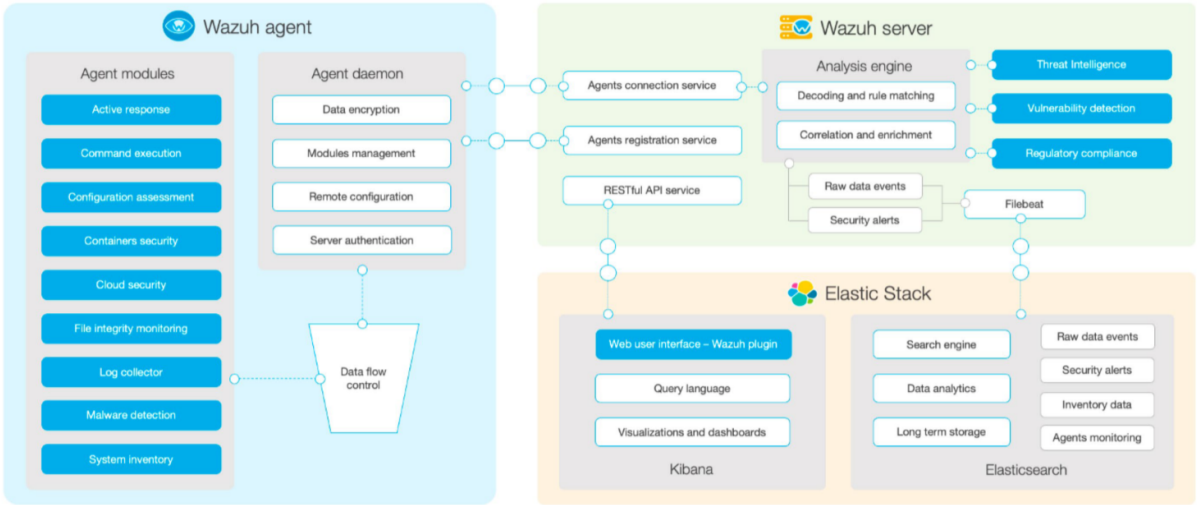
7.6 SCHUTZMECHANISMEN

7.6.1 Endpoint detection and response (EDR)

Als erstes kamen Antivirus-Programme, welche die Clients vor der Ausführung von bekannter Malware schützen. Mittlerweile gibt es Echtzeit-Monitoring und Datensammlung mit regelbasierter automatischer Response und Analyse.

7.6.2 Security Information & Event Management (SIEM)

Bei einem SIEM werden alle Logdaten von Antivirus, Firewall, Proxy, DNS, Server, Applikationen usw. zentral an einem Ort gesammelt. Der zentrale Ort kann z.B. eine Splunk oder Elastic/Kibana Datenbank sein. Ein Open-Source SIEM, das auf Elastic aufbaut ist Wazuh. Alle Daten werden in Echtzeit analysiert, es wird nach Indicators of a Cyberattack (IOC) gesucht und automatisch alarmiert.

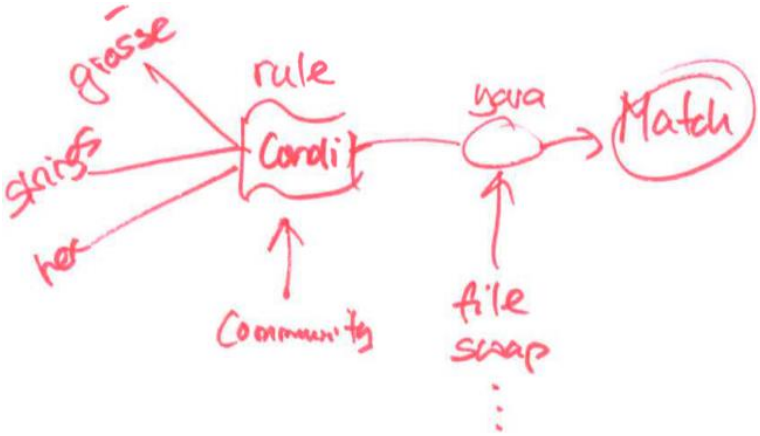


7.6.3 Security Orchestration & Automated Response (SOAR)

Die nächste Generation von SIEM ist SOAR. Es sammelt ebenfalls alle Logs und analysiert sie, zusätzlich kann es aber automatische Interventionen starten und bei der Investigation helfen.

Bei Velociraptor sucht eine Digital Forensic Incident Response (DFIR) Technik in den Bulk Daten nach Mustern. Es kann nach C2 Daten im Prozess-Memory, URL-Daten im Prozess-Memory, Binäre Files nach Malware und Muster in der Registry suchen.

Velociraptor unterstützt auch Suchen mit YARA. YARA kann gemäss vielen Rules gleichzeitig nach binären Patterns in Files oder Memory suchen.



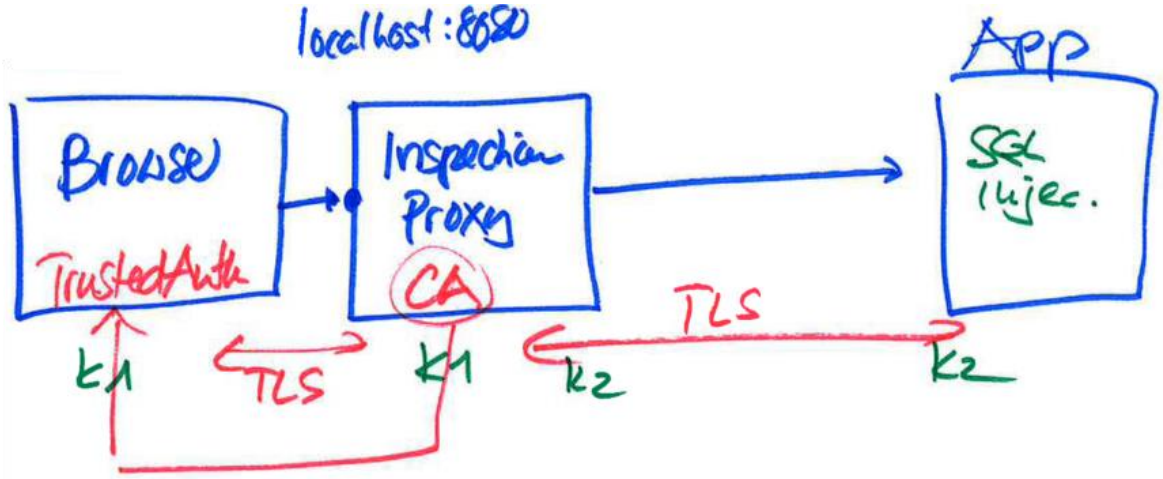
7.7 UNTERSTÜTZUNG

Computer Emergency Response Team (CERT): Trademarked und unpräzise

Computer Emergency Security Incident Response Team (CSIRT): präziser und nicht trademarked

8 Ethical Hacking, Pentesting

8.1 Penetration Testing



8.2 Hacking vs. Ethical Hacking

Beim Hacking nutzt man Vulnerabilities aus, um unautorisierten Zugriff zu erhalten. Man umgeht Sicherheitsmassnahmen, um Dinge zu tun, die der Betreiber davon nicht möchte.

Beim ethischen Hacking verwendet man Tools und Techniken, um Vulnerabilities zu finden und dem Betreiber zu melden.

Hats:

- Black Hat: Bösartiger destruktiver Hacker der anonym bleibt
- Grey Hat: verfügt über Black Hat Skills, arbeitet aber offensiv und defensiv
- White Hat: verfügt über Black Hat Skills, arbeitet aber nur defensiv
- Script Kiddie: Leute die Tools verwenden, ohne zu wissen was sie tun
- Cyber Terrorist: Kompetenter Hacker, der eine Ideologie verfolgt
- State Sponsored: Von der Regierung angestellte Hacker für Offensive und Defensive
- Hacktivist: betreibt sozialen und politischen Aktivismus durch Hacking

Als ethischer Hacker kann man Geld verdienen durch das Melden von Vulnerabilities, beispielsweise auf Hacker One, Bug Crowd oder Bug Bounty.

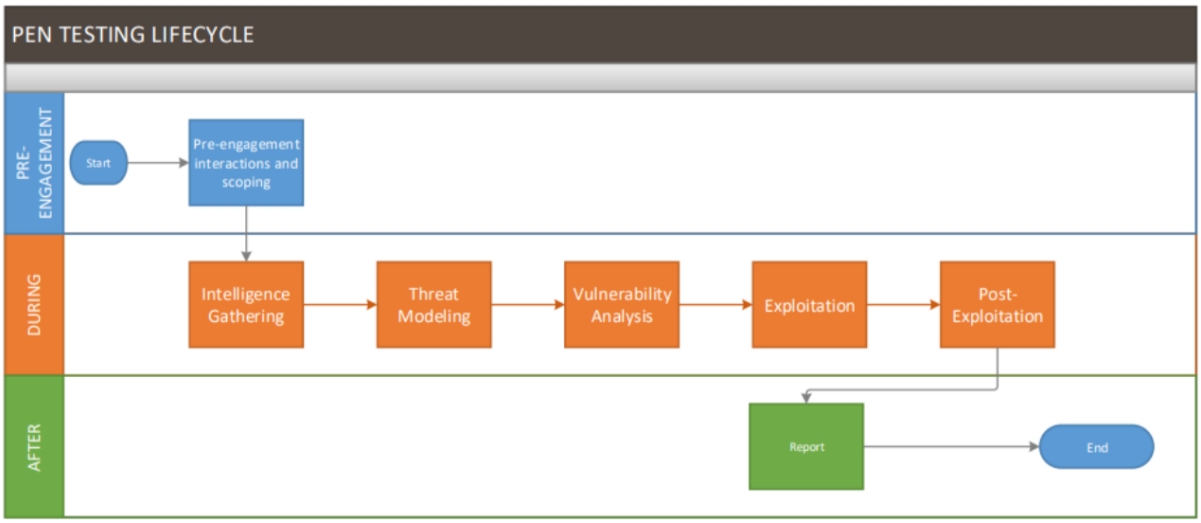
8.3 Penetration Testing

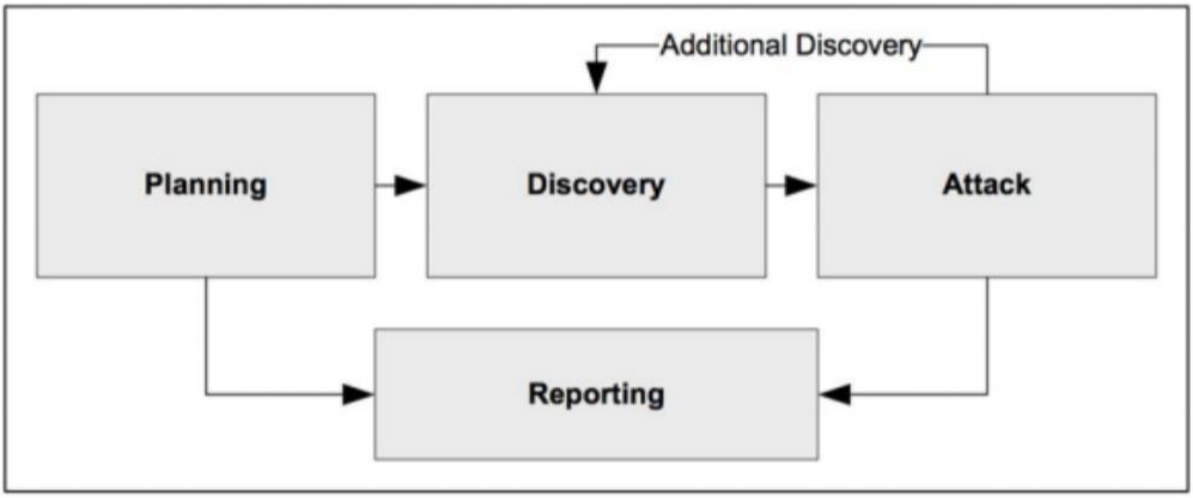
Beim Pentesting engagiert ein Unternehmen einen Hacker, um ihre Systeme auf Vulnerabilities zu überprüfen. Dabei wird zuerst ein Statement of Work (SoW) unterzeichnet, welches besagt, was für Aktivitäten durchgeführt werden sollen, wie lange, der Scope und der Ort. Das SoW kann ein alleiniges Dokument oder Teil eines Master Service Agreements (MSA) sein.

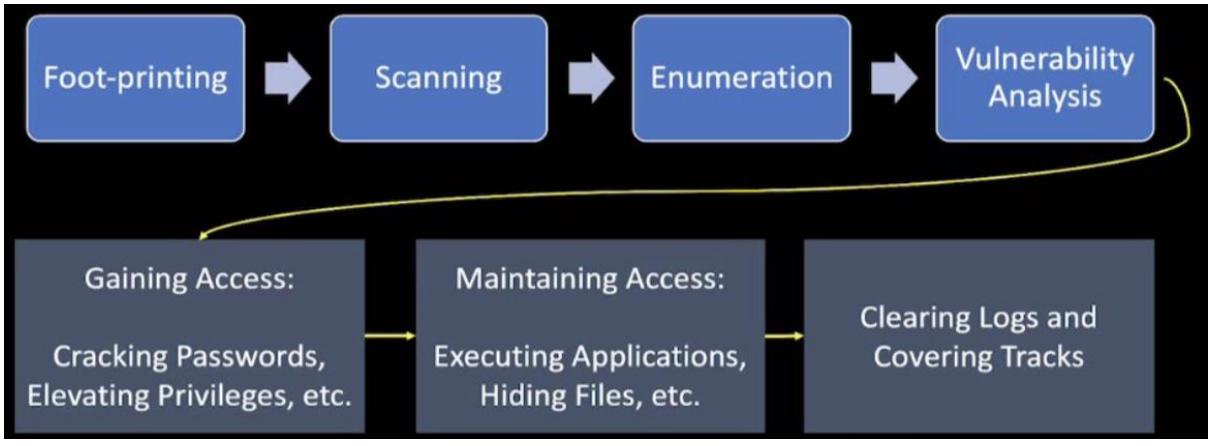
Ausserdem wird ein Non Disclosure Agreement (NDA) abgeschlossen, damit der Hacker allfällige gefundene Lücken nicht öffentlich machen darf.

