

# ISF HS 2019

Victor Fernández  
Pavaskar Parameswaran

Dezember 2019

## Vorwort

Diese Zusammenfassung entstand in einer Gruppe während der Lernphase des HS 2019. Alle Fragen aus der Stoffabgrenzung tragen eine [blaue Farbe](#) und stehen als Unterkapitel. Das Dokument ist Open Source und jeder der möchte und signifikant beiträgt, darf sich als Autor anhängen. Die Source ist [dieses GitHub-Repo](#)<sup>1</sup>. Dies ist mein erstes L<sup>A</sup>T<sub>E</sub>X-Dokument überhaupt. Nichts desto trotz wurde auf eine klare Strukturierung und Lesbarkeit des Dokumentes Wert gelegt.

## Inhaltsverzeichnis

<b>I</b>	<b>Einführung (SW 01)</b>	<b>3</b>
1	Einführung	3
<b>II</b>	<b>Kryptographie (SW 02-04)</b>	<b>5</b>
2	Symmetrische Kryptographie	5
3	Asymmetrische Kryptographie	11
4	Zertifikate und SSL-TLS	13
<b>III</b>	<b>Angriffe (SW 05-06)</b>	<b>15</b>
5	Angriffe auf Webanwendungen	15
6	Angriffe auf Protokollebene	18
<b>IV</b>	<b>Management (SW 07-09)</b>	<b>24</b>
7	Standards & Frameworks, ISMS	24
8	Risiko-Management und IT-Grundschutz	25
9	Awariness	26
<b>V</b>	<b>Access Control (SW 10)</b>	<b>26</b>
10	Access Control	26
<b>VI</b>	<b>Multi-Party-Computation (SW 11)</b>	<b>26</b>

---

<sup>1</sup>[https://github.com/vigi86/HSLU\\_Zusammenfassungen/tree/master/ISF\\_HS19](https://github.com/vigi86/HSLU_Zusammenfassungen/tree/master/ISF_HS19)

11 Cryptographic Protocols	26
12 Secret Sharing	27
13 Zero Knowledge Proof	27
 VII Quantum (SW 12)	 27
14 Quantum Computing and Quantum Cryptography	27
 VIII WAF, Federations (SW 13)	 27
15 Firewalls	27
16 Federations	28
 IX Talks (SW 14)	 28
17 Malware	28
18 WAF	28

## Teil I

# Einführung (SW 01)

## 1 Einführung

### Einführung in das Thema „Management von Informationssicherheit“

**Daten, Information und Wissen** Information ist die Verknüpfung von Daten in Form von Zahlen, Worten und Fakten zu interpretierbaren Zusammenhängen. Durch die Vernetzung von Informationen entsteht Wissen, das zunächst personenbezogen ist.

**Missbrauch** Informationen müssen vor Missbrauch geschützt werden

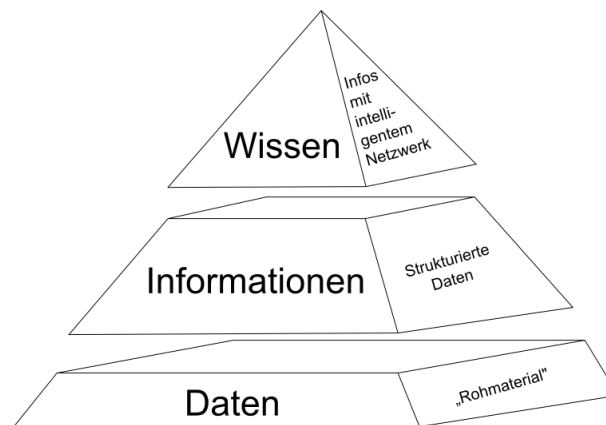


Abbildung 1: Wissenspyramide (Wikipedia)

### Motivation / Bedrohungen

**Was gefährdet die Informationen?** Welche Gefährdungen/Bedrohungen gibt es?

- Nicht vorsätzliche (zufällige) Gefährdungen/Bedrohungen
  - Naturgewalten (Blitz, Hagel, Unwetter, Erdbeben, Hochwasser, etc.)
  - Ausfall von Strom oder Telekommunikation
  - Technische Pannen, z.B. Fehler von Hard- und/oder Software
  - Bedienerfehler / Fahrlässigkeit der Mitarbeitenden
- Vorsätzliche Gefährdungen/Bedrohungen
  - Bösartiger Code (Viren, Würmer, Trojaner, etc.)
  - Informationsdiebstahl
  - Angriffe (von Skript-Kiddies bis Hacker)
  - Wirtschaftsspionage („was die Konkurrenz wissen möchte“)
  - Missbrauch der IT-Infrastruktur

### Grundbegriffe

**Zutritts-, Zugangs-, Zugriffskontrolle**

- **Zutrittskontrolle:** Schutz des physischen Systems (Bsp. Serverraum)
- **Zugangskontrolle:** Schutz des logischen Systems (Bsp. Betriebssystem)
- **Zugriffskontrolle:** Daten-bezogen; Schutz der Operationen (Bsp. Dateisystem)

### Kontrollfragen SW 01

**Wie lauten die Schutzziele der Informationssicherheit? Nennen Sie konkrete Beispiele.**

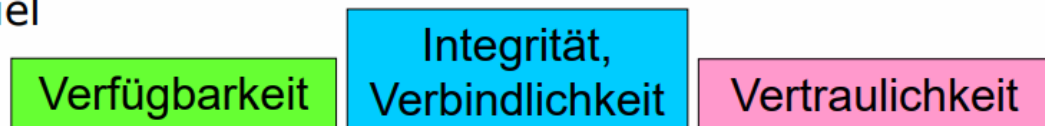
- **Verfügbarkeit:** Zur gewünschten Zeit kann vom Benutzer auf die Daten zugegriffen werden und der Dienst funktioniert (Ausfallquote)
- **Integrität:** Gewährleistung das Daten nicht unautorisiert oder zufällig manipuliert werden können. (Datensicherheit)

- **Verbindlichkeit:** Handlung kann eindeutig einer Person zugeordnet werden und von dieser auch nicht geleugnet werden.
- **Vertraulichkeit:** Informationen können nicht von unautorisierten Personen, Instanzen und/oder Prozessen eingesehen werden.

### Beispiele

- Daten und Informationen (Kundendaten, Rechnungen, Marketingdaten usw.) können nicht vom PC abgerufen werden.
- Durchgängiges Funktionieren von IT Systemen, sowie eine Vollständigkeit und Richtigkeit von Daten und Informationen. Verhindern von nicht genehmigten Veränderungen an wichtigen Informationen.
- Schutz vor Verrat von Informationen oder vertraulichen Daten. Mit Authentizität ist gewährleistet, dass es sich tatsächlich um eine autorisierte Person (Identitätsnachweis) handelt.

## Grundziel



## Entdeckung

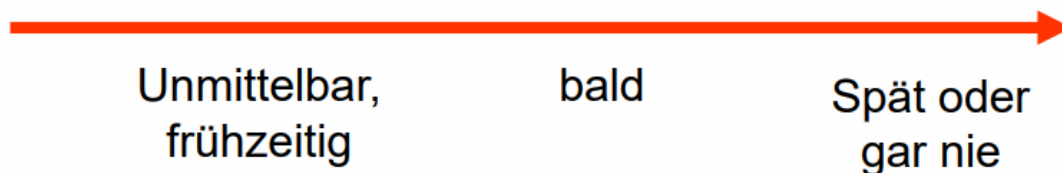


Abbildung 2: Zeitpunkt Entdeckung eines Grundziel-Verlustes

### Erklären Sie die Zusammenhänge zwischen Risiko, Sicherheit, Eintretenshäufigkeit, Schadenshöhe, Restrisiko?

- Risiko: Gefahr, Problem welches entstehen kann.
- Sicherheit: Schutz vor Risiko
- Eintretenshäufigkeit: Wahrscheinlichkeit, dass Risiko eintritt
- Restrisiko: Sicherheit deckt einen Teil des Risikos ab, jedoch nur gewisses Budget, somit bleibt Restrisiko, z.B. Erdbeben in San Francisco oder Istanbul

**Zusammenhang** Durch Sicherheitsmassnahmen kann man die Eintretenshäufigkeit von einem Risiko mindern und somit auch den Schaden reduzieren, zusätzlich wird auch das Restrisiko kleiner oder kann sogar ausgeschlossen werden.

### Was ist eine Risikoanalyse? Wozu dient sie?

1. Risiken zu erkennen
2. Risiken analysieren
3. Risikobewertung
4. Nächster Schritt wäre das Risiko zu mindern mit den Erkenntnissen aus den ersten 3 Punkten.
5. Massnahmen ergreifen und umsetzen.

### Welche Möglichkeiten zur Behandlung («Mitigation») stehen zur Verfügung? Ordnen Sie diese in der Reihenfolge, wie man sie typischerweise anwendet. Geben Sie ein kurzes Beispiel oder eine Erläuterung

- Mitigation = Verminderung
- Priorisieren: welches Risiko muss ich zuerst behandeln
- Vermindern: Massnahmen ergreifen, um Eintretenshäufigkeit und Schadensausmass zu vermindern
- Vermeiden: Massnahmen oder Wege einleiten um das Risiko auszuschliessen
- Akzeptieren: Ähnlich wie Ignorieren und das Risiko einfach hinnehmen

**Welche Gefährdungen und Bedrohungen kennen Sie? Gliedern Sie diese in verschiedene Kategorien.**

- Bruteforce, Phishing, DDoS, Injections, Spoofing, frustrierte Mitarbeiter, Anonymous, NSA, Naturgewalt, Hacker, Wirtschaftsspionage etc.

