

# Netzwerksicherheit

<https://de.wikipedia.org/wiki/Netzwerksicherheit>

## Sicherheitsaspekte:

- Verfügbarkeit ( = Stabilität, Robustheit)
- Einbruchssicherheit
- Datensicherheit (Backups)

## Verfügbarkeit

- Verfügbarkeit <https://de.wikipedia.org/wiki/Verfügbarkeit>
- <https://de.wikipedia.org/wiki/Hochverfügbarkeit>
- SLA <https://de.wikipedia.org/wiki/Service-Level-Agreement>
- CAP-Theorem <https://de.wikipedia.org/wiki/CAP-Theorem>

Stabile Betriebssysteme erhöhen die Verfügbarkeit.

## Redundanz

Redundant heißt "überflüssig" bzw. "mehrfach ausgelegt". Redundanz meint also das Gegenteil des Begriffes des "single point of failure". Oft ist es nicht sofort ersichtlich welche Komponenten eines Systems diese "single points" sein könnten. Beispiele: Stromversorgung, Netzkabel, (Micro-)Services nur einfach vorhanden, Datenbank ..

Rechenzentren tragen dem Rechnung durch Akkus und Notstromaggregate. Es muß genau analysiert werden, was alles redundant ausgelegt werden muß, um hochverfügbar zu sein. Manche single points of failure müssen je nach (finanziellen) Ressourcen in Kauf genommen werden.

Einige Konzepte:

- DNS round robin (ein Hostname -> mehrere IPs): Bsp: google MX, at.pool.ntp.org
- ha-proxy (mehrere Backend-Server "hinter" einer IP)
- anycast (eine IP, viele Server) (Netzwerk-Schnickschnack ist beteiligt)

## Sicherheit vor Einbruch und Datendiebstahl

Problematik: Hin und wieder knackt ein Mathematiker einen crypto-Algorithmus. Immer wieder werden Softwarefehler gefunden, die Einbrüche erlauben.

- regelmäßiges Patchen aller (!) beteiligten Software-Komponenten
- Vulnerability Datenbank CVE: <https://www.cve.org/>
- Inventory über Patch-Stände (eher junges Konzept, noch in den Kinderschuhen) in Kombination mit Vulnerability-Datenbanken.
- Firewalls
  - Layer 3 (Paketfilter, stateful) (geblockte Ports)
  - Layer 7 (deep inspection). Sog. Application-Level Firewalls, z.B. WAF (Web Application Firewall).

Problematik hier: SSL ist End-to-End verschlüsselt

- "Intrusion Avoidance" .. mit KI
- schlecht konfigurierte Firewalls, oft schwierig zu debuggen

## Intrusion Detection

- KI wird eingesetzt, um Anomalien zu erkennen. Zentrales Logging Voraussetzung. 2-factor authentication, Ausblick auf Layer 8 Unterscheidung Authentication / Authorization
- merke: Georg (authenticated) -> "is not authorized" (permission denied)

## Kryptographie

*Transport Layer Security (TLS)* ist das am Meisten verbreitete Crypto-Protokoll (*xxx over TLS*), z.B.:

- https: http over TLS
- smtps, imaps, ... usw

<https://de.wikipedia.org/wiki/Kryptographie>

## Ziele

- Vertraulichkeit/Zugriffsschutz (kann von unbefugt nicht entschlüsselt werden)
- Integrität/Änderungsschutz (es kann bewiesen werden, dass Nachricht unverändert ist)
- Authentizität/Fälschungsschutz (es kann bewiesen werden, dass die Gegenstelle tatsächlich die ist, die sie vorgibt zu sein)
- Verbindlichkeit/Nichtabstreitbarkeit (es kann bewiesen werden, daß die Nachricht von niemand anders kommt)

## Vier Teilkomponenten der Verschlüsselung

1. asymmetrischer Schlüsselaustausch (A und B machen sich einen geheimen Schlüssel über ein unsicheres Medium aus).
2. symmetrische Verschlüsselung für große Datenmengen (verwenden den unter Punkt 1. ausgehandelten Schlüssel).
3. Zertifikate zur Authentisierung der Gegenstelle
4. Hashfunktionen zur Sicherung der Nachrichten-Integrität

## Ciphersuite

Eine Ciphersuite ist eine Kombination aus allen 4 oben genannten Komponenten, diese wird beim sog. "TLS-Handshake" zwischen Browser und Server ausgemacht. `openssl ciphers` zum Auflisten möglicher Kombinationen.