Beim Vulnerability Scanning hingegen wird automatisch regelmässig gescannt, es ist oft der erste Schritt bei einem Pentest.

Verschiedene Methoden:







Es gibt verschiedene Vorgehen:

- Black-Box: Pentester kennt internen Aufbau nicht, die Inputs und Outputs werden analysiert
- Gray-Box: Pentester kennt die Struktur teilweise und kann das Wissen nutzen
- White-Box: Pentester schaut den Code und die Struktur des Systems an und nutzt das Wissen für die Tests, setzt tiefgehendes Wissen über die Technologie, die zur Entwicklung benutzt wurde, voraus

8.4 Arten von Malware

8.4.1 Zero-Day Angriff

Zero-Day Vulnerabilities wurden von Hackern entdeckt und werden von ihnen ausgenutzt, jedoch gibt es noch keine Patches. Das Zeitfenster zwischen dem Finden vom Zero-Day Exploit und dem Patching/Antivirus Update wird window of vulnerability genannt. Viele Systeme sind lange verwundbar, weil das Patching durch Sysadmins oft lange dauert.

8.4.2 Script Kiddie Angriff

Jeder mit minimalem technischem Verständnis kann fertige Viren herunterladen und im Internet verteilen. Die Kiddies wohnen meist in Ländern mit schwacher Polizei und möchten Geld und Identitäten stehlen.

8.4.3 Drive-By Download

Bei einem Drive-By Download muss nichts angeklickt werden, es kann beim Browsen mit einem veralteten Browser, bei einer App oder dem OS passieren. Die Malware wird unwissentlich einfach heruntergeladen, ohne dass man einen Fehler gemacht hat.

8.4.4 Virus

Ein Virus richtet Schaden an und kann sich selbst verbreiten, es braucht jedoch einen menschlichen Fehler auf jedem Host, um ihn zu infizieren. 200'000 neue Malware Varianten erscheinen im Internet täglich. Die Verbreitungstechnologien werden zunehmend intelligenter, um die immer besser werdenden Antivirus Technologien zu umgehen. Der Schaden kann alle Buchstaben des CIA-Triads treffen. Früher war vor allem Windows betroffen, heute ist kein OS mehr sicher.

Es gibt verschiedene Virus-Verbreitungstechnologien:

- **Master Boot Record (MBR) Infection**: auf jedem bootbaren Medium (Disk, USB-Drive) ist ein kleiner Teil (meist 512 Bytes) für den MBR reserviert. Darin kann ein Virus installiert werden, der den Rest des Virus z.B. in den Memory lädt.
- **File Infection**: Viren werden oft in ausführbaren Files (.com, .exe oder .bat auf Windows) gespeichert. Um nicht sofort aufzufallen, heissen sie ähnlich wie normale OS-Files, jedoch wird es meist schnell durch AV Programme entdeckt.
- **Macro Infection**: Macros sind Scripts in einer Macro Language, welche in Word und Excel verwendet werden. Sie werden automatisch ausgeführt wenn die Dokumente geöffnet werden und installieren dann den Virus. Änderungen bei den Berechtigungen der Macros haben aber diese Angriffe stark reduziert.
- **Service Infection**: Der Virus wird in normale Prozesse des OS injiziert wie svchost.exe, winlogin.exe oder explorer.exe. Es fällt nicht auf, da die Prozesse trusted werden.

Dann gibt es noch verschiedene Arten von Viren:

- **Multipartite Virus**: hat mehrere Verbreitungstechniken, falls eine nicht funktioniert nimmt er eine andere, z.B. probiert er zuerst File Infection und dann MBR Infection
- **Stealth Virus**: verstecken sich und geben dem Antivirus saubere Angaben, z.B. wird der MBR überschrieben, wenn der Antivirus jedoch den MBR prüft, sendet ihm der Virus eine saubere Kopie
- **Polymorphic Virus**: der Virus-Code ändert sich von Host zu Host, somit ist die Signatur anders, er macht aber immer noch das Gleiche, das ist schwerer zu finden, die Formeln

dieser Viren wurden aber oft geknackt, es dauert aber auf jeden Fall länger bis AV-Updates dafür erscheinen

- **Encrypted Virus**: dabei ist irgendwo ein kleiner Code, der den Virus entschlüsseln kann, der eigentliche Virus ist aber verschlüsselt irgendwo anders auf dem System, zusätzlich ändert sich der Key von System zu System, somit ist hat der Virus immer eine komplett andere Signatur

8.4.5 Logic Bombs

Diese Malware macht nichts, bis sie triggered wird, beispielsweise von einer Zeit, einem Programmstart oder Website Login.

8.4.6 Trojan Horses

Das ist eine Software, die nützlich und sauber aussieht, im Hintergrund aber noch etwas anderes bösartiges macht.

8.4.7 Keystroke logging

Ein Keylogger zeichnet alle getippten Buchstaben auf einer Tastatur auf, ohne dass die Person es merkt. Das kann entweder eine Software sein oder Hardware (USB-Dongle mit PC auf der einen und Tastatur auf der anderen Seite).

8.4.8 Ransomware

Ransomware verschlüsselt alle Files auf einem infizierten System mit einem Key, den nur der Angreifer weiss. Gegen eine Gebühr werden die Daten möglicherweise wieder entschlüsselt. Um zusätzlichen Druck auszuüben, werden noch Daten geklaut und veröffentlicht, wenn nicht bezahlt wird.

8.4.9 Worms

Worms verbreiten sich selbständig auf andere Hosts, ohne menschliches Zutun. Beispielsweise scannen sie das Netzwerk nach verwundbarer Software und greifen diese dann an.

8.4.10 Spyware

Alle Aktivitäten auf dem infizierten Gerät werden aufgezeichnet und an den Angreifer gesendet.

8.4.11 Adware

Werbebanner werden auf dem infizierten System angezeigt, beispielsweise beim Surfen. Fortgeschrittene Adware könnte auch das Einkaufsverhalten analysieren auf Konkurrenzseiten weiterleiten. Sie benutzen oft Browser-Plugins.

8.5 ANTIVIRUS

Die Aufgaben eines AV sind Malware zu verhindern, entdecken und entfernen.

1. Es wird probiert den Virus aus den infizierten Files zu entfernen und den Host in einen sicheren Zustand zu bringen
2. Wenn das nicht geht, werden die Files in Quarantäne gesteckt, oft werden sie dann in einer isolierten Sandbox ausgeführt und monitort
3. Wenn das auch nicht geht, werden die Files einfach gelöscht

8.5.1 Detection

- Signature-based: Der AV hat eine riesige Datenbank mit den Charakteristiken von bekannter Malware, die DB muss so oft wie möglich aktualisiert werden

- Heuristic-based: das Verhalten von Software wird analysiert und nach Zeichen bösartiger Aktivität gesucht wie Privilege Escalation, wenn eine Software verdächtig agiert, wird es sofort auf eine Blacklist für alle Hosts der Organisation gesetzt
- Data Integrity: Es wird eine DB mit Hashes aller Files des Hosts erstellt, ohne Patching von OS oder Software sollten sich die Hashes nicht ändern, wenn schon ist das ein Zeichen für eine Infektion, das geht z.B. mit Tripwire

8.6 Application Attacks

8.6.1 Buffer Overflows

Das kann vorkommen, wenn eine Applikation den User Input nicht auf eine sinnvolle Länge limitiert. Dann schreibt die Applikation mehr in den Memory, als Platz vorgesehen ist. Somit können möglicherweise Commands ausgeführt werden.

8.6.2 Time of Check to Time of Use

TOCTTOU oder TOC/TOU ist ein Timing Vulnerability, bei der das Programm die Berechtigungen zu früh vor der Ressourcenanfrage prüft. Beispielsweise hat der Admin eine Berechtigung entzogen, es gilt aber erst nach dem nächsten Login. Wenn der User nun die Session nie beendet, hat er die Berechtigung immer noch.

8.6.3 Backdoor

Backdoors sind undokumentierte Wege zum Zugriff auf Systeme, welche die normalen Zugriffsbeschränkungen umgehen. Sie können durch Malware-Installation erstellt werden oder aus Versehen durch Softwareentwickler, welche beispielsweise ohne Authentication auf Debugging Logs zugreifen möchten.

8.6.4 Escalation of Privilege / Rootkits

Eine Escalation of Privilege ist ein Angriff, bei dem ein Angreifer seine Berechtigung von einem normalen User zu einem User mit mehr Privilegien erhöht. Ein Weg dafür ist ein Rootkit, welches Vulnerabilities.im OS ausnutzt, um Root/Administrator zu werden.

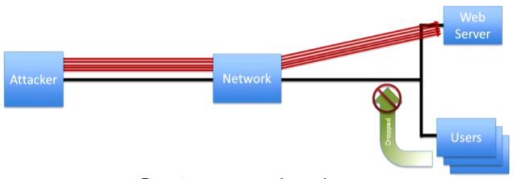
8.7 Network Security

Es macht Sinn, Netzwerkgeräte anzugreifen, aus folgenden Gründen:

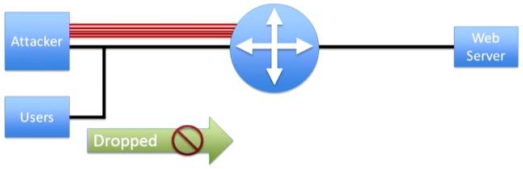
- Werden oft nicht so gut monitored wie Hosts
- Haben einen langen Lifecycle
- Kein Antivirus
- Funktionieren mit Protokollen aus den 80ern
- Features wie Port Mirroring können ausgenutzt werden
- Protocol Authentication unterstützt nur MD5

8.7.1 Denial of Service (DoS)

Dabei werden so viele Ressourcen benutzt, dass normale Anfragen nicht mehr möglich sind. Man kann Server oder Netzwerkgeräte überlasten und es gibt immer genau einen Angreifer und ein Ziel.



System overload



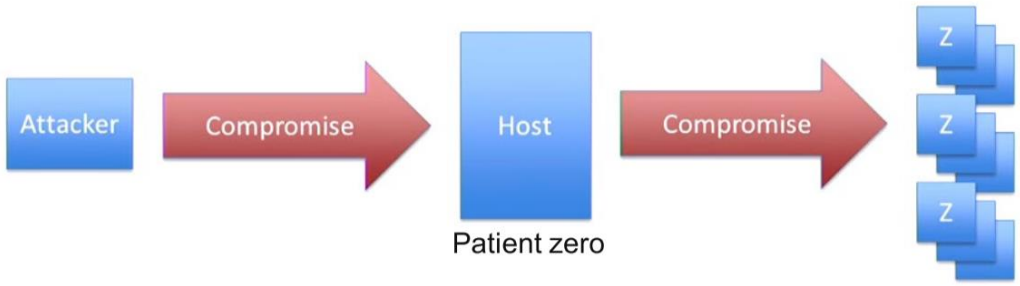
Link overload

Es gibt verschiedene Methoden für DoS:

- **SYN Flooding**: es werden viele TCP SYN Anfragen an den Server gesendet, immer mit einer anderen gespooften IP-Adresse. Der Server Antwort mit SYN/ACK, somit ist die Session halboffen und wird nicht beendet. Irgendwann hat der Server keine Kapazität mehr für noch mehr TCP-Sessions.
- **Service Request Floods**: Es werden HTTP Get Requests gesendet, bis der Server nicht mehr mithalten kann mit antworten.
- **Application Level**: Es wird eine Vulnerability in Hard- oder Software ausgenutzt, ein bestimmtes Paket wird gesendet, welches das System zum Crashen, Hängen, Freezen oder Überlasten bringt
- **Permanent DoS**: Mit Phlashing wird Hardware permanent zerstört über das Netzwerk, z.B. indem man ein bösartiges Firmware Update installiert.

8.7.2 Distributed Denial of Service (DDoS)

Ein Botnet ist eine logische Collection von Geräten mit Internetverbindung. Jedes Gerät hat eine Malware installiert, die Aktionen ausführen kann, welche vom C&C Server des Angreifers vorgegeben werden. Das Botnet kann sich selbständig vergrössern und wird oft vermietet für DDoS Attacken.

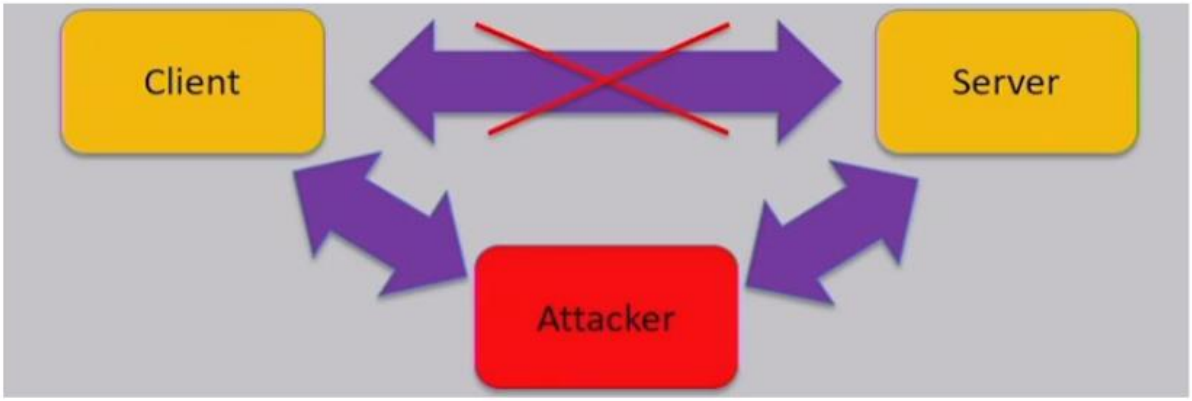


Patient zero

Bei einem DDoS Angriff führt jeder Bot/Zombie des Botnets einen DoS Angriff auf ein vorgegebenes Ziel aus. Der Angegriffene kann zwar die einzelnen Zombies ausfindig machen, jedoch findet er den wahren Angreifer dahinter nicht heraus.