**Stellen Sie Wissen – Information – Daten in eine Beziehung. Worauf bezieht sich die IT-Sicherheit? Was verstehen Sie unter integraler / holistischer Sicherheit?**

- **Informationssicherheit:** Schutz der Information als solche, Medien unabhängig Elektronischer Datenträger
- **IT-Sicherheit:** Schutz der Informationen in ICT-Systemen. (Server/Host)
- **Integrale Sicherheit vs. holistische Sicherheit:** Umfassende Betrachtung aller Sicherheitsaspekte einer Organisation.  
Ziel ist das grösste Niveau der Sicherheit zu erzielen (Gesamtsystem) wichtig ist wie sich die einzelnen Sicherheitskomponenten ergänzen und zusammenspielen.

## Teil II

# Kryptographie (SW 02-04)

## 2 Symmetrische Kryptographie

**Sie verstehen was Steganographie ist**

**Steganographie** Verstecken von Information, z.B. in Bildern oder Audiofiles.

**Sie verstehen was Private-Key-Kryptographie ist, welche Arten von Sicherheit es gibt und welche Angriffsarten auf Verschlüsselung existieren**

**Zeichencodierung** Kodierung (*Encoding*) heisst, einen Wert mit Symbolen eines Zeichensatzes darzustellen. Beispiel:

Dezimalsystem	100
Binärsystem	1100100
Hexadezimalsystem ('hex')	64
ASCII	hello
Base64	aGVsbG8=

**Achtung: Kodierung  $\neq$  Verschlüsselung**

**Symmetrische Verschlüsselung** Bei symmetrischen Verschlüsselungsverfahren gibt es im Gegensatz zu den asymmetrischen Verfahren, **nur einen einzigen Schlüssel**. Dieser Schlüssel ist für die Verschlüsselung, als auch für die Entschlüsselung zuständig.

**Secret Key Verschlüsselung** Secret Key ('Symmetrische') Verschlüsselung wird zwischen zwei Parteien verwendet, welche einen **gemeinsamen Schlüssel** besitzen. Ausserdem wird sie oft verwendet, wenn der gleiche Benutzer ein Dokument verschlüsseln und zu einem späteren Zeitpunkt wieder entschlüsseln muss.

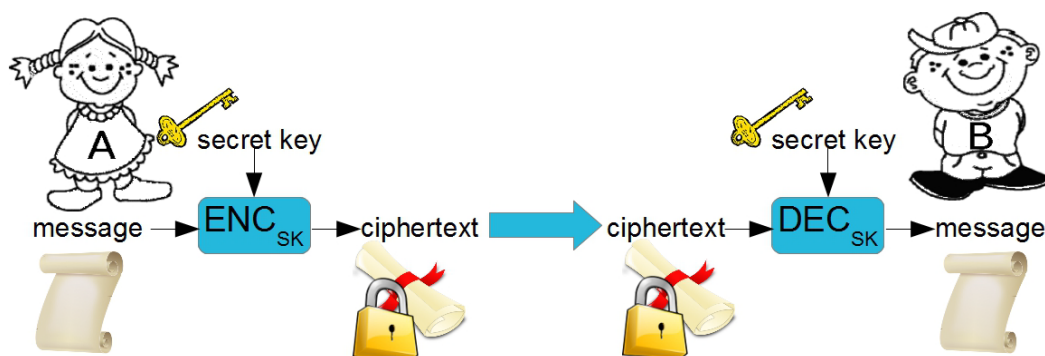


Abbildung 3: Alice verschlüsselt, Bob entschlüsselt mit dem gemeinsamen Schlüssel

**Informationstheoretische Sicherheit** Das Ziel informationstheoretischer Sicherheit ist der Schutz von Daten vor unbefugtem Zugriff während der Übertragung. Im Unterschied zur Kryptographie basiert informationstheoretische Sicherheit nicht auf der Annahme, dass die Rechenleistung eines unberechtigten Empfängers nicht gross genug ist, um die Daten zu decodieren. Vielmehr garantiert informationstheoretische Sicherheit, dass ein unberechtigter Empfänger selbst bei beliebig grosser Rechenleistung nicht in der Lage ist, solcherart geschützte Nachrichten zu decodieren. Mit anderen Worten erhält ein Angreifer durch den Geheimtext keinerlei (zusätzliche) Information über den Klartext. Beispielsweise ist OTP informationstheoretisch sicher.

Formal:  $P(M = m) = P(M = m | C = c)$  **Erklärung der Variablen??**

**Berechenmässige Sicherheit** Der sicheren Übertragung und Aufbewahrung vertraulicher Daten kommt in unserer von Information dominierten Gesellschaft immer grössere Bedeutung zu. Die heute gebräuchlichen Verfahren zur Datenverschlüsselung bieten allerdings nur beschränkte, sogenannte berechenmässige Sicherheit. Das bedeutet, dass diese prinzipiell von einem Angreifer, der über genügend Rechenleistung (zum Beispiel einen, heute noch hypothetischen, Quantencomputer) verfügt, gebrochen werden können.

**Kerckhoff's Prinzip** Der Angreifer kennt den Algorithmus und alle Details des Systems. Nur der Schlüssel ist geheim.

**Angriffsarten** Bei der Sicherheit von modernen Verschlüsselungssystemen wird zwischen den Angriffsmöglichkeiten des Angreifers unterschieden:

- **Ciphertext only attack:** Angreifer erhält nur den zu entschlüsselnden Geheimtext

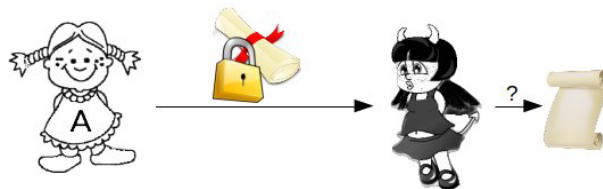


Abbildung 4: Nur Geheimtext

- **Known plaintext attack:** Angreifer erhält zusätzlich andere Klartext-Geheimtext-Paare

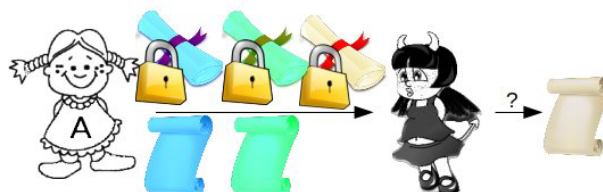


Abbildung 5: Klartext-Geheimtext-Paare

- **Chosen plaintext attack:** Angreifer kann zusätzliche Klartexte wählen, zu denen er auch die Geheimtexte erhält

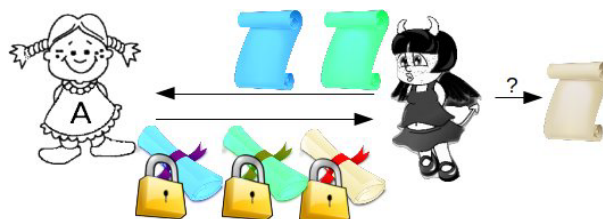


Abbildung 6: Klartexte und Geheimtexte

## Sie können „klassische“ symmetrische Verschlüsselungsverfahren wie Caesar cipher, Vigenère cipher, one-time pad anwenden und verstehen die Vor- und Nachteile bzw. Schwachstellen dieser Verfahren

**Caesar cipher** Caesar-Verschlüsselung ist ein einfaches symmetrisches Verschlüsselungsverfahren, das auf der monographischen und monoalphabetischen Substitution basiert.

**Vorteil:** es ist **einfach**.

**Nachteil:** es ist **unsicher**, da es sehr schnell geknackt werden kann.

**Schwachstelle:** Die in der natürlichen Sprache ungleiche Verteilung der Buchstaben wird durch diese Art der Verschlüsselung nicht verborgen, so dass eine Häufigkeitsanalyse (Frequenzanalyse) das Wirken einer einfachen monoalphabetischen Substitution enthüllt.

### Caesar cipher: Vorgang

- Verschiebt jeden Buchstaben des Alphabets um eine bestimmte Anzahl Stellen
- Soll bereits von Julius Caesar verwendet worden sein, daher der Name
- Der Schlüssel wird entweder als Anzahl Stellen, um die verschoben wird, oder als Buchstaben, auf den 'A' verschoben wird angegeben
- Variante: ROT13 (Verschlüsselung = Entschlüsselung)
- Problem 1: Schlüssellänge (nur 26 verschiedene Schlüssel)
- Problem 2: Frequenzanalyse

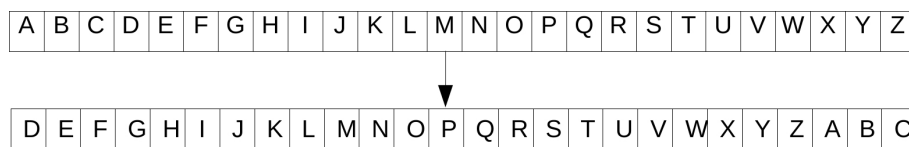


Abbildung 7: Caesar cipher mit Verschiebung um 3 Stellen

Das folgende Diagramm zeigt die Häufigkeitsverteilung der Buchstaben in einem längeren Text in deutscher Sprache:

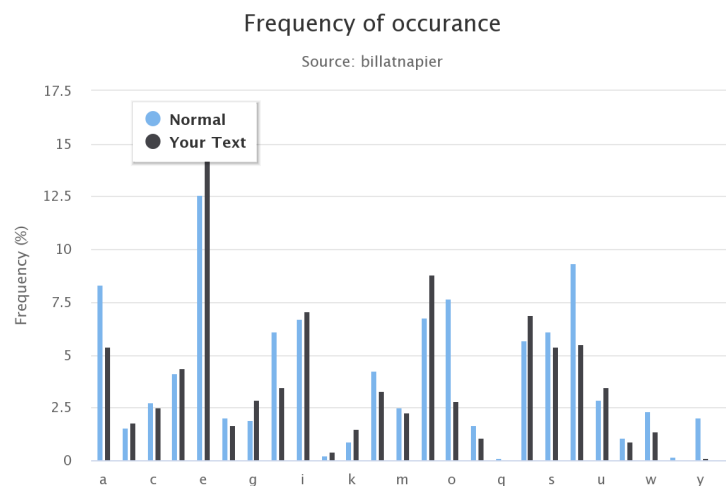


Abbildung 8: Frequenzanalyse unchiffriert

Wie zu erwarten, ist der häufigste Buchstabe E, gefolgt von N und I, wie es im Deutschen üblicherweise der Fall ist. Wird der Text mit dem Schlüssel 10 (oder anders gesagt, mit dem Schlüsselbuchstaben J) chiffriert, erhält man einen Geheimtext, der folgende Häufigkeitsverteilung besitzt:

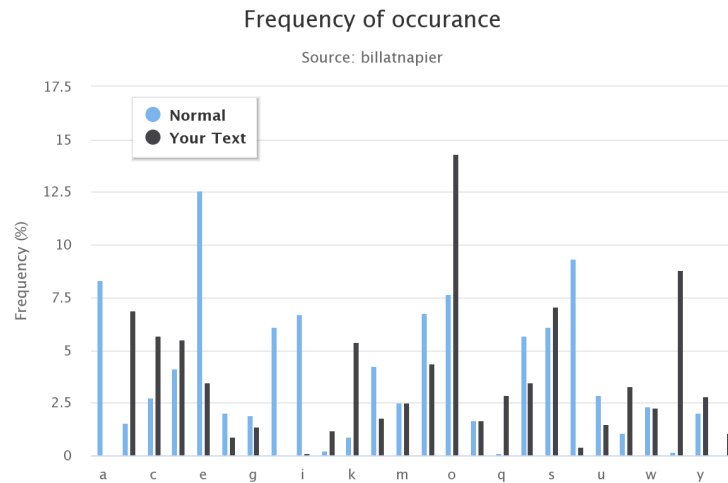


Abbildung 9: Frequenz um 10 Stellen verschoben

Der häufigste Buchstabe ist hier O, gefolgt von X und S. Man erkennt auf den ersten Blick die Verschiebung des deutschen „Häufigkeitsgebirges“ um zehn Stellen nach hinten und besitzt damit den Schlüssel. Voraussetzung ist lediglich, dass man die Verteilung der Zeichen des Urtextes vorhersagen kann.

Besitzt man diese Information nicht oder möchte man auf die Häufigkeitsanalyse verzichten, kann man auch die Tatsache ausnutzen, dass bei der Cäsar-Chiffre nur eine sehr kleine Anzahl möglicher Schlüssel in Frage kommt. Da die Größe des Schlüsselraums nur 25 beträgt, was einer „Schlüssellänge“ von nicht einmal 5 bit entspricht, liegt nach Ausprobieren spätestens nach dem 25. Versuch der Klartext vor.

### Vigenère cipher

- Schlüssel: Wort der Länge  $L$
- Jeder Buchstabe im Text wird mit der Caesar cipher des entsprechenden Schlüsselwortes verschlüsselt
- Anzahl möglicher Schlüssel:  $26^L$
- Problem: Frequenzanalyse jeder  $L$ 'ten Stelle

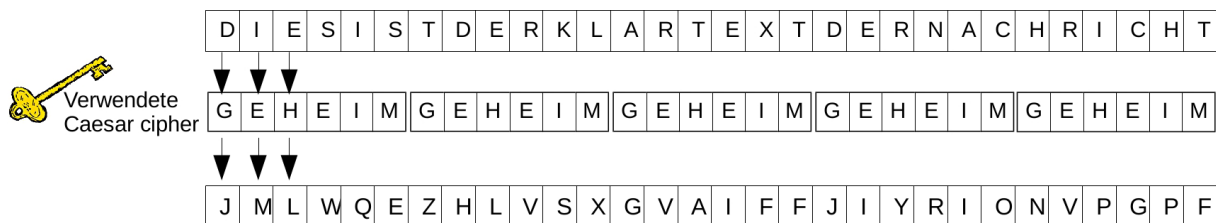


Abbildung 10: Vigenère cipher

### One-time pad

- Jede Stelle wird mit einem anderen Schlüssel verschlüsselt
- Darf nur 1 Mal verwendet werden!
- Anzahl möglicher Schlüssel = Anzahl möglicher Nachrichten
- Ist sicher, d.h. Geheimtext verrät keinerlei (zusätzliche) Information über den Klartext
- Intuitiv: Für einen bestimmten Geheimtext sind **alle** Klartexte (dieser Länge) möglich



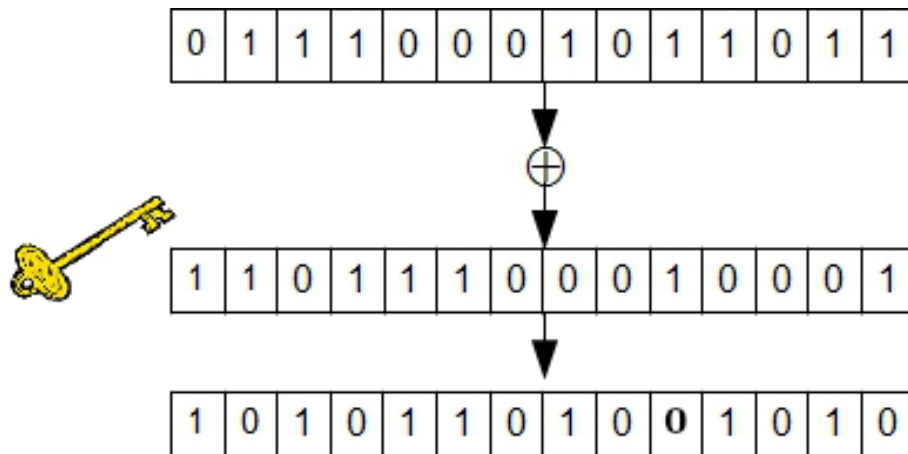


Abbildung 11: Funktionsweise des OTP

Sie wissen welche modernen Verschlüsselungsalgorithmen in der Praxis verwendet werden und was deren Eigenschaften sind

TODO

Sie verstehen was eine Hashfunktion ist und welche Eigenschaften eine kryptographische Hashfunktion ausmachen, bzw. was es heisst, wenn eine Hashfunktion gebrochen ist

**Hashfunktion** Eine Hashfunktion ist eine Abbildung, die eine grosse Eingabemenge (die Schlüssel) auf eine kleinere Zielmenge (die Hashwerte) abbildet. Die Eingabemenge kann Elemente unterschiedlicher Längen enthalten, die Elemente der Zielmenge haben dagegen meist eine feste Länge.

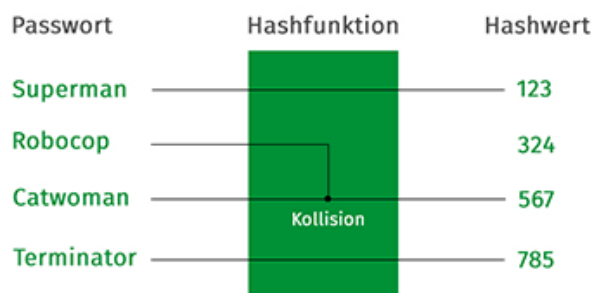


Abbildung 12: einfaches Beispiel einer Hashfunktion

Auf der linken Seite sehen wir 4 Passwörter von beispielsweise 4 Mitarbeitern eines Unternehmens. Die Hashfunktion wandelt nun diese Passwörter in eine Zeichenfolge (dem Hashwert) mit einer festen Länge (hier 3 Zeichen) um. Für das Passwort „Superman“ bekommt man den Hashwert 123, dem Passwort „Robocop“ wird der Hashwert 567 zugeordnet, genauso wie dem Passwort „Catwoman“ und „Terminator“ bekommt 785. Hashfunktionen reduzieren zunächst nur Zeichen beliebiger Länge (unterschiedliche Passwörter) auf Zeichen fester Länge (im Beispiel 3 Zeichen). Sie werden also in eine kleine, kompakte Form gebracht.

**Zusatzinfo zum Hashwert** Der Hashwert ist das Ergebnis, das mittels einer Hashfunktion berechnet wurde. Man definiert eine feste Länge, wie lang ein Hashwert immer sein darf. Oft wird der Hashwert als eine hexadezimale Zeichenkette codiert, d.h. der Hashwert besteht aus einer Kombination von Zahlen und Buchstaben zwischen 0 und 9 sowie A bis F (als Ersatz für die Zahlen 10 bis 15). Ein Hashwert aus 10 hexadezimalen Zeichen könnte so aussehen: „3d180ab86e“.

**Eigenschaft einer Hashfunktion**

- Einwegfunktion: Aus dem Hashwert darf nicht der originale Inhalt erzeugt werden können. In unserem Beispiel darf es nicht möglich sein, aus dem Hashwert „123“ den Ursprungstext „Superman“ zu erzeugen.
- Kollisionssicherheit: Den unterschiedlichen Texten darf nicht derselbe Hashwert zugeordnet sein. Ist diese Voraussetzung erfüllt, so spricht man auch von **kryptographischen Hashfunktionen**. In unserem Beispiel liegt eine Kollision vor, da die Passwörter „Robocop“ und „Catwoman“ denselben Hashwert haben. Damit ist die Hashfunktion im Bild nicht kollisionssicher und es handelt sich nicht um eine kryptografische Hashfunktion.
- Schnelligkeit: Das Verfahren zur Berechnung des Hashwertes muss schnell sein.