[https://de.wikipedia.org/wiki/Cipher\\_Suite](https://de.wikipedia.org/wiki/Cipher_Suite)

- Schlüsselaustausch, z. B.: RSA, DH (Diffie-Helman Verfahren)
- Authentifizierung, bspw.: RSA, DSA (auch ECDSA)
- Verschlüsselung zB DES, AES
- Hashfunktionen (MD5, SHA)

## Allgemeines

Kerckhoffs Prinzip: Die Sicherheit eines Systems darf nicht von der Geheimhaltung der Algorithmen abhängen (Security by Obscurity), sondern nur von der Geheimhaltung eines Schlüssels. Stichwort: Reverse-Engineering

Man In The Middle

- entweder "rein passiv" oder
- aktiv (ändert Pakete oder gibt sich als einer der 2 Partner aus)
- aktiv relevant bei DNS-Spoofing und Cache-Poisoning

Symmetrische Kryptographie (AES, DES): Es gibt **einen** Schlüssel, der zum Verschlüsseln **und** Entschlüsseln verwendet wird.

Vorteil: Schnelle Algorithmen, Stromchiffren, dieser wird über ein asymmetrisches Verfahren ausgehandelt.

Nachteil: Beide Verbindungspartner müssen den symmetrischen Schlüssel kennen, jedoch darf ihn *sonst niemand* wissen.

Asymmetrische Kryptographie (RSA, DH): Jeder Partner hat ein Schlüssel**paar**, davon ist einer öffentlich, einer privat.

- Vorteile
  - public/private Keys
  - passiver Angreifer wird den gemeinsamen Schlüssel nicht kennen, auch wenn er den ganzen Handshake mitbekommen hat.
- Nachteile
  - hoher Rechenaufwand
  - nicht geeignet zur Verschlüsselung großer Datenmengen
- Zweck:
  - Asymmetrische Crypto wird verwendet, um gemeinsam mit dem Partner auf einen symmetrischen Schlüssel zu kommen, den sonst niemand weiß.

Tools

- OpenSSL, kennt die allermeisten Algorithmen
- sha256sum, md5sum, ... Hashfunktionen über Files

## Zertifikate

Sind so etwas wie ein "digitaler Ausweis", enthalten einige digital signierte Informationen:

- issuer (Wer ist der Aussteller)
- subject (also "Betreff" des Zertifikates)
- Gültigkeitszeitraum
- erlaubte Verwendungszwecke

Es gibt etliche weltweit anerkannte "Root CA's" (Certificate Authorities). Bei diesen Zertifikaten ist *immer* der Issuer und das Subject identisch, das heißt maW. "self-signed". Diese Root-CA's sind in jedem Browser eingebaut. Diese haben oft eine Gültigkeitsdauer von einigen Dutzend Jahren.

Die Root CA's erstellen meistens weitere CA Zertifikate, diese CA's werden "Sub-CA's" genannt.

Diese Sub-CA's stellen Server- und Endbenutzer-Zertifikate aus.

Ein Zertifikat ist *immer* nur mit einem dazugehörigen *key* verwendbar (anzeigen und auslesen kann man den öffentlichen Teil aber immer).

## Aufgaben einer CA

- Erstellung von Zertifikaten
  - Server Certs
  - Client Certs
  - CA Certs
- Pflegen von CRL's (Certificate Revocation List) und anbieten zum Download
  - anderes Protokoll: OCSP

Was ist eine CRL?: Eine Liste von Zertifikaten, welche verlorengegangen sind. Verloren heißt hier: Ich weiß (oder vermute), dass das Zertifikat und der dazugehörige Schlüssel in bössartige Hände geraten sind. Das melde ich der CA, und diese nimmt das Zertifikat in die CRL heinein. Jeder Browser der CRLs kontrolliert, weiß anhand dessen, dass dieses Cert dadurch ungültig geworden ist und verhindert das Verwenden einer Website mit diesem Zertifikat.