8.7.3 Man-in-the middle

Bei einem Man-in-the-middle Angriff schaltet sich der Angreifer zwischen Client und Server und lässt den ganzen Traffic durch seinen Host fliessen und kann ihn so kontrollieren.



8.7.4 Man-in-the browser

Dabei ist ein Trojaner getarnt als Browser Plugin im Browser installiert. Es kann z.B. Logins abgreifen durch POST-Interception, XSS und XSRF ausführen oder Session hijacken.

8.7.5 Eavesdropping

Dabei wird Traffic abgehört mit einem Sniffer um sensible Daten abzugreifen oder zu duplizieren für andere Angriffe.

8.7.6 Impersonation/Masquerading

Dabei wird vorgegaukelt, jemand anders zu sein, um unautorisierten Zugriff auf ein System zu erhalten, meist geschieht das durch gestohlene Credentials.

8.7.7 Spoofing

Dabei wird ebenfalls vorgegaukelt, jemand anders zu sein, jedoch ohne einen Beweis dafür zu haben. Es kann z.B. eine falsche MAC, IP, E-Mail, Hostname, Domainname, usw. sein.

8.7.8 Replay Attack

Dabei möchte eine authentifizierte Session erneut hergestellt werden, indem man aufgezeichneten Traffic nochmals dem System sendet. Somit ist Eavesdropping eine Voraussetzung.

8.7.9 Modification Attack

Das ist ähnlich wie der Replay Attack, jedoch wird der Traffic vorher noch abgeändert, um etwas ausgereiftere Authentication Systeme auszutricksen.

8.7.10 Session Hijacking

Session IDs werden verwendet, um eine stateful Verbindung zwischen Client und Server aufzubauen. Sie werden meist nach dem Login generiert. Wenn sie einem Angreifer in die Hände fallen, hat er Zugriff auf die Session.

Beim Session Hijacking zeichnet der Angreifer Session Keys auf und probiert ein Muster zu finden, um dann im Brute Force verfahren Sessions zu hijacken.

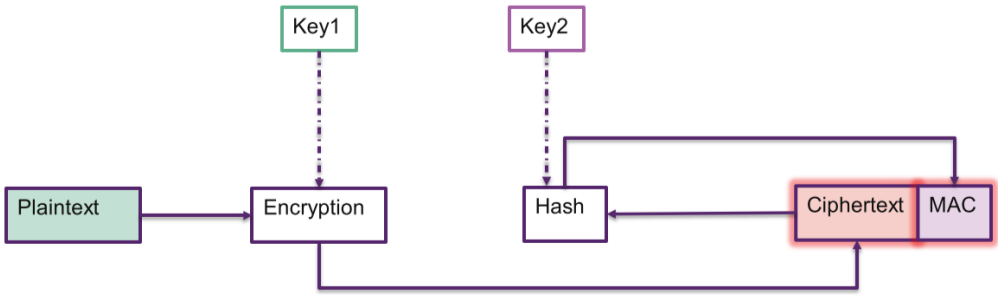
Das Hijacking gelingt durch Session Fixation (Attacker probiert den Client in eine von ihm erstellte Session zu locken), XSS, CSRF oder TCP/IP Hijacking (Angreifer antwortet schneller beim 3-Way Handshake sodass die TCP Session mit ihm aufgebaut wird).

9 Complete Cryptographic Systems

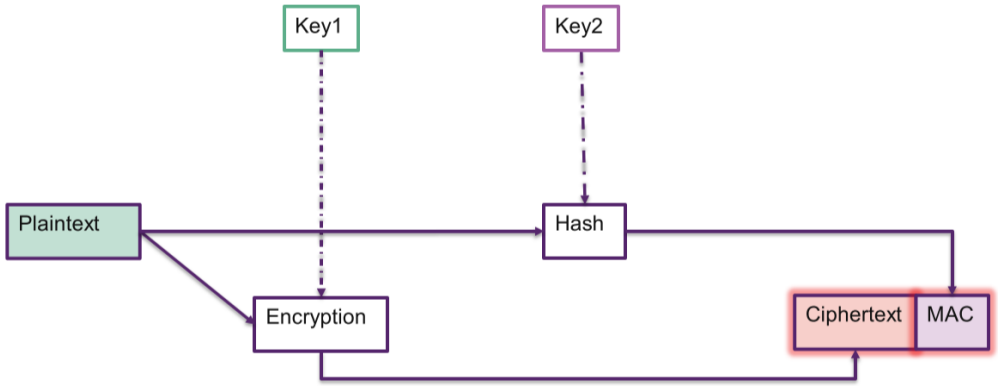
9.1 Authenticated Encryption

Es gibt 3 Arten von Authenticated Encryption:

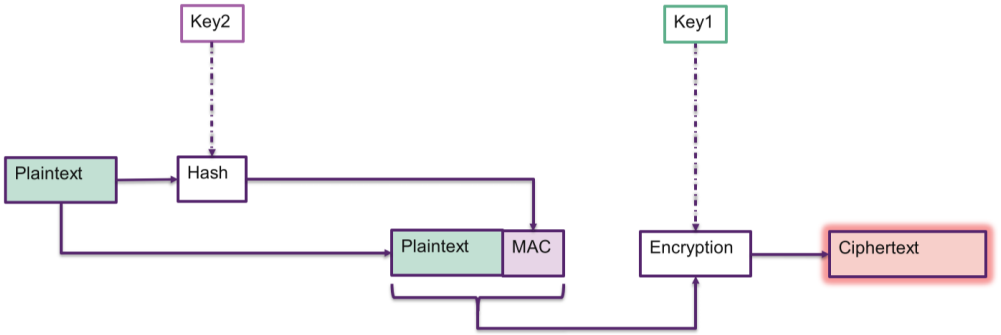
- Encrypt-then-MAC (SSL)



- Encrypt-and-MAC (IPsec)

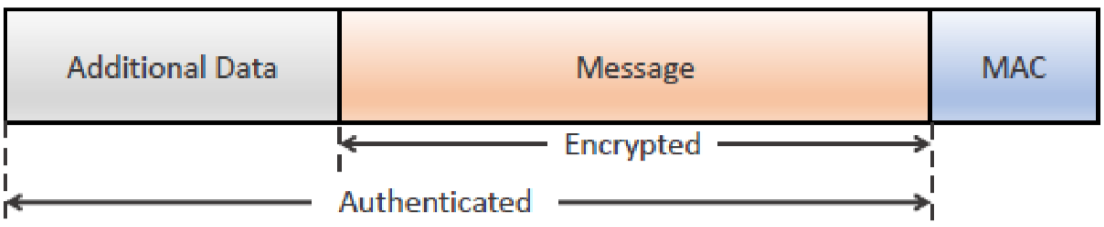


- MAC-then-encrypt (SSH)



9.2 Authenticated Encryption with Associated Data (AEAD)

Das ist eine andere Variante von Authenticated Encryption. Damit kann der Empfänger die Integrität von sowohl den verschlüsselten als auch den unverschlüsselten Daten überprüfen. Die zusätzlichen associated Daten werden zusammen mit dem Ciphertext gehasht.



Beispiel einer AEAD ist der AES Galois Counter Mode (GCM). Das Standardverfahren von AES für die Verschlüsselung der Blocks ist CTR. GCM berechnet dann einen Galois Message Authentication Code (GMAC) über den Ciphertext und die associated Daten.

Ein anderes Beispiel ist Chacha20_Poly1305. Chacha20 ist eine Stream Cipher, welche auf CPUs ohne AES schnneller läuft. Poly1305 erstellt den MAC. Diese Cipher ist neben AES in TLS 1.3 erlaubt.

10 Transport Layer Security (TLS)

TLS macht Verschlüsselung auf Layer 4 und stellt somit Confidentiality und Integrity sicher. Optional auch Authentication. TLS ist der Nachfolger von SSL, der Begriff SSL wird aber immer noch oft verwendet. Sämtliche HTTPS Kommunikation basiert auf TLS, es kann aber für jedes beliebige Protokoll eingesetzt werden.

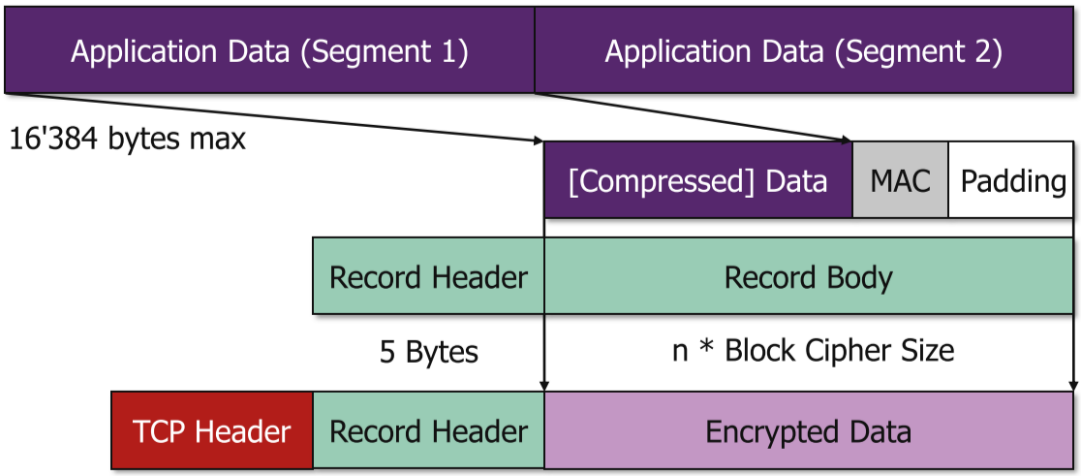
10.1 Versionen

Version	***Status***	***Bemerkungen***
SSL 1.0	Broken	Unpublished
SSL 2.0	Broken	
SSL 3.0	Broken	
TLS 1.0	Broken	RFC 2246, ziemlich identisch mit SSL 3.0
TLS 1.1	Broken	RFC 4346, einige Security Fixes und Ciphers entfernt
TLS 1.2	Sicher wenn richtig konfiguriert	RFC 5246, AE hinzugefügt, Hashing verbessert, TLS Extensions hinzugefügt
TLS 1.3	Sicher	RFC 8446, Redesign für Performance, viele Ciphers entfernt, AEAD ist Pflicht

10.1.1 TLS 1.3

- Broken Crypto entfernt: DES, 3DES, RC4, MD5, SHA-1, Kerberos, RSA PKCS#1v1.5
- Broken Features entfernt: Komprimierung und Renegotiation
- Static RSA/DH entfernt (verhindert passive Inspektion durch Enterprise Proxies)
- Performance verbessert: 1-RTT und 0-RTT Handshakes
- Privacy verbessert: fast alle Handshake Messages sind bereits verschlüsselt
- Rückwartskompatibel

10.2 Headers



10.3 Content Types

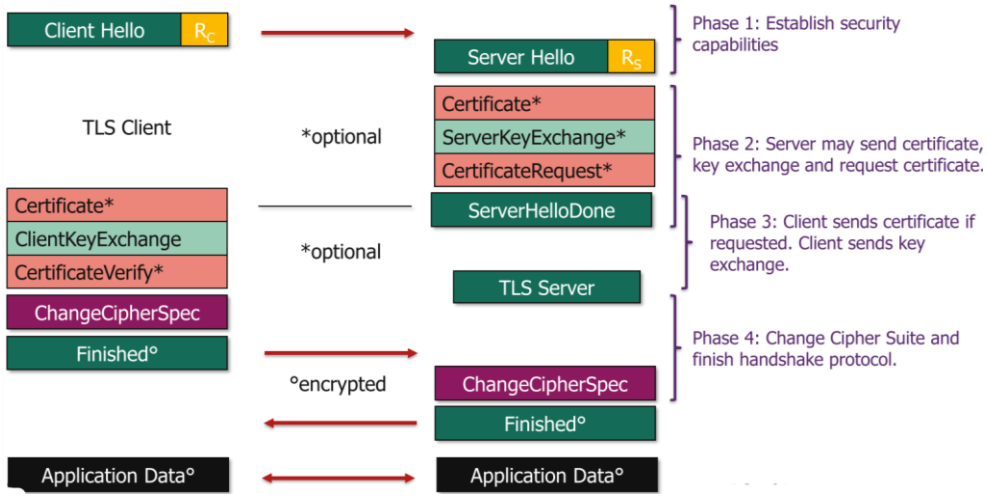
- Handshake
  - ClientHello
  - ServerHello
  - Certificate
  - ServerHelloDone
- Change Cipher Spec
- Alert
  - Warning
  - Fatal
- Application: Übertragung von verschlüsselten Daten

10.4 Handshake

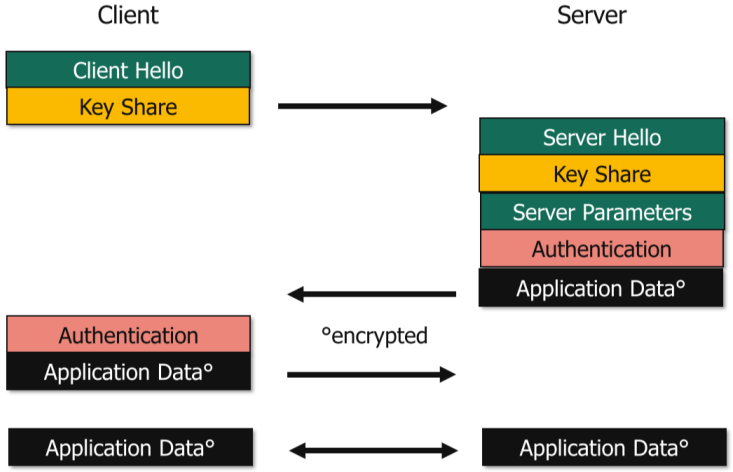
Der Handshake ist der komplizierteste Teil von TLS, er muss vor der Datenübertragung zwischen Client und Server gemacht werden. Er hat folgende 3 Aufgaben:

- Gegenseitige Authentifizierung
- Encryption- und MAC-Algorithmus verhandeln
- Kryptografische Keys verhandeln

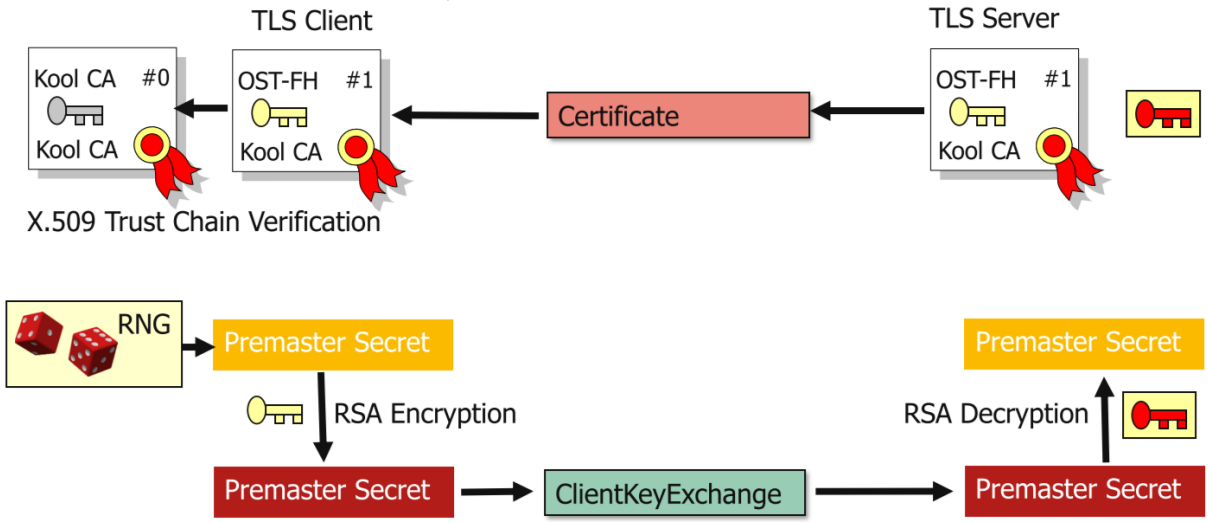
10.4.1 TLS 1.2



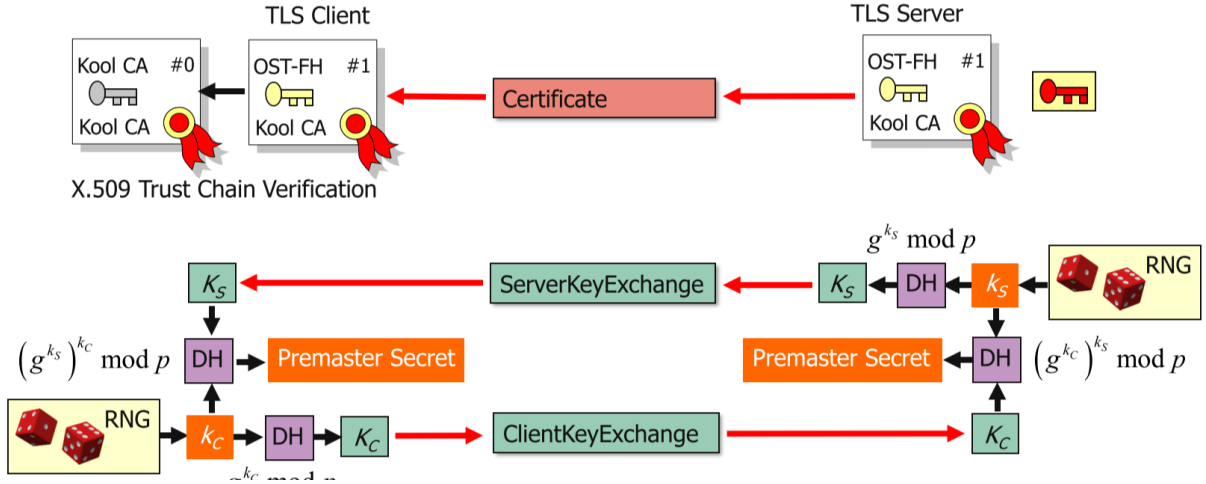
10.4.2 1-RTT Handshake (TLS 1.3)



10.4.3 Ohne Perfect Forward Secrecy (unsicher)

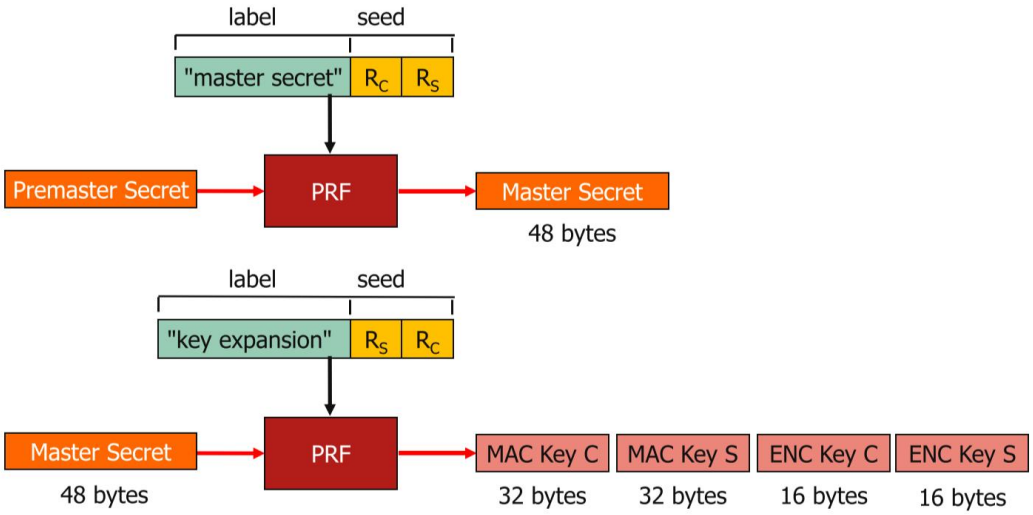


10.4.4 Mit Perfect Forward Secrecy (sicher)

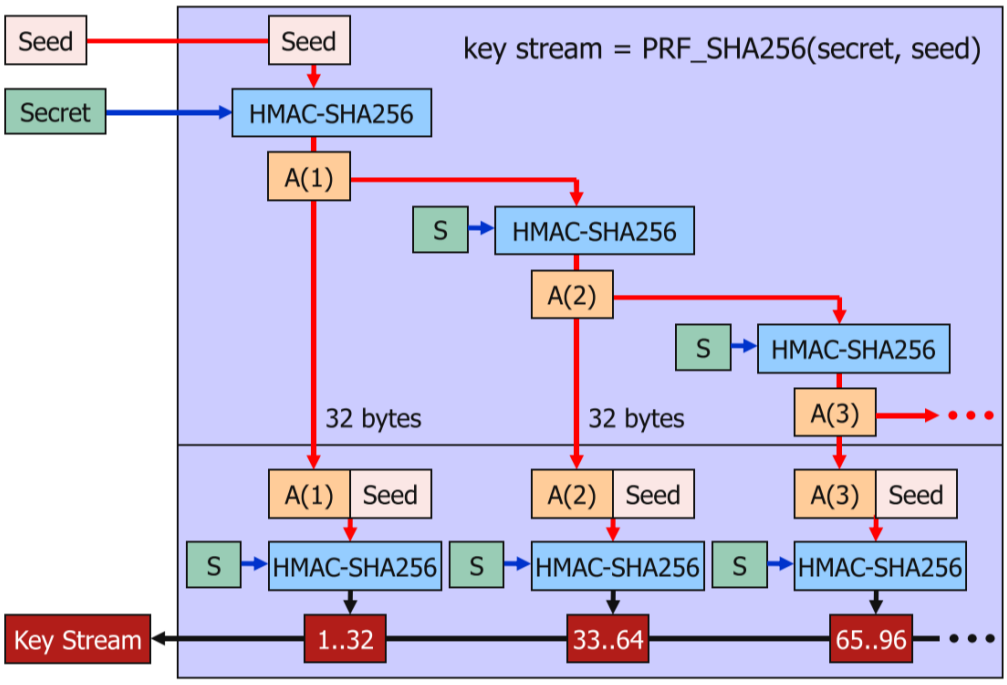


Mit PFS schützt man sich davor, dass aufgezeichneter, vergangener Traffic nicht entschlüsselt werden kann, selbst wenn der Private Key des Servers gestohlen wird. Es werden ständig neue DH-Aushandlungen durchgeführt, somit einstehen immer neue Session Keys und wenn man einen klaut, hat man keinen Anhaltspunkt für den nächsten Session Key.

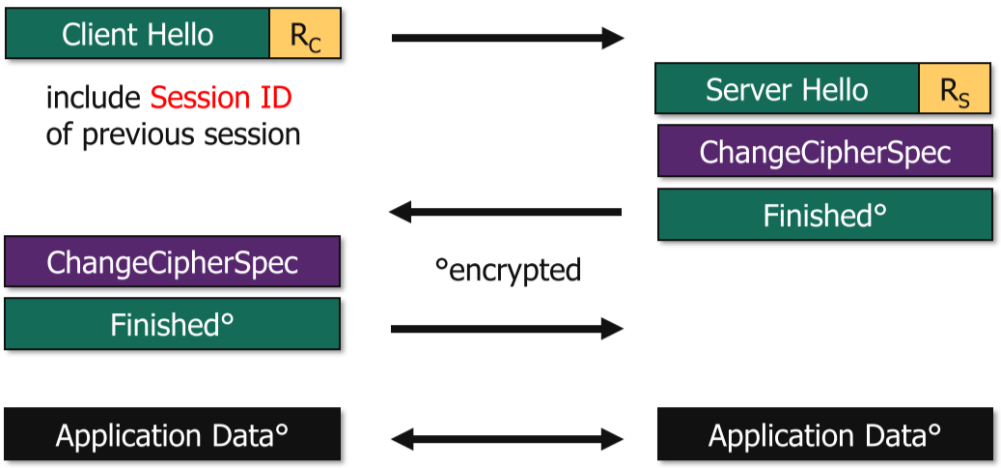
10.4.5 TLS Master Secret und Session Keys aus Premaster Secret berechnen



PRF = Pseudo Random Function:

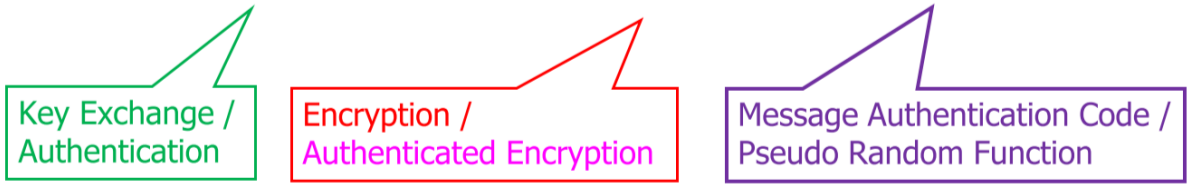


10.4.6 Session Resumption



10.5 Cipher Suites

TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256



10.5.1 TLS 1.3

Nur noch 5 Cipher Suites untersützt, bei TLS 1.2 waren es noch mehr als 100.

TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_128_CCM_SHA256 (IoT)
TLS_AES_128_CCM_8_SHA256 (IoT)

10.6 Heartbeat Protocol

Ein periodisches Signal, welches die Verfügbarkeit eines Protokolls überwacht. Es läuft auf dem TLS-Protokoll und wird in Phase 1 vom Handshake definiert. Es hat zwei Zwecke:

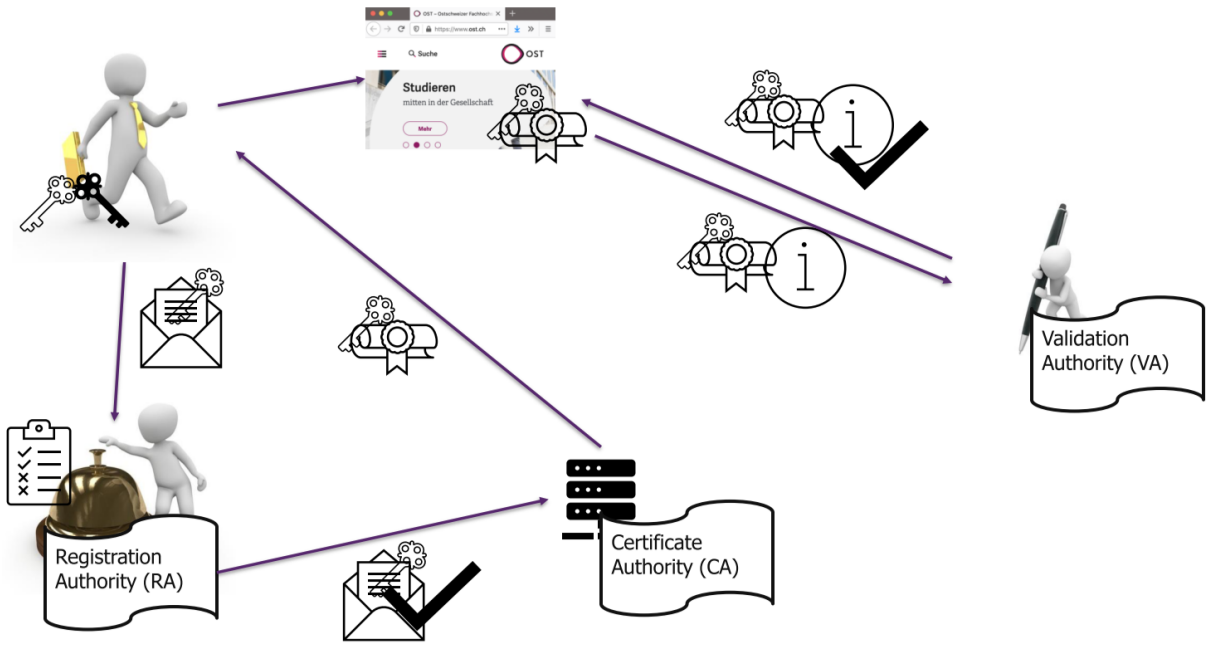
- Sicherstellen, dass Sender und Empfänger alive sind
- Aktivität auf der Verbindung, während Idle Zeiten kreieren

11 PUBLIC KEY INFRASTRUCTURE

Eine PKI dient der Authentifizierung, Verschlüsselung und Signierung. Es wird sichergestellt, dass das Zertifikat wirklich zu dieser Website gehört und gültig ist. Es wird ein Public Key (Zertifikat) einer Person oder Organisation zugeordnet.