**Algorithmen für Passwortspeicherung** Um Passwörter zu speichern werden sog. **Password Based Key Derivation Functions (PBKDF)** verwendet, d.h. **kryptographische Hashfunktionen** welche zusätzlich ressourcen-intensiv (langsam) zu berechnen sind.

- basieren auf einer herkömmlichen Hashfunktion, welche mehrmals verknüpft ausgeführt wird
- die Geschwindigkeit wird durch einen Parameter bestimmt, welcher die Anzahl Runden angibt
- damit werden Angriffe mittels speziell für die Berechnung von Hashfunktionen optimierte Hard- und Software erschwert

Beispiele sind PBKDF2 und bcrypt (Blowfish-Algorithmus), welche zusätzlich viel Memory benötigen, oder scrypt (Entwicklung motiviert durch Verwundbarkeit von PBKDF2 und bcrypt durch Brute-Force-Attacks) und Argon2.

**Gebrochene Hashfunktionen** „Gebrochen“ = „geknackt“. Dies war z.B. bei LinkedIn und Dropbox der Fall. Wie können aber Passwörter geknackt werden, wenn man wegen der Einweg-Eigenschaften der Hashfunktionen nicht auf den ursprünglichen Text zurückschliessen kann? Zunächst muss man wissen, dass fast alle Algorithmen „offen“ liegen, diese also auch von Angreifern genutzt werden können. Das hat zur Folge, dass der Hashwert von einem Passwort immer gleich ist, egal ob es die Plattform oder der Angreifer berechnet. Passwort „Superman“ = MD5-Hash: „527d60cd4715db174ad56cda34ab2dce“. Ein Angreifer kann sich also eine Liste mit typischen unsicheren Passwörtern erstellen und es durch den Hashgenerator jagen. Wenn er nun die Datenbank mit den Hashwerten der Plattform stiehlt, kann er die Hashwerte mit seiner Liste vergleichen. Findet er in der geklauten Liste den Hashwert „527d60cd4715db174ad56cda34ab2dce“, so weiss er, dass dieser Hashwert dem Passwort „Superman“ zugeordnet ist. Solche Listen nennt man **rainbow tables**.

#### Hashfunktionen Algorithmen

Name	Block Länge	Output Länge	Bemerkung
MD5	512	128	gebrochen
SHA-1	512	160	gebrochen
SHA-256	512	256	
SHA-384	1024	384	
SHA-512	1024	512	
SHA3-256	1088	256	
SHA3-384	832	384	
SHA3-512	576	512	

**Sie kennen moderne Hashfunktionen und wissen welche Eigenschaften diese haben**

TODO

#### Sie kennen Anwendungen von Hashfunktionen

Verwendung von Hashfunktionen

- Identifikation einer Datei in peer-to-peer Netzwerken
- Fehlererkennung
- Integritätsprüfung
  - Symmetric Key Solution: Message Authentication Code (MAC) durch einen 'keyed hash'
  - Asymmetric Key Solution: Digital Signature durch Signatur des Hashwertes
- „Proof of work“ in Blockchain

## Sie wissen was ein keyed Hash (HMAC) ist und wofür dieser verwendet werden kann

**HMAC** Ein Keyed-Hash Message Authentication Code (HMAC) ist ein Message Authentication Code (MAC), dessen Konstruktion auf einer kryptografischen Hash-Funktion, wie z.B. MD5 und einem geheimen Schlüssel basiert.

## Sie kennen die „Best-practices“ zu Passwortsicherheit und wissen, gegen welche Angriffe diese schützen

**Passwortsicherheit** Best practices

- Gespeichert wird nur der **Hashwert** des Passwortes
- Ziel: Admin oder Angreifer mit Zugang zur DB erhalten das Passwort nicht

Oder noch besser:

- Das Passwort wird gemeinsam mit einem **Salt** gehasht. Dieser neue Hash wird in der DB abgelegt. Der Salt muss nicht geheim, aber einzigartig (*unique*) sein.
- Ziel: Aufgrund der einzigartigen DB-Einträge ist nicht erkennbar, ob zwei Benutzer dasselbe Passwort haben. Zusätzlich kann ein Angreifer nicht die häufigsten Passwörter hashen und danach vergleichen, welcher Benutzer in der DB dieses Passwort verwendet hat. Er muss jeden Eintrag einzeln angreifen.
- Als Hashfunktion wird eine langsame und ressourcen-intensive Hashfunktion verwendet, z.B. scrypt.
- Ziel: Verlangsamen einer Offline-Attacke auf die Passwort-Hashes.

## 3 Asymmetrische Kryptographie

**Asymmetrische Verschlüsselung** In der asymmetrischen Kryptographie (Verschlüsselung) arbeitet man nicht mit einem einzigen Schlüssel, sondern mit einem **Schlüsselpaar**. Bestehend aus einem **öffentlichen** und einem **privaten Schlüssel**. Man bezeichnet diese Verfahren als asymmetrische Verfahren oder Public-Key-Verfahren.

## Sie verstehen was Public-Key-Kryptographie ist, worauf deren Sicherheit basiert und wie sie zur Verschlüsselung, für Signaturen und zur Authentisierung verwendet werden kann

**Public Key Verschlüsselung** Basiert auf Funktionen, welche einfach zu berechnen sind, deren Umkehrfunktion aber (vermutlich) schwierig zu berechnen ist. Beispiel:

Multiplikation (einfach):	$97 \times 84 = 8051$
Faktorisieren (schwierig):	$8051 = ?$

**TODO** BILDER aus „The Science of Secrecy“

## Sie kennen die gängigen asymmetrischen Verschlüsselungs- und Signaturalgorithmen und wissen, worauf deren Sicherheit basiert

**TODO**

## Sie wissen wie Diffie-Hellmann-Schlüsselaustausch bzw. ElGamal-Verschlüsselung funktioniert

**Diffie-Hellman (DH)** Diffie-Hellman ist ein Schlüsselvereinbarungsprotokoll. Der vereinbarte gemeinsame geheime Schlüssel kann danach zur Verschlüsselung der Nachricht verwendet werden.

**TODO** BILD & ev. Beispiel Wiki

**ElGamal-Verschlüsselung** ElGamal verwendet DH um einen asymmetrischen Verschlüsselungsalgorithmus zu erstellen.

**TODO** BILD ElGamal

**TODO** ev. Beispielrechnung machen

## Sie wissen was kryptographisch sichere Zufallszahlen sind und wo diese verwendet werden

TODO

## Sie wissen was eine elektronische Signatur ausmacht

**Signatur** Die elektronische Signatur ist ein technisches Verfahren zur Überprüfung der Echtheit eines Dokuments, einer elektronischen Nachricht oder anderer elektronischer Daten sowie der Identität des Unterzeichnenden. Sie basiert auf einer Zertifizierungsinfrastruktur, die von vertrauenswürdigen Dritten verwaltet wird: den Anbieterinnen von Zertifizierungsdiensten. Die elektronische Signatur und die handschriftliche Unterschrift werden zudem mit dem neuen Gesetz unter bestimmten Bedingungen als gleichwertig betrachtet.

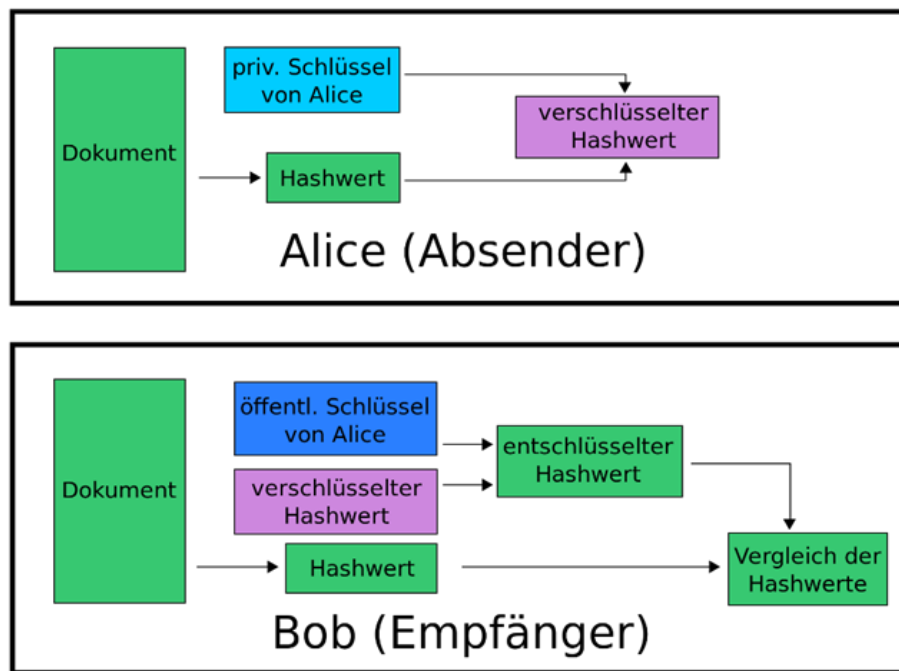


Abbildung 13: Signatur im Detail

**Beispiel** Max Meier erhält von seinem Kunden den Geschäftsvertrag – digital als PDF, wie es heutzutage üblich ist. Vergleichsweise altmodisch geht es aber nachher weiter: Meier druckt das Dokument aus und unterzeichnet es handschriftlich. Den unterschriebenen Vertrag steckt er schliesslich in einen Umschlag und wirft diesen in den nächstgelegenen Briefkasten. Viele Schritte für eine Unterschrift.