Begriffe:

- **Registration Authority (RA)**: Stelle, die sich um Certificate Signing Requests kümmert, Users und Devices authentifiziert und validiert und Credentials revoked
- **Certificate Authority (CA)**: Stellt signierte Zertifikate aus
- **Validation Authority (VA)**: Stelle die die Gültigkeit von Zertifikaten (nicht revoked) und CAs bestätigt



11.1 X.509

Das ist der Standard für Public Key Zertifikate. Aktuell ist die Version 3.

Ein Public Key Zertifikat beweist den Besitz des Keys, enthält Informationen über den Key, den Owner (Subject) und den Issuer (CA).

```
Certificate ::= SEQUENCE {
    tbsCertificate TBSCertificate,  signatureAlgorithm AlgorithmIdentifier, signature BIT STRING }

tbsCertificate ::= SEQUENCE {
    version
    serialNumber,
    signature,
    issuer
    validity
    subject
    subjectPublicKeyInfo
    issuerUniqueID: OPTIONAL,
      -- If present, version MUST be v2 or v3
    subjectUniqueID: OPTIONAL,
      -- If present, version MUST be v2 or v3
    Extensions: OPTIONAL
      -- If present, version MUST be v3
}

AlgorithmIdentifier ::= SEQUENCE {
    algorithm OBJECT IDENTIFIER,
    parameters ANY DEFINED BY algorithm  OPTIONAL }
```

11.1.1 Extensions

Key Usage:

- 0: digitalSignature
- 1: nonRepudiation
- 2: keyEncipherment
- 3: dataEncipherment
- 4: keyAgreeement
- 5: keyCertSign
- 6: cRLSign
- 7: encipherOnly
- 8: decipherOnly

Extended Key Usage: serverAuth, clientAuth, codeSigning, emailProtection, timestamping oder OCSPSigning

Certificate Policies: Jede CA hat hier eigene Policies

Authority Key Identifier (AKI): Key Identifier des CA-Zertifikats mit dem das Zertifikat signiert wurde

Subject Alternative Name (SAN): Zusätzliche DNS-Namen/IP-Adressen für die das Zertifikat gültig ist

Subject Key Identifier (SKI): Hash-Value des Zertifikats

11.1.2 Revocation

Zertifikate können widerrufen werden. Es gibt 2 Methoden, um zu prüfen, ob ein Zertifikat revoked wurde:

- **Certificate Revocation List (CRL)**: Das Feld «CRL Distribution Point» im Zertifikat verweist auf einen Ort mit der CRL. Jeder moderne Browser prüft das Zertifikat an diesem Ort.
- **Online Certificate Status Protocol (OCSP)**: Im Authoritative Information Access (AIA) Feld hat es Informationen über die CA, z.B. OSCP-Validierung oder CA Policy Daten

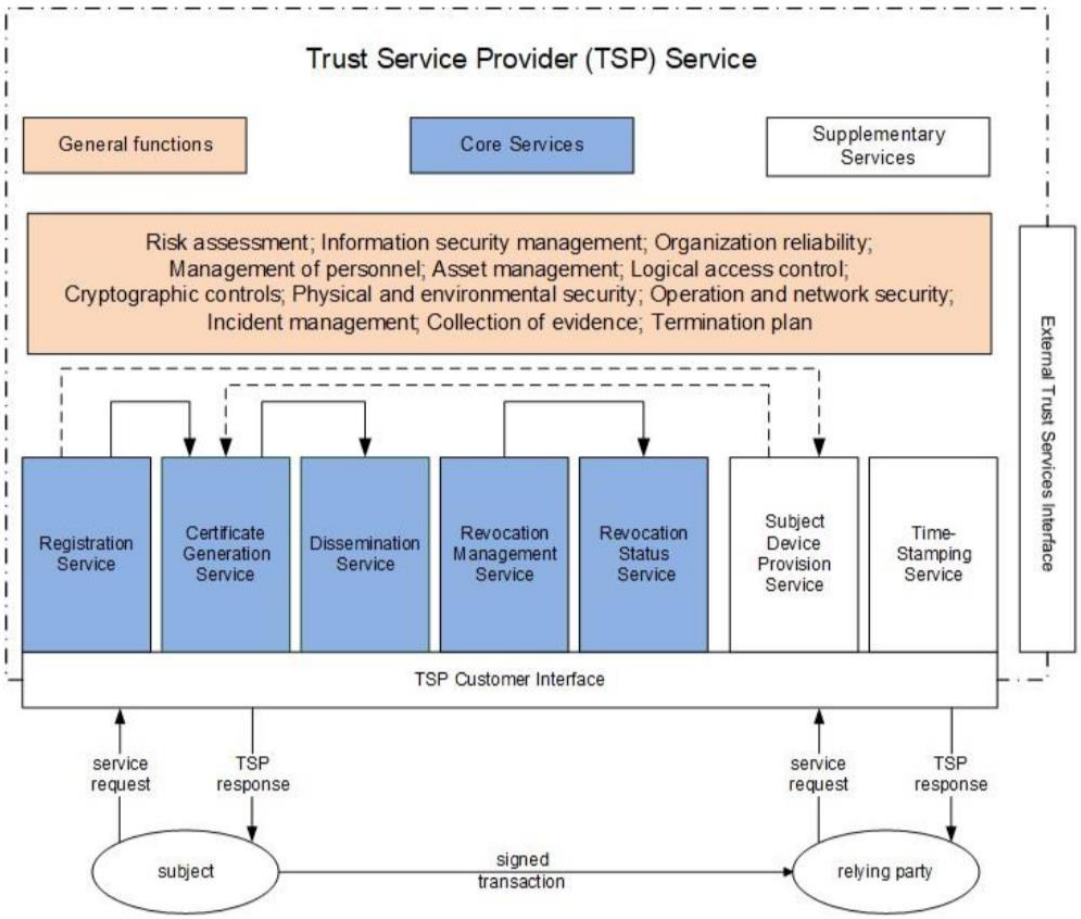
11.2 Verifizierungsmöglichkeiten

Es gibt verschiedene Arten von Verifizierungsmöglichkeiten durch CAs, um die Qualität des Zertifikats zu bestimmen:

- **Domain Validated (DV)**: Der Requester muss beweisen, dass er die Domain kontrolliert, für die er ein Zertifikat möchte.
- **Organisation Validated (OV)**: CAs müssen Firmennamen, Domainnamen und andere Daten in öffentlichen Verzeichnissen prüfen
- **Extended Validation (EV)**: Der Requester muss eine strikte Authentifizierungs-Prozedur durchlaufen.
- **Qualified Website Authentication Certificate (QWAC)**: Qualifiziertes Zertifikat, das die Anforderungen der eIDAS und PSD2 Regulierungen erfüllt

11.3 Trust Service Provider (TSP)

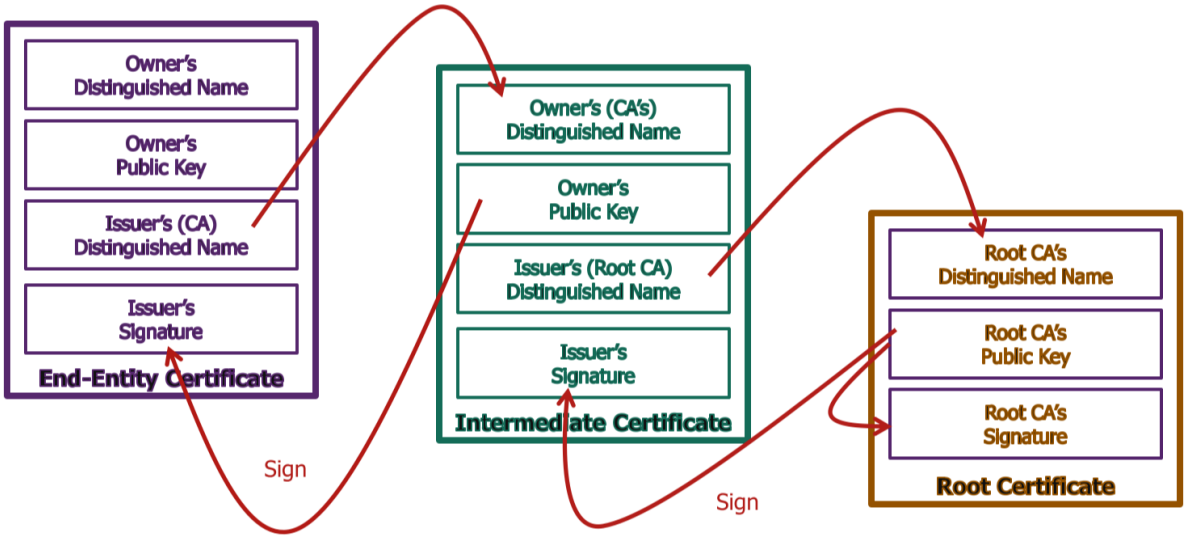
Ein TSP stellt Vertrauen zwischen kommunizierenden Parteien sicher. Er stellt vertrauenswürdige Identitäts-Informationen, sichere Authentifizierung, integere Kommunikation und verschlüsselte Kommunikation zur Verfügung.



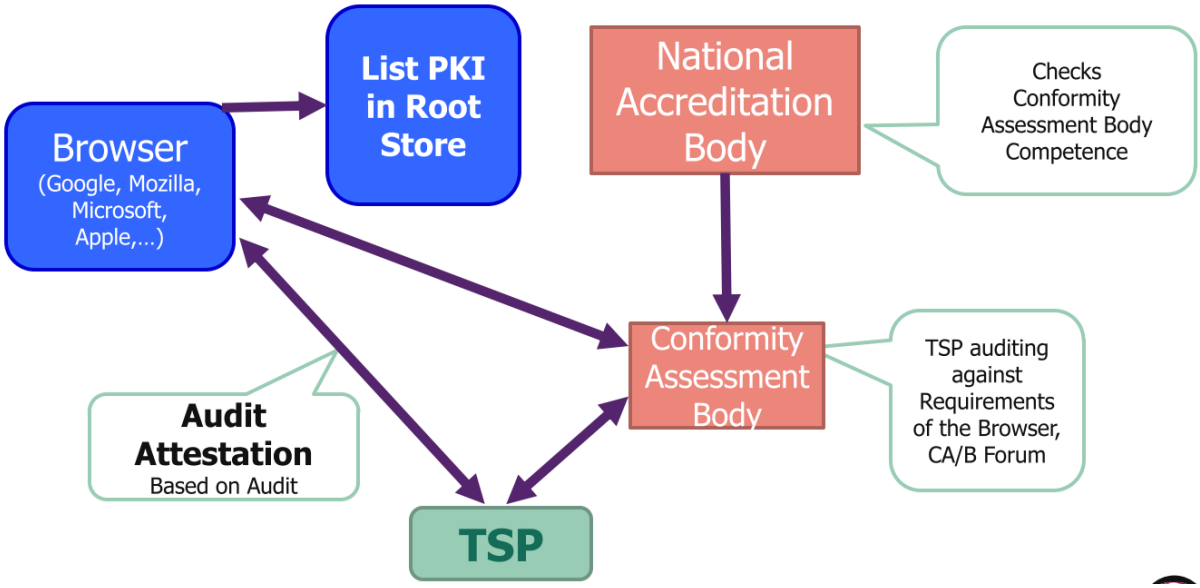
11.3.1 Zertifikatsausstellung

1. Der Subscriber (Besitzer des Key Pairs, zu dessen Public Key ein Zertifikat ausgestellt werden sollte) erstellt auf der Basis des Public Keys einen Certificate Signing Request (CSR).
2. Der CSR enthält Informationen zum Objekt, das signiert werden sollte und Informationen zu Key Typ und Länge.
3. Der CSR wird im Base64-Format der CA gesendet.
4. Die CA prüft die Daten und wenn alles in Ordnung ist, signiert sie den CSR und stellt ein X.509 Zertifikat aus.

11.3.2 Chain of Trust



Die Hersteller von Browsern und OS müssen entscheiden, welchen CAs sie vertrauen. Dafür gibt es das CA / Browser Forum (CA/B).



Zum Teil gibt es auch regulatorische Anforderungen wie die eIDAS Regulation in der EU, NIST/FIPS, usw.