Was Max Meier nicht weiss. Dokumente lassen sich auch digital unterzeichnen. Jede digitale Signatur basiert auf der sogenannten asymmetrischen Verschlüsselung. Sie wird auch als Public-Key-Verfahren bezeichnet und nutzt einen öffentlichen und einen privaten (geheimen) Schlüssel. Mit dem privaten Schlüssel wird die digitale Signatur erzeugt, während mit dem öffentlichen Schlüssel die Authentizität der Unterschrift überprüft wird. Eigenschaft einer Signatur:

1. **Fälschungssicherheit:** Nach dem Unterschreiben kann das Dokument nicht mehr (unerkannt) verändert werden.
2. **Authentizität:** Die Unterschrift kann zweifelsfrei (überprüfbar) einer bestimmten Person zugeordnet werden.
3. **Unleugbarkeit:** Der Unterzeichner kann später nicht abstreiten, das Dokument unterschrieben zu haben.
4. **Willenserklärend:** Die Unterschrift kann nur willentlich (bewusst) unter das Dokument gesetzt worden sein.

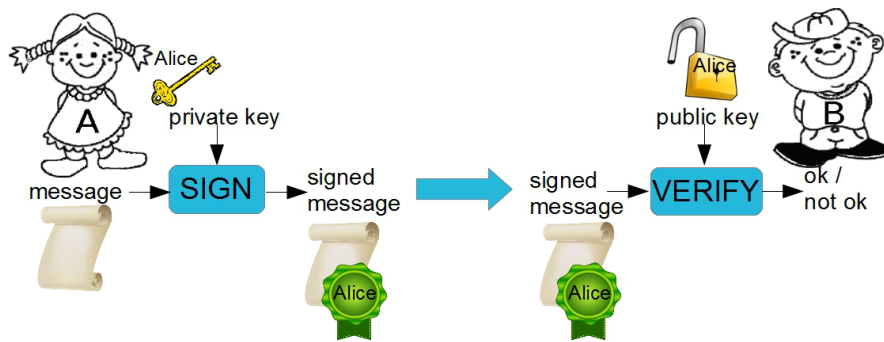


Abbildung 14: Alice verschickt eine signierte Nachricht an Bob

Sie wissen wie hybride Verschlüsselung bzw hybride Signaturen funktionieren  
 TODO

## 4 Zertifikate und SSL-TLS

Sie kennen die verschiedenen Arten von „Trust“

Problematik: Wie ordnet man ein Public Key einer bestimmten Person / Entität zu?

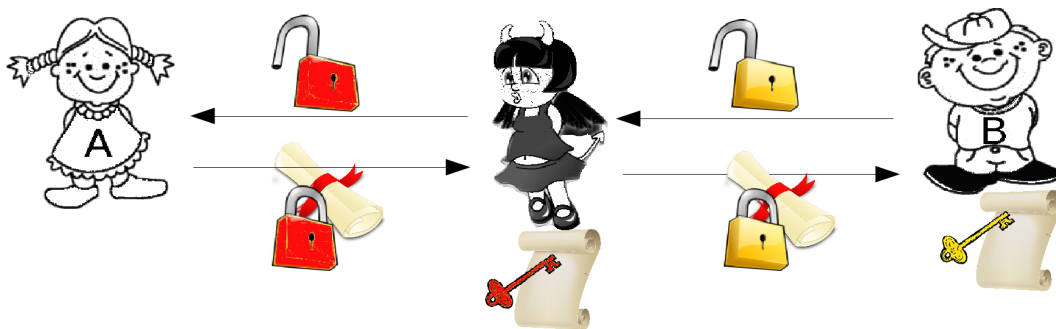


Abbildung 15: Eve als man in the middle

**Direct Trust** Alice vertraut der Authentizität von Bob's Public Key, durch direktes Überprüfen, normalerweise über den Fingerprint des Key's.

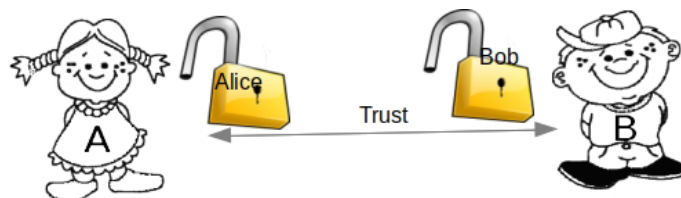


Abbildung 16: Alice vertraut Bob durch direkte Überprüfung

- Persönliche Überprüfung
- Vorinstalliert in System oder Software (z.B. Public Key von Google-Server in Chrome, Apps, VPN-Clients)
- Publiziert auf Webseite oder in Zeitung

Benötigt jedoch einen authentischen Kanal zum Etablieren des Trust.

**Web of Trust (WOT)** Alice vertraut der Authentizität von Daves Public Key, weil dieser von Charlie signiert wurde, dessen Public Key wiederum von Bob signiert wurde, dem sie vertraut.

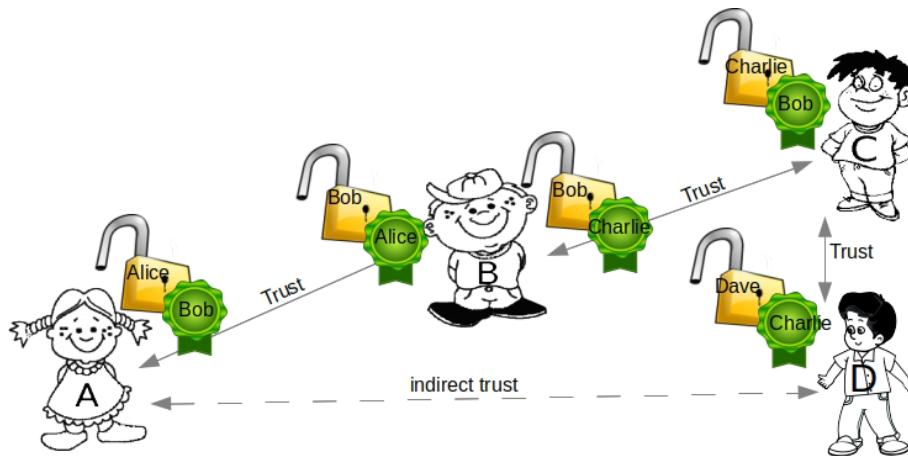


Abbildung 17: Alice vertraut Dave indirekt durch Vertrauensnetz

**Hierarchical Trust (PKI)** Eine *Public Key Infrastructure (PKI)* ist ein System, das digitale Zertifikate ausstellen, verteilen und Prüfen kann. Im Gegensatz zum WOT ist eine PKI hierarchisch aufgebaut und bedingt deshalb Root Certification Authorities, welche über alle anderen Zertifizierungstellen steht.

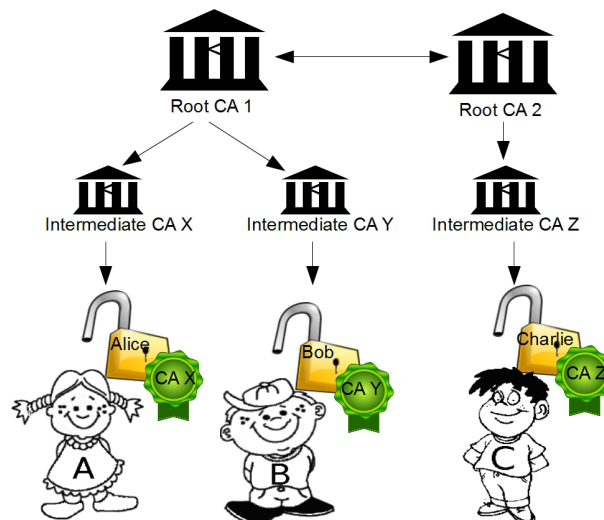


Abbildung 18: Hierarchical Trust durch Certificate Authorities

**Sie wissen was eine Public-Key-Infrastruktur, eine Certificate Authority und ein Zertifikat ist, wofür und wie diese verwendet werden und wie Zertifikate ausgestellt und revoziert werden**

**Grundbegriffe PKI** Die Zertifizierungstelle *Certificate Authority (CA)*

Eine CA ist eine Organisation, welche digitale eZertifikate ausstellt. Ein digitales Zertifikat ordnet einen bestimmten öffentlichen Schlüssel einer Person oder Organisation zu. Diese Zuordnung wird von der Zertifizierungstelle beglaubigt, indem sie sie mit ihrer eigenen digitalen Unterschrift versieht.

Ein Zertifikat wird durch eine sog. *Chain of Trust* beglaubigt. Eine *intermediate CA* signiert das Zertifikat (Public Key) des Endbenutzers. Das Zertifikat der intermediate CA wird wiederum von einer anderen CA unterschrieben. Das letzte Zertifikat in dieser Kette heisst *Root Certificate* und enthält den Public Key der *root CA*. Dieses Zertifikat ist normalerweise *self signed*, also von der root CA selbst unterschrieben.

**Sie wissen was SSL/TLS ist, welche Funktionalität es erreicht und wie das Protokoll konzeptionelle abläuft**

SSL-TLS erreicht

- Authentisierung des Servers gegenüber dem Client

- *Optional*: Authentisierung des Clients gegenüber dem Server ('mutual SSL')
- Verschlüsselung und Authentisierung der Daten

Das SSL/TLS-Protokoll läuft in zwei Phasen ab:

- **Handshake**: vereinbart mittels Public-Key-Kryptographie einen Schlüssel
- **Datenaustausch**: verwendet Secret-Key-Kryptographie zum Verschlüsseln und Authentisieren

Beispiele für SSL/TLS-ciphers:

- TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384

## Teil III

# Angriffe (SW 05-06)

## 5 Angriffe auf Webanwendungen

**Bedrohungen auf Anwendungsebene** Webanwendung, Session, Headers, CSRF

### Sie wissen was eine Webanwendung ausmacht, wie HTTP funktioniert

Was unterscheidet eine Webanwendung aus Sicherheitssicht zu anderen Anwendungen?