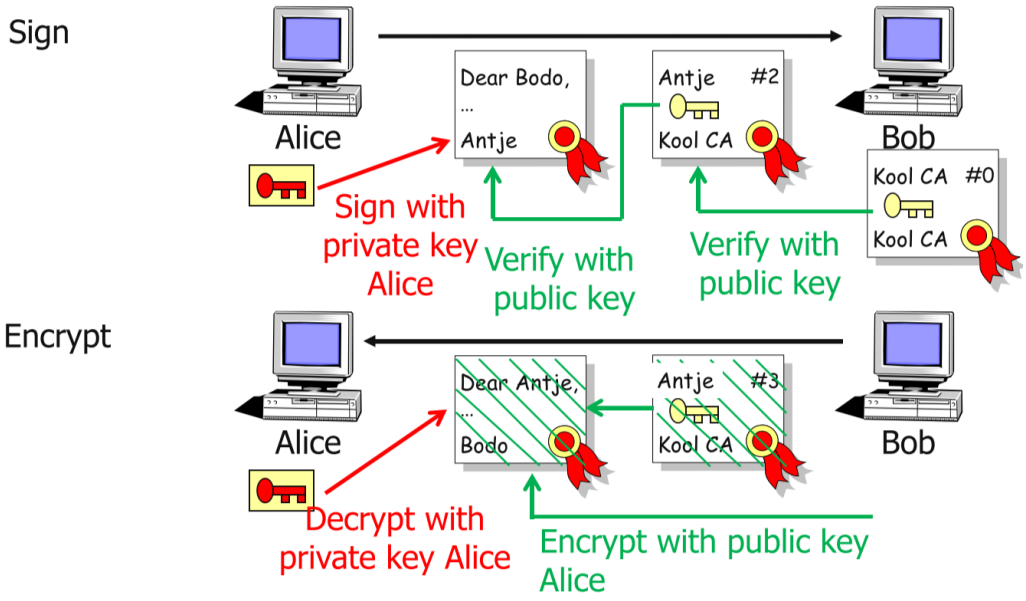
11.4 Certificate Pinning

Beim Certificate Pinning muss ein Zertifikat einer Website nicht von einer Trusted Root CA stammen. Stattdessen kann man manuell das Zertifikat, das Intermediate Zertifikat oder das Root CA pinnen für die bestimmte Website pinnen. Das ist nützlich für Tests.

12 E-MAIL SECURITY

12.1 S/MIME

S/MIME heisst Secure Multipurpose Internet Mail Extension. Damit können Mails sowohl verschlüsselt als auch signiert werden.



Es gibt 2 Typen für Signed Messages:

- Type 1: Die Nachricht wird im Klartext gesendet, als Anhang wird die Signatur im PKCS #7 Format mitgesendet.

  ASN.1 structure for the **SignedData** content type

  ```
  version
  digestAlgorithms
  contentInfo
  certificates (OPTIONAL)
  crls (OPTIONAL)
  signerInfos (SET OF)
  ```

  empty field (content carried in separate MIME entity)

  several signers possible

  ASN1 = Abstract Syntax Notation One

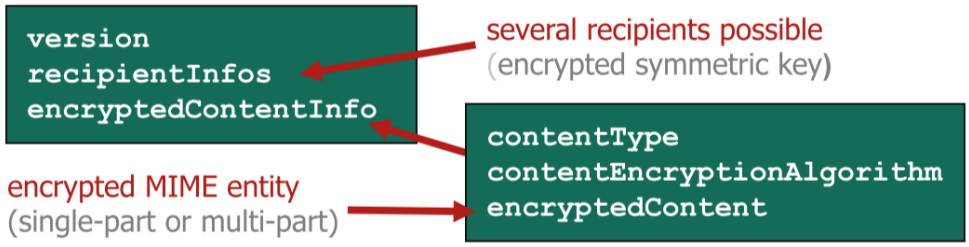
  ASN.1 structure for the **SignerInfo** type

  ```
  version
  issuerAndSerialNumber
  digestAlgorithm
  authenticatedAttributes
  digestEncryptionAlgorithm
  encryptedDigest
  unauthenticatedAttributes
  ```

  signature

- Type 2: Dabei wird der Content nicht als Clear Text gesendet, sondern im PKCS#7 Signed Data Object. Der Vorteil davon ist, dass der MIME-Inhalt nicht anfällig ist für Veränderungen beim Transport, der Mail Client des Empfängers muss jedoch S/MIME unterstützen.

Bei einer verschlüsselten Mail wird das Mail mit einem symmetrischen Key verschlüsselt als Anhang gesendet. Auch hier gibt es zwei Typen, beim ersten wird vor dem Verschlüsseln signiert und beim zweiten nach dem Verschlüsseln.



12.2 SPAM

Ab 2002 wurde Spam über E-Mail zu einem ernsthaften Problem, ab 2010 wurde es weniger.

Man setzte Apache SpamAssassin ein, um Spam zu erkennen. Diese Software analyisiert alle Teile des Mails und gibt ihm einen Score, man kann konfigurieren, ab welchem Score das Mail als Spam klassifiziert wird. Die Software lernte mithilfe der ankommenden Mails laufend dazu.

Wenn man zu wenig zulässt gibt es viele False Positives und wenn man zu viel zulässt gibt es viele False Negatives, man muss also eine gute Balance finden.

12.2.1 Sender Policy Framework (SPF)

SPF Records sind TXT Records im DNS. Es ist eine Liste von erlaubten Hostnames und IPs, die Mails von dieser Domain versenden dürfen.

12.2.2 Domain Keys Identified Mail (DKIM)

Der sendende Mail Server hängt jedem gesendeten Mail eine Signatur im Mail Header an. Diese Signatur ist natürlich wie immer mit dem Private Key signiert, welcher nur dieser Mail Server weiss. Im Header befindet sich auch eine Anleitung, wie die Signatur berechnet werden kann.

Der empfangende Mail Server berechnet dann mit der Anleitung im Header ebenfalls die Signatur. Danach entschlüsselt er die angekommene Signatur mit dem Public Key, welcher in einem TXT Record im DNS steht. Wenn die Signaturen übereinstimmen, kommt das Mail von einem autorisierten Mailserver.

12.2.3 Domain-based Message Authentication, Reporting and Conformance (DMARC)

DMARC ist ebenfalls ein TXT Record im DNS. Darin steht, wie ein empfangender Mail Server damit umgehen soll, wenn ein Mail nicht compliant ist mit SPF und DKIM.

13 Authentication & Federation

13.1 Authentication Protocols

13.1.1 Phasen

Es gibt 4 Phasen beim Einloggen:

- Identification: der User gibt an wer er ist, z.B. mit Username
- Authentication: der User beweist, dass er der ist, wer er angibt zu sein, z.B. mit Passwort
- Authorization: das System prüft, ob der User berechtigt ist, das zu tun was er möchte
- Accounting: Aktionen werden verrechnet

13.1.2 Authentication Schemes

Basierend auf einem IETF-Survey gibt es folgende Authentication Schemes:

Summary: Vulnerability Matrix

Attack	A1 Basic Authentication	A2 One Time Passwords	A3 Challenge / Response	A4 Anonymous Key Exchange	A5 Zero Knowledge PW Proof	A6 Server Cert + User Auth	A7 Mutual Public Key Auth
Passive Password Sniffing	x						
Offline Brute Force Password Attack	x		x	x			
Active Man-in-the-Middle Attack (Phishing)	x	x	x	x			
Identity Theft on Server	x	x	x	x	x	x	
CA Compromise						x	x

13.1.2.1 Basic Authentication

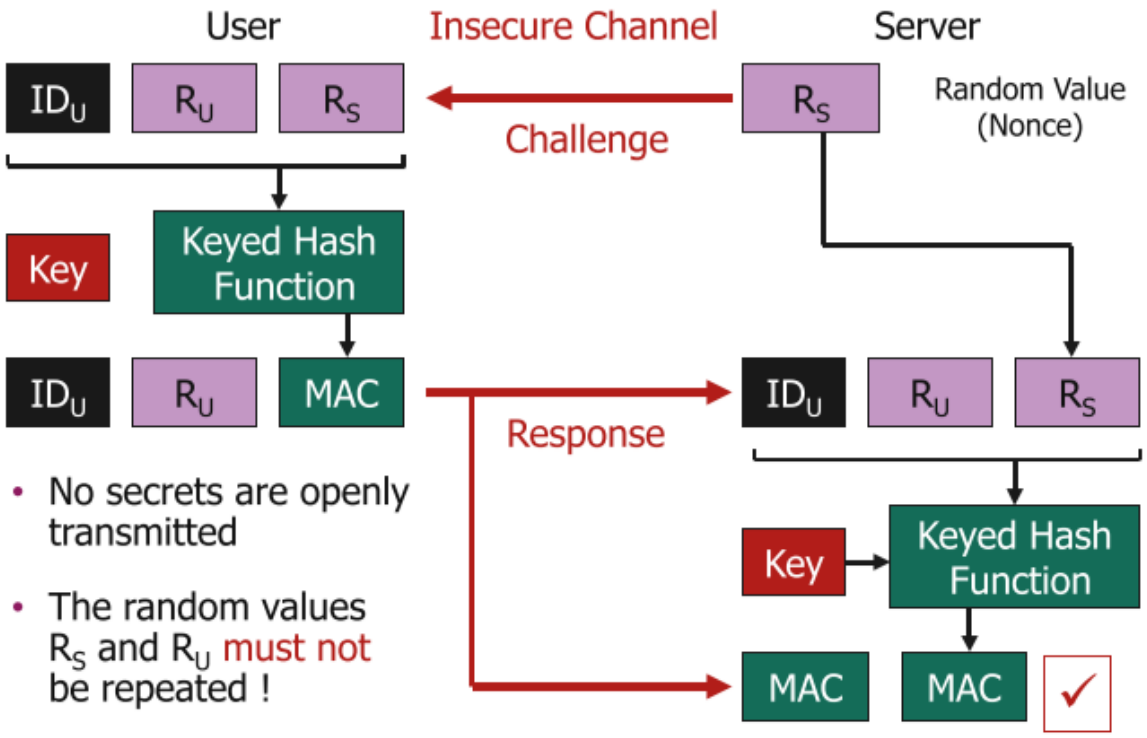
User und Passwort wird im Klartext an den Server übertragen, es darf daher niemals über einen unsicheren Kanal geschehen und die Passwörter müssen sicher sein (Brute-Force).

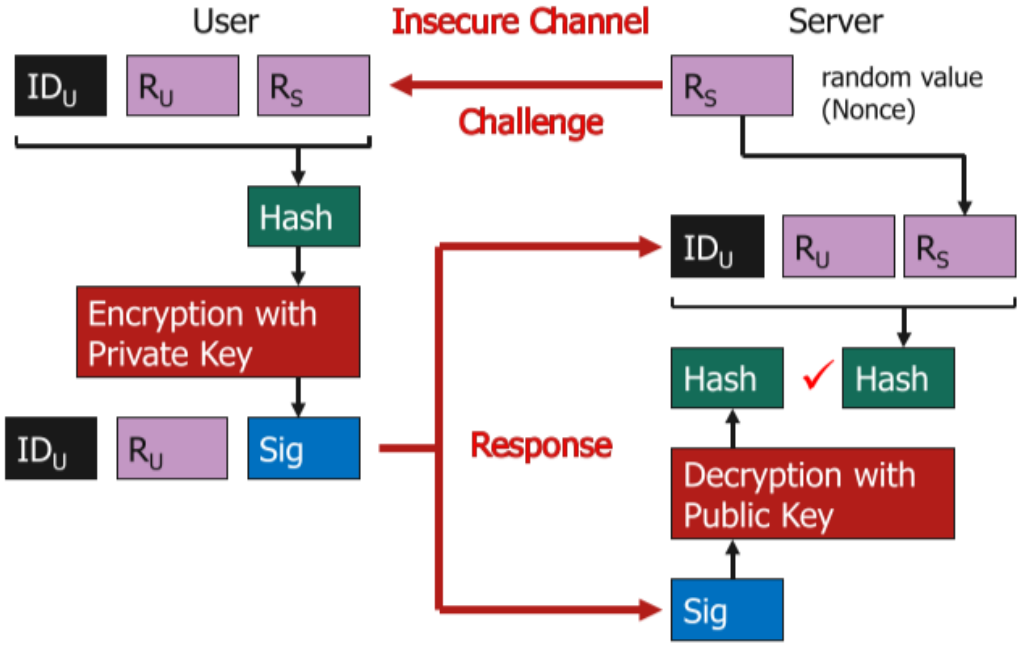
13.1.2.2 One-Time Passwords

Passwort wird im Klartext zum User und dann zum Server übertragen aber kann nur 1x benutzt werden, wenn ein Hacker es auf dem Weg zum User oder zum Server abfängt, kann er nur die aktuelle Session hijacken.

13.1.2.3 Challenge/Response

Der Server sendet dem Client eine Challenge. Um sie zu lösen, braucht der Client das Passwort, die Lösung schickt er als Response zurück zum Server. Somit wird das Passwort selbst nicht übertragen, der Client kann aber trotzdem beweisen, dass er es kennt. Brute-Force und MITM ist immer noch möglich, daher wird es ebenfalls nur über sichere Kanäle gemacht.



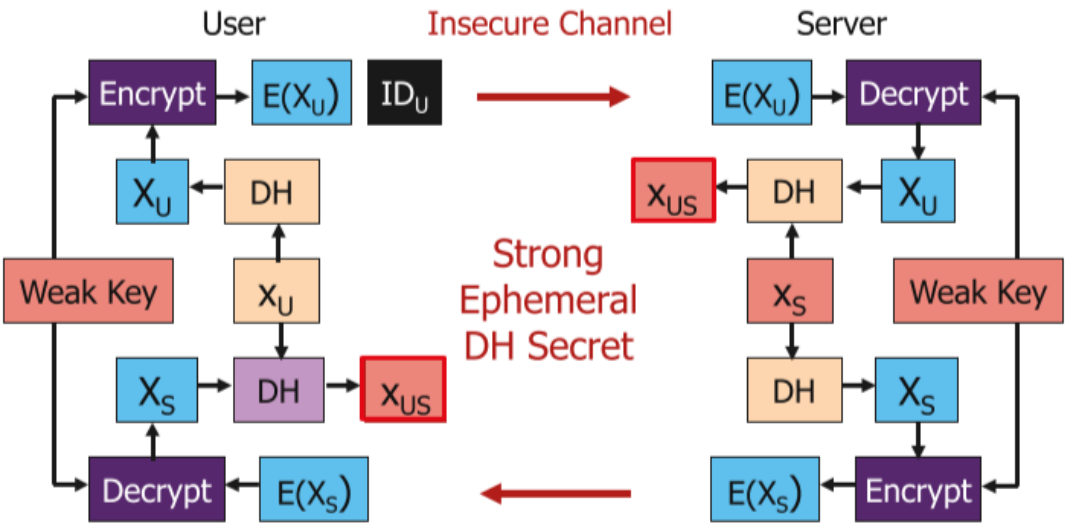


13.1.2.4 Anonymous Key exchange

Dabei machen die beiden Partner zuerst mit z.B. Diffie-Hellman einen Shared Key ab. Mit diesem Shared Key können dann z.B. die Kommunikationen von Basic Auth oder Challenge/Response verschlüsselt werden.