- Kommuniziert über HTTP mit einem Server
  - zustandsloses Protokoll
- Läuft in einem Browser
  - Mehrere Webanwendungen können parallel im gleichen Browser laufen
  - Die Webanwendung *erbt* vom Browser implementierte Features bzw. muss diese richtig ansprechen

**HTTP** Der Browser kommuniziert mit dem Webserver über das **Hypertext Transfer Protokoll (HTTP)**. HTTP besteht aus *Requests* und *Reponses*.

**HTTP-Request-Methoden** Die häufigsten HTTP-Request-Methoden sind **GET** und **POST**. Es existieren aber auch **PUT, HEAD, DELETE, PATCH, OPTIONS**.

**GET** `https://www.hslu.ch/?p=5 HTTP/1.1`

User-Agent: Mozilla/5.0

- Message Body: kein
- Ruft Daten vom Server ab
- Sollte Serverzustand nicht verändern

**POST** `https://www.hslu.ch/ HTTP/1.1`

User-Agent: Mozilla/5.0

- Message Body: `id=123&pwd=password`
- Darf Serverzustand verändern
- Wird nicht gecachet

### Häufigste Reponse-Codes

- 200 OK
- 204 No Content
- 301 Moved Permanently
- 302 Found (Vorher: „Moved temporarily“)
- 304 Not Modified
- 400 Bad Request
- 403 Forbidden
- 404 Not Found
- 500 Internal Server Error



**HTTP Zustand** HTTP ist ein zustandsloses Protokoll, d.h. es hat kein ‘Gedächtnis’, bzw. Erinnerung.

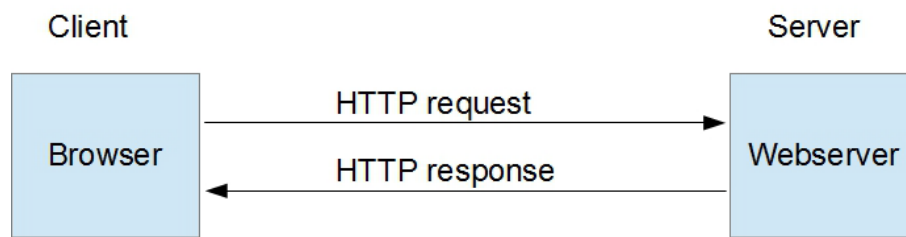


Abbildung 19: HTTP zustandslos

Die einzige Möglichkeit einen Zustand an den Client zu übergeben ist, diesen per weiteren Requests mitzuschicken. Die Zustände werden mit einem Cookie oder einem „Hidden field“ erfasst.

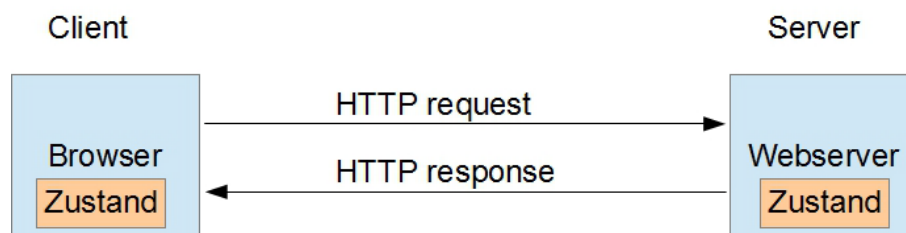


Abbildung 20: HTTP Zustand per Request (hidden field)

**Cookies** Cookies sind kurze Textdaten, welche vom Server als Header an den Browser übermittelt werden und von diesem ebenso als Header bei requests wieder mitgesendet werden. Cookies werden vom Browser verwaltet. Die meistgenutzte Möglichkeit ist es, ein Cookie zu setzen. Jedoch dürfen auch Cookies nicht client-seitig angepasst werden können!

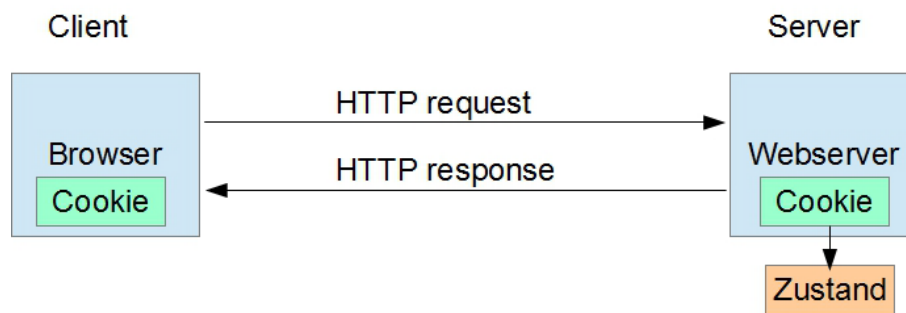


Abbildung 21: Einsatz eines Cookies

**Cookie Eigenschaften** Die Eigenschaften von Cookies sind:

- **Persistent** (mit Ablaufdatum) oder **Session-Cookie** (ohne Ablaufdatum)
- **Secure** (wird nur über HTTPS übertragen)
- **HTTP Only** (darf nur von HTTP gelesen werden)
- **Same Site** (wird nicht bei Cross-Domain-Aufrufen mitgesendet, z.B. ‘embedded’ Link, Image)

**Sie wissen was eine Session ist und welche Eigenschaften einer Session bei welchen Angriffen wichtig sind bzw wie sie gegen gewisse Angriffe Schutz bieten**

**Session** Eine Session ist der Zeitraum, in dem ein Client eine stehende Verbindung mit einem Server hat; vom Login bis zum Logout. Der Server vergibt dem Client eine eindeutige Session-ID. Die Sitzungsdaten (z.B. Warenkorb) werden im Server gespeichert. Bei jedem Request gibt der Client seine Session-ID mit, damit der Server beim Response die zugehörigen Daten dieser ID übermitteln kann. Es gibt auch Sessions



ohne stehende Verbindung (ohne Login). Dies wird zu Statistikzwecken verwendet, z.B. um die Bewegung des Besuchers auf der Website zu verfolgen. Oder aber auch um einen Warenkorb ohne Login verwenden zu können.

#### **Schwaches Session-Management** Was ist das?

- der Sessionwert ist vorhersagbar
- der Sessionwert kann vom Client gesetzt werden
- die Cookie-Attribute 'Secure', 'HttpOnly' oder 'Same Site' sind nicht gesetzt
- Cookie-Domain oder -Pfad sind nicht so eingeschränkt wie möglich
- die Session wird bei einem Logout nicht invalidiert
- die Session hat kein server-seitiges Timeout (Inaktivitäts- und absolutes Timeout)

#### **Schwaches Session-Management** Was kann man dagegen tun?

- lange und kryptographisch zufällige Sessionwerte wählen
- nur vom Server gewählte Sessionwerte akzeptieren
- Cookies als 'Secure', 'HttpOnly' oder 'Same Site' mit so eingeschränkter Domain und Pfad wie möglich setzen
- Session **server-seitig** bei einem Logout oder Timeout invalidieren

**Same Origin Policy** Mehrere Webanwendungen können im gleichen Browser parallel laufen. Die Same-Origin-Policy verhindert, dass eine parallel laufende Webanwendungen uneingeschränkt

- auf die Daten einer anderen Anwendung zugreifen
  - die Cookies einer anderen Anwendung lesen oder mitschicken
  - Requests auf die andere Anwendung absetzen
- kann.

Same Origin Policies im Browser gibt es z.B. für Cookies, DOM access (Zugang zu document.cookie), HTML5Storage, XMLHttpRequests.

**Same Origin Policy: Cookies** Cookies haben eine **domain** und **path**.

- **Setzen des Cookies:** Nur Domain-Suffix des URL-Hostname dürfen gesetzt werden. (Aber keine Top-Level Domains!)  
Path kann beliebig gesetzt werden.
- **Senden des Cookies:** Cookies werden nur dann mitgeschickt, wenn die Cookie-Domain ein Domain-Suffix der URL-Domain und der Cookie-Path ein Prefix des URL-Path ist.

#### **Session Fixation** Was ist das?

Der Sessionwert wird nach einem Login oder Loginschritt nicht geändert. Ein angreifer mit Zugang zu einer unauthentisierten Session kann warten bis ein Benutzer sich einloggt und ist damit selbst eingeloggt.

#### **Session Fixation** Was kann man dagegen tun?

Sessionwert nach jedem Authentisierungsschritt ändern.

## Sie kennen sicherheitsrelevante Header

### **Sicherheitsrelevante Response-Header**

1. **HSTS: Strict-Transport-Security: max-age=31536000; includeSubDomains**  
Seite wird nur via HTTPS aufgerufen. max-age muss hoch gesetzt werden!
2. **Frame-Options: X-Frame-Options: deny**  
Verbietet das Einbinden der Seite in einem Frame oder erlaubt es nur für bestimmte Domains
3. **XSS-Protection: X-XSS-Protection: 1; mode=block**  
Filtert und säubert oder blockiert die Anzeige der Seite, wenn ein XSS-Angriff entdeckt wird
4. **Content-Type-Options: X-Content-Type-Options: nosniff**  
Verhindert, dass der Content als einen anderen MIME-Type interpretiert wird als angegeben
5. **CSP: Content-Security-Policy: script-src 'self'**  
Definiert, welche Ressourcen (z.B. Bilder, Scripts, Fonts, etc.) von wo eingebunden werden können
6. **CORS Access-Control-Allow-Origin: http://foo.example**  
Cross-Origin Resource Sharing (CORS) ist ein Mechanismus, der Webbrowsern oder auch anderen Webclients Cross-Origin-Requests ermöglicht. Zugriffe dieser Art sind normalerweise durch die Same-Origin-Policy (SOP) untersagt. CORS ist ein Kompromiss zugunsten grösserer Flexibilität im Internet unter Berücksichtigung möglichst hoher Sicherheitsmassnahmen.

## 7. Caching-Options **TODO: hat jemand Infos?**

### 8. HPKP (**deprecated!**): Public-Key-Pins:

```
pin-sha256=d6qzRu9z0ECb90Uez27xWltNsjoelMd7GkYYkVoZWmM=;
pin-sha256=Ë9CZ9INDbd+2eRQozYqqbQ2yXLVKB9+xcprMF+44U1g=;
report-uri=http://example.com/pkp-report;
max-age=10000; includeSubDomains
```

HTTP Public Key Pinning: Nur das Serverzertifikat mit dem korrekten Fingerprint wird akzeptiert. Wurde wieder abgekündigt und die meisten Browser unterstützen es nicht mehr.

## Sie verstehen wie ein Cross-Site-Request-Forgery-Angriff abläuft und wie man sich dagegen schützen kann

### CSRF - Cross-Site Request Forgery Was ist das?

Der Angreifer bringt einen Benutzer dazu, einen Request aus seinem Browser abzusetzen und dadurch eine Aktion auf dem Server auszulösen. Ist der Benutzer zu dem Zeitpunkt eingeloggt, wird das Cookie automatisch mitgeschickt.

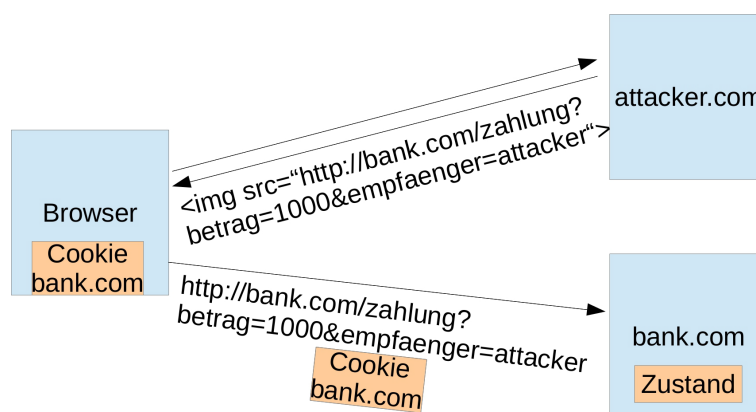


Abbildung 22: Cross-Site Request Forgery

### CSRF - Cross-Site Request Forgery Was kann man dagegen tun?

- **CSRF-Token:** ein Secret als Teil des Form Field oder Header mitgeben (Secret darf nicht vorhersagbar sein)
- **Zusätzlich:** Same-Site-Attribut setzen

## 6 Angriffe auf Protokollebene

### Sie kennen die Grundbegriffe der Anwendungssicherheit

**Bedrohungen auf Protokollebene** Begriffe: Social Engineering, Angriffe auf ARP, TCP/IP, DNS, SSL, HTTP

### Kurzübersicht

- **Bedrohungen auf Link-Layer:** Spoofing
- **Bedrohungen auf Transport-Layer:** Denial of Service (DoS)
- **Bedrohungen auf SSL / TLS:** Preisgabe Sensitiver Daten
- **Bedrohungen auf Anwendungslayer:** Cross Site Scripting (XSS), Code Injection
- **Bedrohungen auf Layer 8 (Mensch):** Social Engineering

**Flaws vs. Bugs** Bei Softwaredefekten wird unterschieden zwischen Flaws und Bugs

- **Flaw:** Ein Flaw ist ein Defekt im Design der Software
- **Bug:** Ein Bug ist ein Defekt in der Implementation

### Grundbegriffe: Bedrohung

- **Threat:** Möglicher Grund für einen ungewollten Vorfall, der das System oder die Organisation schädigen kann.
- **Threat Agent:** Individuum oder Gruppe welche eine Bedrohung darstellt.

### Aktive vs. passive Angriffe

Bei einem **passiven Angriff** hält sich der Angreifer an das Protokoll. Er verändert z.B. die ausgetauschten Nachrichten nicht, hört aber die Kommunikation ab. Bei einem **aktiven Angriff** hält sich der Angreifer nicht an das Protokoll. Er verändert z.B. Nachrichten.

## Sie kennen Beispiele von Angriffen auf verschiedenen Ebenen des Protokollstacks und wissen, was diese bewirken

### OSI-Layers

Die 7 Tierschichten des OSI-Modells, wobei die 8te sich auf den Mensch bezieht.

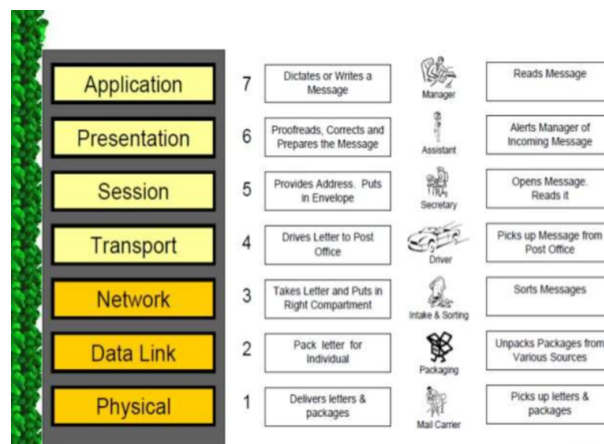


Abbildung 23: OSI-Layers

### OSI vs Internet Reference Model

Die 7 Tierschicht des OSI-Modells, wobei die 8te sich auf den Mensch bezieht.

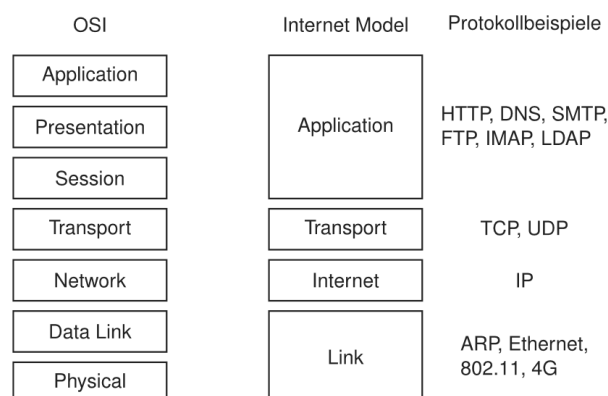


Abbildung 24: OSI vs. Internet Reference Model

## Encapsulation (Datenkapselung)

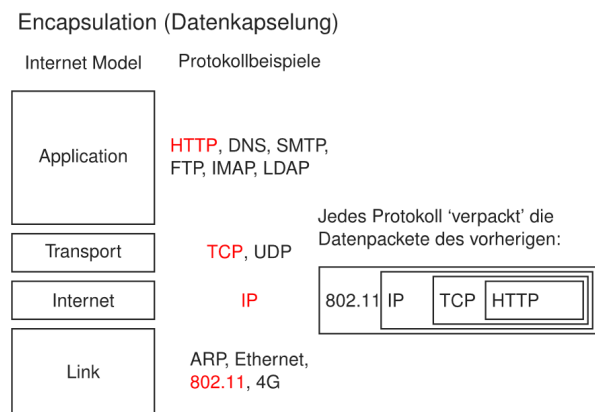


Abbildung 25: TCP-Encapsulation

## Beispiel: ARP-Spoofing auf dem Link-Layer

Beispiel: ARP spoofing

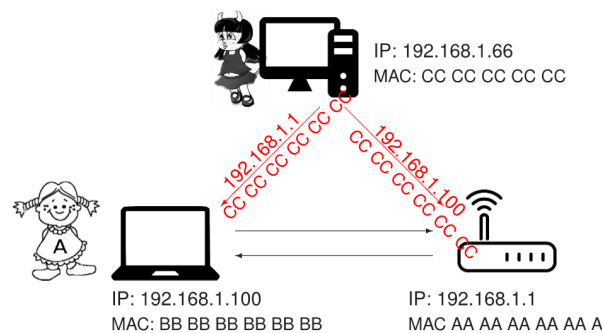


Abbildung 26: ARP Spoofing

**Spoofing: Was ist das?** Eine person oder ein Programm gibt sich als jemand anderen oder etwas anderes aus.

**Beispiele:**

- Telefonnummern-Spoofing (Call Centers etc.)
- Email-Adressen-Spoofing
- IP Spoofing
- DNS Spoofing
- ARP Spoofing
- Content Spoofing

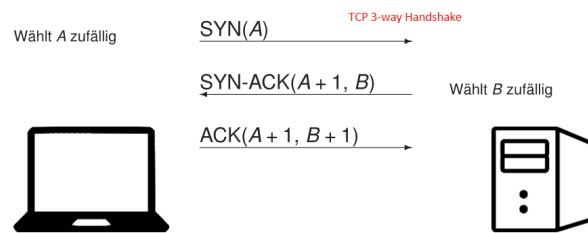
„ARP-Spoofing (vom engl. to spoof – dt. täuschen, reinlegen) oder auch ARP Request Poisoning (zu dt. etwa Anfrageverfälschung) bezeichnet das Senden von gefälschten ARP-Paketen. Es wird benutzt, um die ARP-Tabellen in einem Netzwerk so zu verändern, dass anschließend der Datenverkehr zwischen zwei (oder mehr) Systemen in einem Computernetz abgehört oder manipuliert werden kann. Es ist eine Möglichkeit, einen Man-in-the-Middle-Angriff im lokalen Netz durchzuführen.“ - Wikipedia

**Spoofing: Was kann man dagegen tun?**

**Je nach Situation unterschiedlich, zB.:**

- Authentisieren
- Angaben überprüfen

## Repetition: TCP Verbindungsaufbau (vereinfacht)



Folie 21, Woche vom 21. Oktober 2019

Abbildung 27: 3-Way Handshake

## Bedrohungen auf Transport-Layer: Denial of Service (DoS)

**Denial of Service** Syn-Nachrichten werden mit gespoofter IP gesendet. Syn-Acknowledgements Nachrichten gehen nirgendwo hin. Server wird überflutet mit Abfragen und kann nicht schneller abarbeiten als Sie reinkommen. Als Resultat kann somit von den meisten Benutzern die Seite nicht angezeigt werden.