13.1.2.5 Zero-Knowledge Password Proofs

Prover muss dem Verifier beweisen, dass es das Passwort kennt, ohne es zu senden. Der Verifier stellt dem Prover eine Aufgabe mit zwei Möglichkeiten. Der Prover muss diese z.B. 10000x richtig beantworten, dann ist es kein Glück mehr und der Prover weiss wie man die Aufgabe löst, ohne zu sagen wie er die Aufgabe löst.

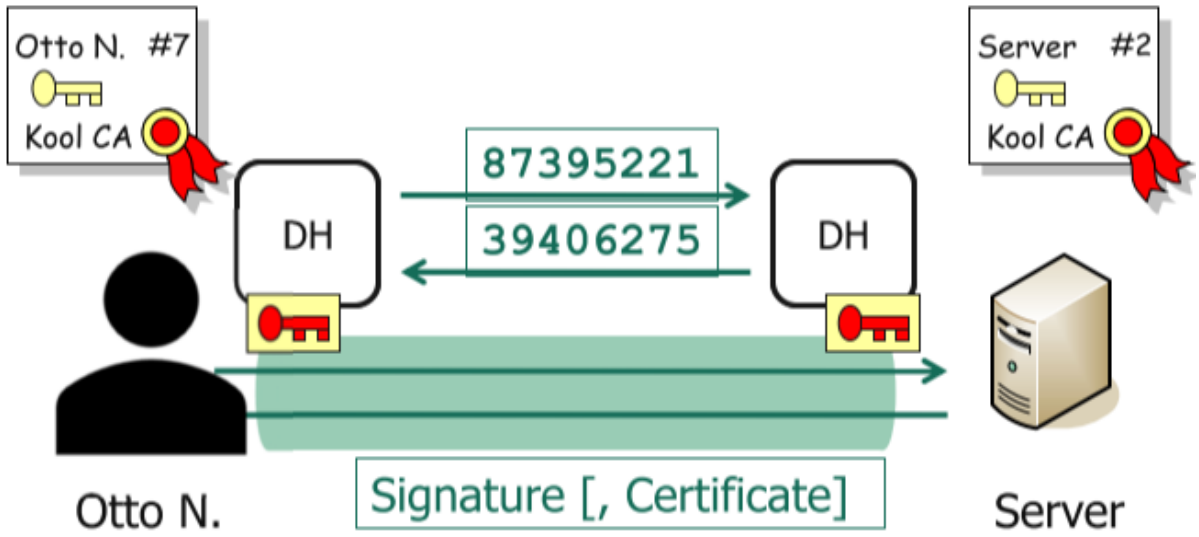


13.1.2.6 Server Certificates plus User Authentication

Zuerst schickt der Server dem Client sein Zertifikat (Public Key). Dann schickt der Client dem Server eine Challenge, welche der Server mit dem Private Key verschlüsseln und zurückschicken muss. Der Client kann dann die Antwort entschlüsseln mit dem Public Key und wenn es übereinstimmt mit der gesendeten Challenge, sendet er das Passwort/One-Time-Passwort/Response verschlüsselt mit dem Public Key an den Server. So werden MITM-Angriffe unmöglich.

13.1.2.7 Mutual Public Key Authentication

Beide Seiten signieren ihren Diffie Hellman Teil mit ihrem Private Key, die Signatur kann von beiden mit dem Public Key des anderen verifiziert werden.

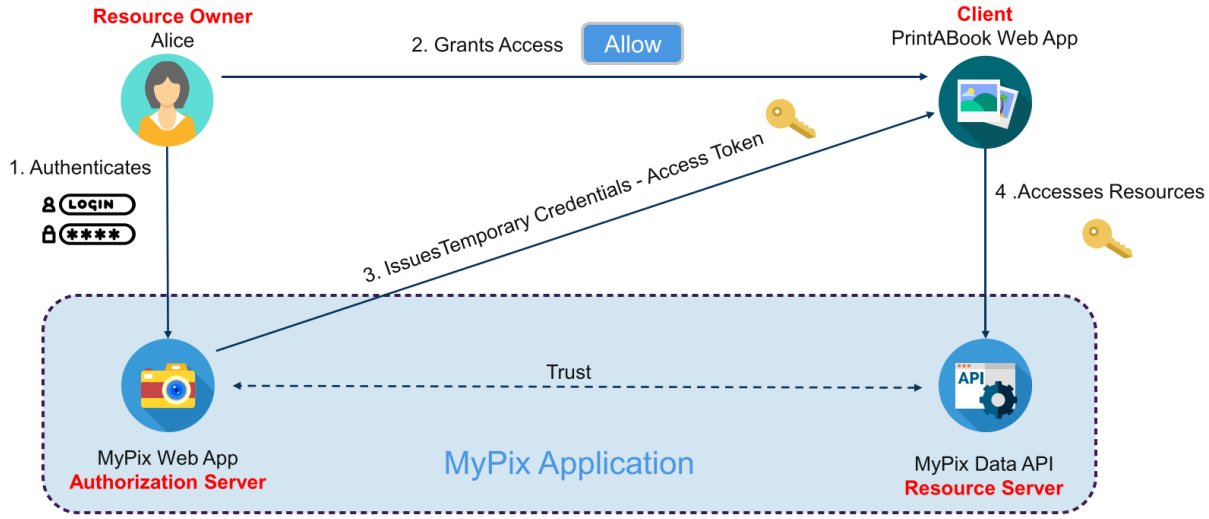


13.2 Federation

Bei einer Federation einigen sich Organisationen auf ein Set von Standards und Rules, damit man sich mit dem gleichen Account auf die Dienste aller Organisationen einloggen kann. Ein Beispiel dafür ist Switch eduID.

13.2.1 OAuth 2.0

OAuth stellt access delegation für Websites zur Verfügung, es ist kein Authentication Protokoll. Es definiert einen Prozess, um Zugriff auf private Ressourcen für eine 3rd-Party Website zu geben. Es hat keinen eingebauten Schutz und muss daher über TLS laufen.



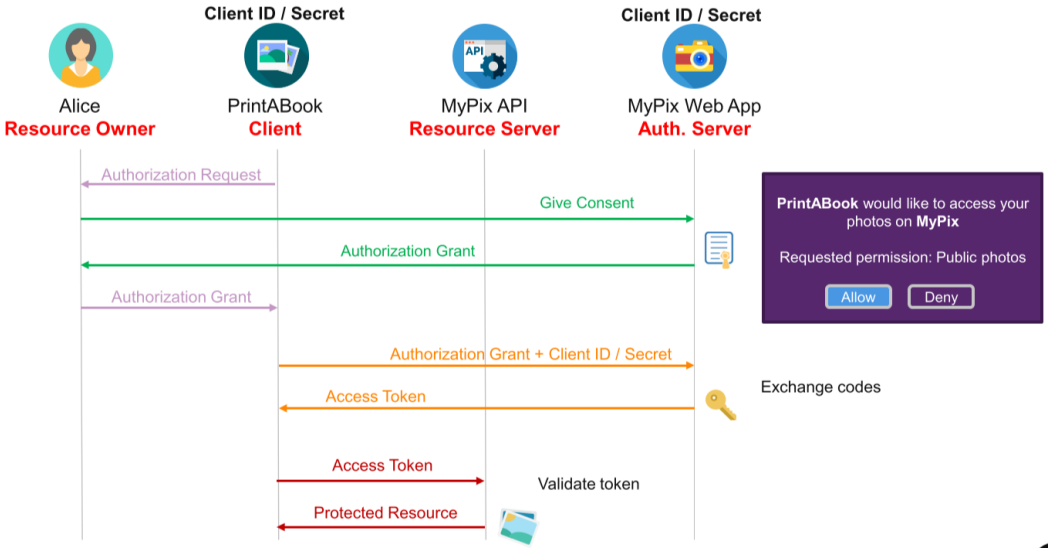
Begriffe:

- Resource Owner: Der User, der die Ressourcen besitzt, in diesem Beispiel besitzt er die Bilder auf MyPix
- Client: Die 3rd-Party Applikation, welche auf die geschützten Bilder zugreifen möchte, es gibt 2 Arten:
    - **Confidential Client**: kann ein Secret behalten, braucht ein Backend, das auf die API des Resource Servers zugreift
    - **Public Client**: kann kein Secret behalten, greift übers Frontend auf die API des Resource Servers zu, z.B. Single page oder Mobile application
- Resource Server: Der Server welcher die geschützten Ressourcen hostet
- Authorization Server: der Server, welcher den User authentifiziert und Access Tokens ausgibt

13.2.2 Grant Types

- Authorization Code + PKCE: Empfohlen
- Client Credentials Flow: Funktioniert nur für Machine-to-machine Kommunikation ohne Enduser
- Authorization Code: Nicht empfohlen
- Implicit Flow: Deprecated
- Resource Owner Flow: Deprecated

13.2.3 Authorization Code



Der Resource Owner sendet folgendes an den Authorization Server:

```
https://mypix.com/authorize?response_type=code&client_id=6779ef20e7581&
    redirect_uri=https://printabook.com/callback&scope=public&state=Ax3B0fqK
```

- **https://mypix.com/authorize**: the authorization API of the auth. server
- **response_type**: specifies that your application is requesting an authorization code grant
- **client_id**: the client ID of PrintABook (how the auth. server identifies the client)
- **redirect_uri**: where the service redirects the user-agent after an authorization code is granted
- **scope**: specifies the level of access/scope that the client is requesting
- **state**: randomly generated value to prevent/detect CSRF attacks (recommended)

Wenn der Resource Owner den Authorization Grant gegeben hat, sendet der Authorization Server dem Client folgendes. Die URL sollte immer validiert werden, die URL wird dazu bei der Client Registration angegeben.

```
Location: https://printabook.com/callback?authorization_code=Ax3lfsx492Aldv
          &state=Ax3B0fqK
```

- **https://printabook.com/callback_grant**: API of the client that accepts the auth code
- **authorization_code**: Authorization code that can be used to request an access token
- **state**: reflected value received in the authorization request. Must match the previous value (CSRF prevention)

Nun fragt der Client einen Access Token an:

```
https://mypix.com/token?client_id=6779ef20e7581&
    client_secret=xB837sdfoBqq2842Bd&grant_type=authorization_code&
    code=Ax3lkdfaB33jfsx492Aldv&redirect_uri=https://printabook.com/callback
```

https://mypix.com/token: access token API of the auth. server

client_id: the application's client ID (how the auth. server identifies the application)

client_secret: “password” of the application to authenticate itself to the auth. server

grant_type: authorization code flow that is used

code: authorization grant issued to the client

redirect_uri: must be identical to the previous redirect URI

Wenn die Anforderungen erfüllt sind wird ein Token ausgegeben:

- Client Credentials sind korrekt
- Anfrage kommt vom gleichen Client wie für den die Credentials ausgestellt wurden
- Der Code wurde nicht schon einmal verwendet
- Der Code ist noch nicht abgelaufen

Das JSON-Format des Tokens ist vorgegeben, der Token selbst aber nicht. Der Token hat einen Scope und ist optional zeitlich begrenzt.


```
{"access_token":"aSxl2bw958djkjcvqiowefasdf234dfDFWdf2","token_type":"bearer",
"expires_in":2592000,"refresh_token":"2xk02dkjfoFDBjf29865uhSdhbodfasdfF",
"scope":"public","uid":923,"info":{"name":"Alice","email":"alice@gmail.com"}}
```


The `access_token` eventually grants access to the user's data (i.e. pictures)

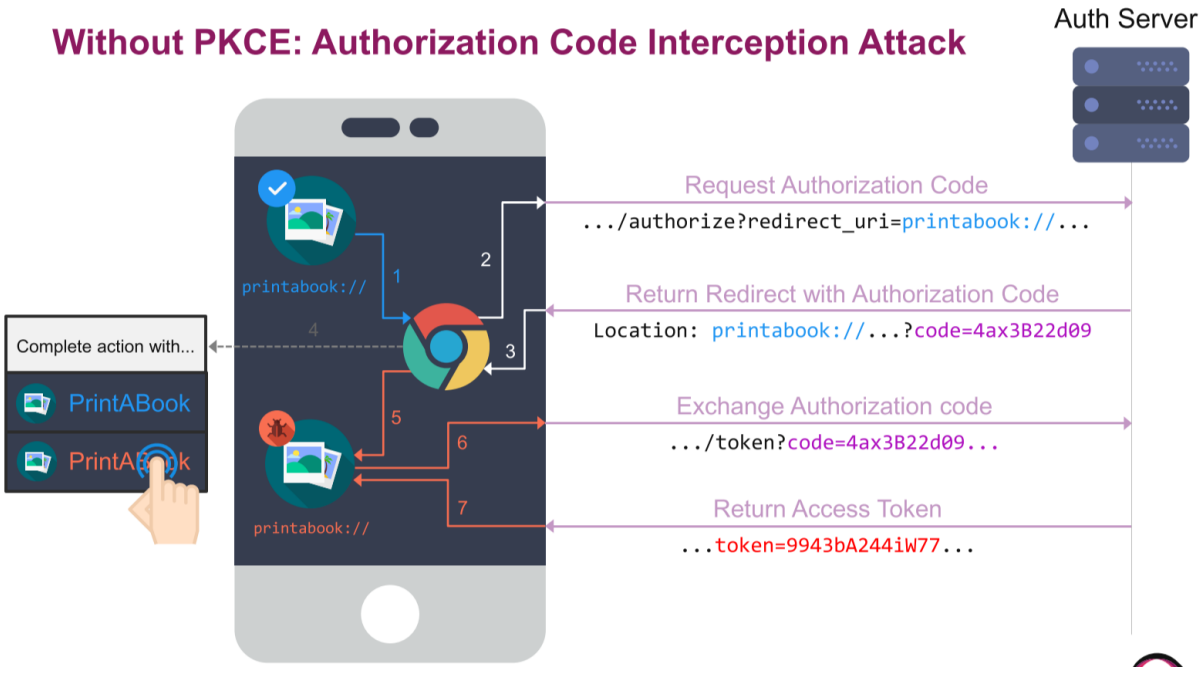
The `token_type` specifies how the token is to be used

The `refresh_token` (if issued) can be used to fetch a new `access_token`

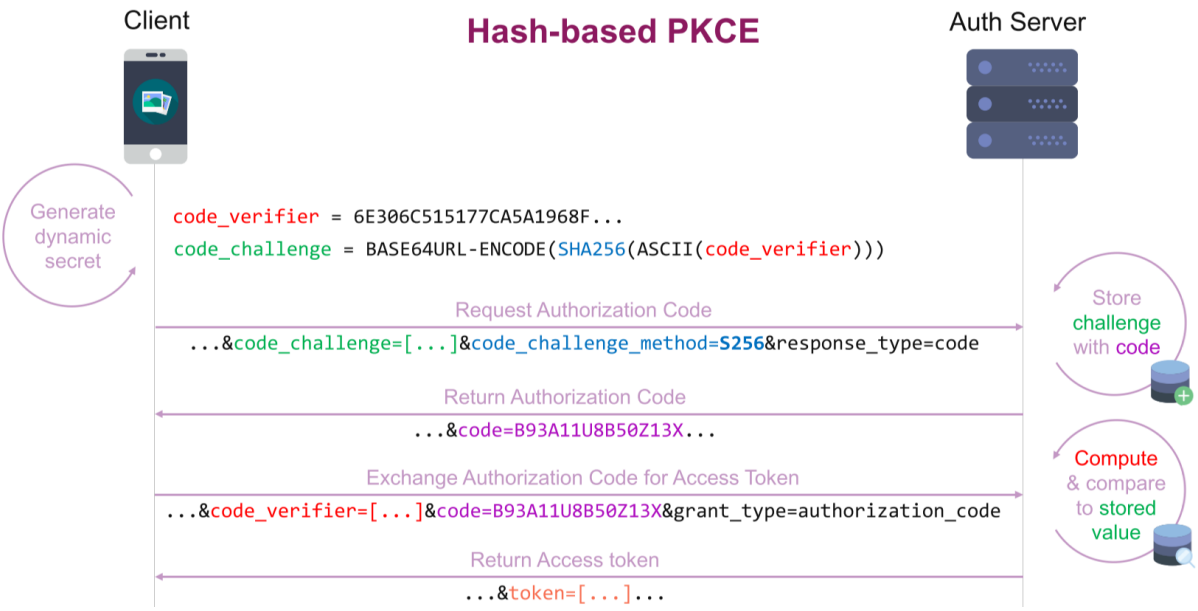
Refresh Tokens erlauben es, einen neuen Access Token zu holen. Das erlaubt Access Tokens, welche nur 5 – 10 Minuten gültig sind. Das beschränkt die Zeit in der ein Angreifer auf Ressourcen zugreifen kann, falls es ihm gelingt, einen Access Token zu stehlen.

13.2.4 Authorization Code Flow with PKCE

PKCE wurde ursprünglich für Smartphones entwickelt, welche Client Secrets nicht sicher speichern können und Custom URL Schemes wie printabook:// unterstützen. Sicherheitslücke:

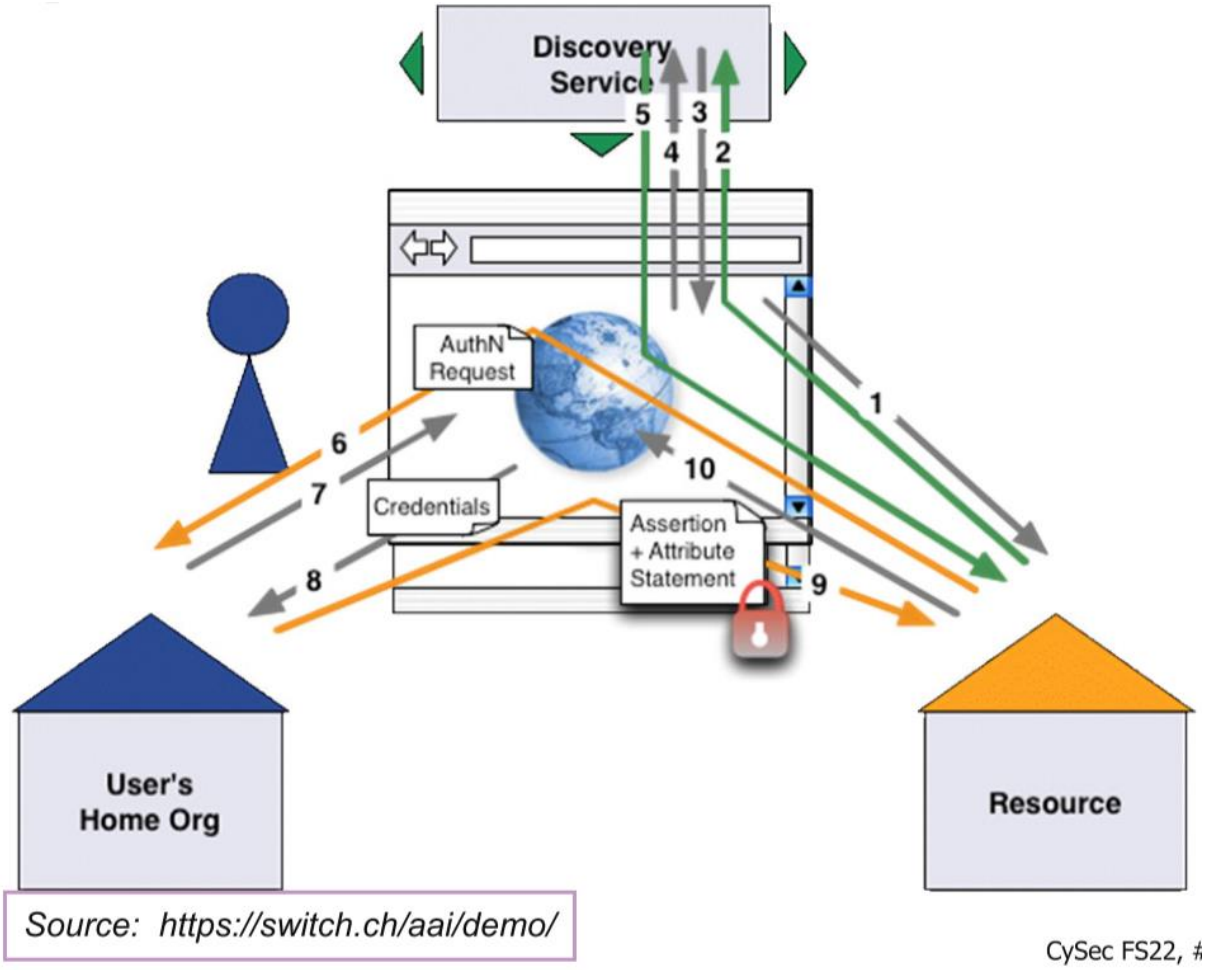


Um das zu verhindern, wird ein dynamisches Secret (code_verifier) vom Client generiert.



13.3 SHIBBOLETH

Shibboleth ist eine Open Source Middleware, welche von Unis eingesetzt wird. Es basiert auf SAML.



```
<samlp:AuthnRequest
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="aaf23196-1773-2113-474a-fe114412ab72"
  Version="2.0"
  IssueInstant="2004-12-05T09:21:59Z"
  AssertionConsumerServiceIndex="0"
  AttributeConsumingServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"/>
</samlp:AuthnRequest>
```

13.4 Extensible Markup Language (XML) Security

XML Signature stellen Integrität, Message Authentication und Signatur sicher. Ausserdem können Daten verschlüsselt werden.

13.5 Secure Assertion Markup Language (SAML)

Das sind XML-codierte Behauptungen über Authentifizierung, Attribute und Autorisierung. Damit lässt sich Single-Sign On realisieren. Die Teile sind:

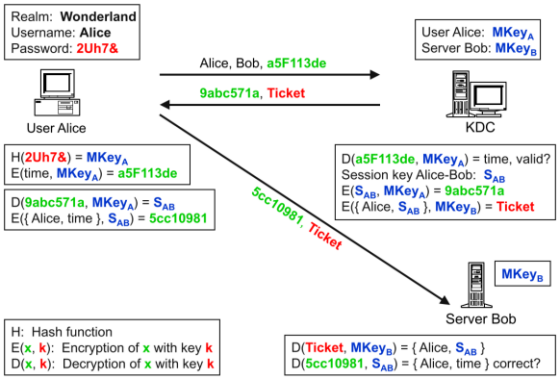
- Behauptungen und Protokolle
- Bindings
- Profile
- Metadaten
- Authentication Context
- Conformance Anforderungen
- Sicherheits- und Privacy-Anforderungen
- Glossar

Es wird über http POST Requests oder SOAP-Messages verwendet.

13.6 Kerberos

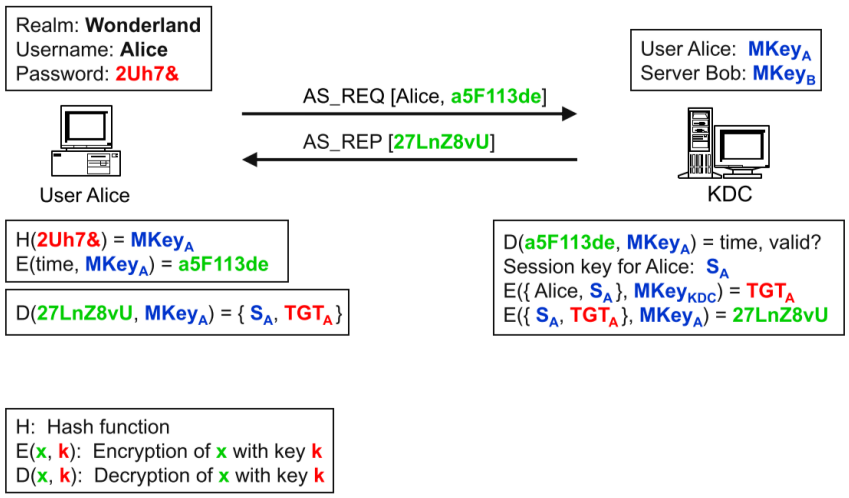
Kerberos wurde am MIT entwickelt, weil Studenten mit Sniffing Root Passwörter im Netzwerk abgreifen konnten. Es wird in MS AD verwendet. (KDC=Key Distribution Center)

13.6.1 Simple Darstellung der Funktionsweise

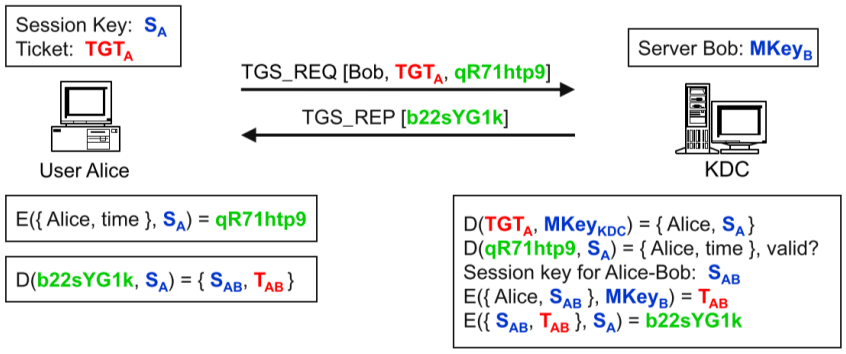


13.6.2 Ausführlichere Darstellung

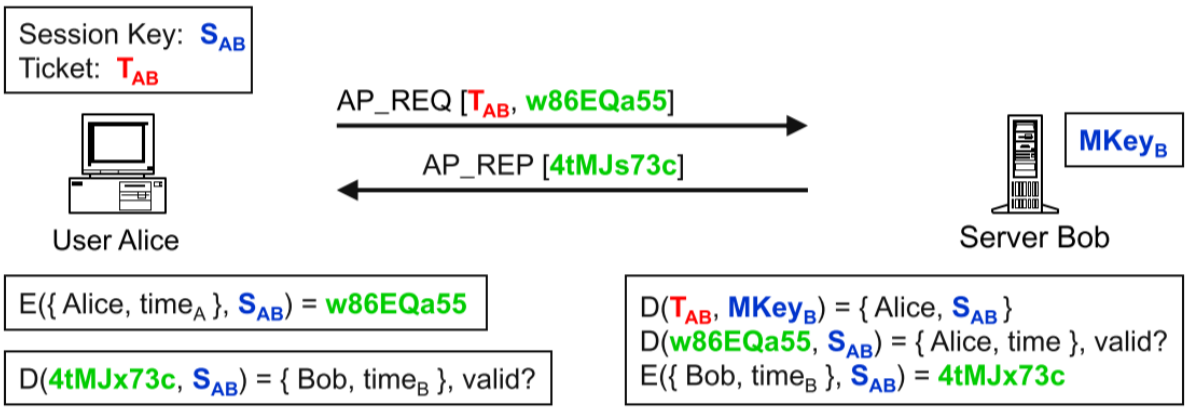
In der Praxis wird zuerst ein Ticket-Granting-Ticket (TGT) ausgestellt:



Mit dem TGT kann der Client dann weitere Tickets bestellen:



Mit dem Ticket kann dann auf den Server zugegriffen werden:



13.6.3 Skalierung

KDCs können auf Slave KDCs repliziert werden, indem die DB mit den MKeys und KDC MKey repliziert wird.

Realms können auch miteinander verbunden werden. Dazu fragt der User zuerst beim eigenen KDC für ein Ticket für den KDC in der anderen Realm an. Mit diesem Ticket fragt er dann beim fremden KDC ein Ticket für eine Ressource auf einem fremden Server an.