Beispiel: SYN Flood

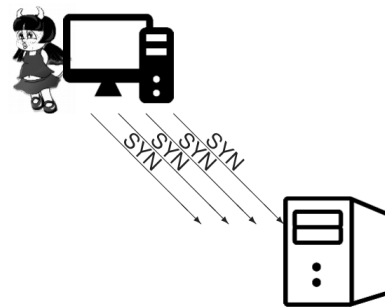


Abbildung 28: SYN Flood

**Beispiel: Distributed Reflection Denial of Service** „Hierbei adressiert der Angreifer seine Datenpakete nicht direkt an das Opfer, sondern an regulär arbeitende Internetdienste, trägt jedoch als Absenderadresse die des Opfers ein (IP-Spoofing). Die Antworten auf diese Anfragen stellen dann für das Opfer den eigentlichen DoS-Angriff dar. Durch diese Vorgehensweise ist der Ursprung des Angriffs für den Angegriffenen nicht mehr direkt ermittelbar.“ - Wikipedia

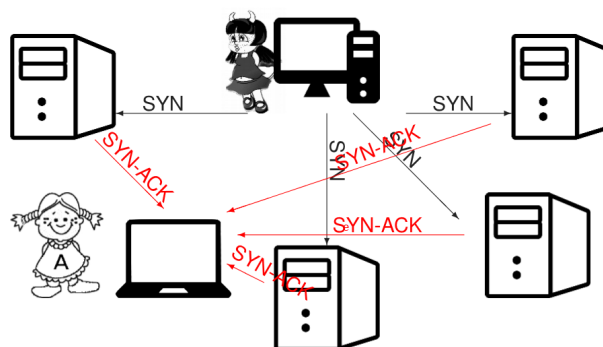


Abbildung 29: DRDoS

**Denial of Service: Was kann man dagegen tun?** Jedes System bricht irgendwann zusammen! Es geht darum sicherzustellen, dass dabei keine bleibenden Schäden am Kernsystem entstehen und das System nach einem Angriff schnell wieder funktionsfähig zu machen.

#### Schutzbeispiele:

- Beschränkung der Anzahl (Web-) Requests pro Zeiteinheit / IP
- Sicherstellen, dass der „Flaschenhals“ weit vorne auftritt (z.B. Firewall) um Kernsysteme zu schützen
- Sicherstellen, dass das System sich nicht selbst überlastet durch freigeben nicht mehr verwendeter Ressourcen, vermeiden von unendlichen Loops etc.
- Disaster Recovery Plan

#### SSL/TLS im Internet Modell

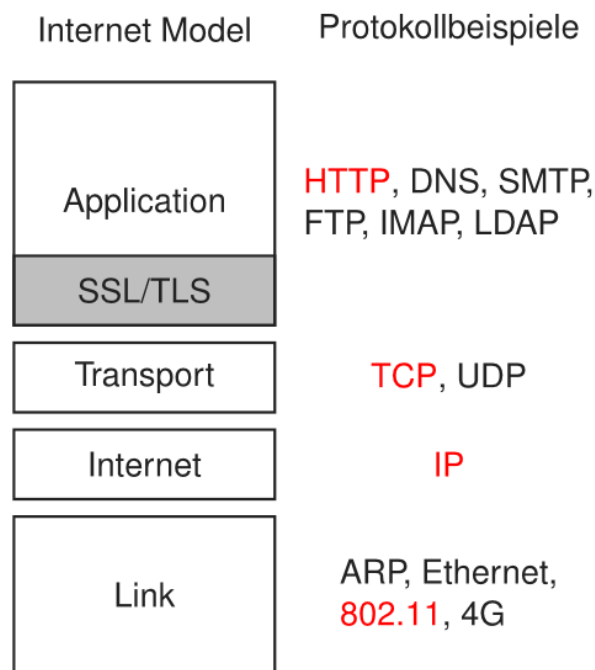


Abbildung 30: SSL/TLS

#### Bedrohung auf SSL / TLS (Preisgabe Sensitiver Daten)

**Preisgabe Sensitiver Daten: Was ist das?** Angreifer stehlen Schlüssel, Passwörter, Geschäftsgeheimnisse, Personendaten oder andere sensitive Daten vom Server, bei der Übertragung oder vom Client

#### Preisgabe sensativer Daten: Was kann man dagegen tun?

- Keine Daten speichern oder übertragen, welche nicht benötigt werden.
- Daten nach ihrer Sensitivität klassifizieren und entsprechend behandeln
- Sensitive Daten nur gespeichert auf dem Server ablegen
- Passwörter mit Salt und Pepper und einer starken Passwort-Hashfunktion gehasht ablegen.
- Daten verschlüsselt übertragen (FTP >SFTP, HTTP >HTTPS, etc). Zertifikat überprüfen!
- Sicherstellen, dass sichere Ciphers verwendet werden. Keine sensiblen Daten auf der Clientseite cachen.

#### Bedrohungen auf Anwendungslayer: Cross Site Scripting (XSS) und Code Injection

**XSS: Was ist das?** Ein Angreifer bringt den legitimen Server dazu ein Script an den Browser zu senden. Dieses wird im Kontext des legitimen Servers ausgeführt. Es wird zwischen **stored** und **reflected** XSS unterschieden

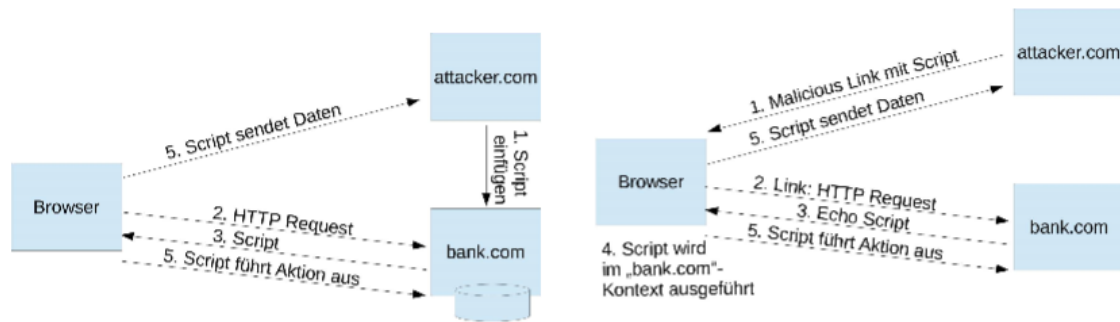


Abbildung: 'Stored' und 'reflected' XSS.

Abbildung 31: XSS

**XSS: was kann man dagegen tun?**

- **Escaping** aller unsicheren Daten (z.B. vom Benutzer eingegebene) bevor sie angezeigt werden.  
Bsp. Ersetzen von `&` `<` `>` `"` durch `&amp;` `&lt;` `&gt;` `&quot;` `&#x27;` `&#x2F;`

Zusätzlich sollen folgende Massnahmen getroffen werden:

- Cookie als HttpOnly-Cookie setzen
- Header-Felder setzen  
Bsp. Content-Security-Policy: default-src: 'self'; script-src: 'self' static.domain.tld
- Bsp. X-XSS-Protection: 1; mode=block

**Code Injection: Was ist das?** Vermischung von 'Code' und 'Daten'

Ausgeführter Code:

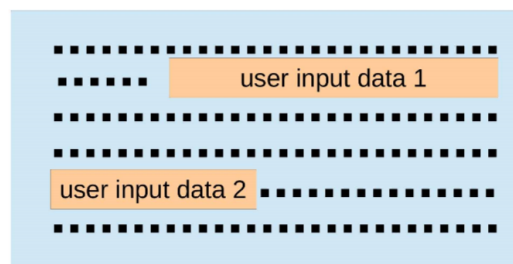
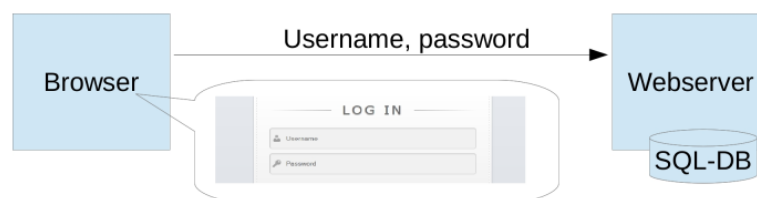


Abbildung 32: Code Injection

**Auf Server-Seite ausgeführter Code:**

```
$result = mysql_query(" select * from Users
where(name='frank' OR 1=1); DROP TABLE Users; --'
and password='whatever')");
```

Abbildung 33: Beispiel Code Injection

**Code Injection: Was kann man dagegen tun?**

- 'Prepared statements' verwenden  
Bsp:  

```
$statement = $db->prepare('select * from Users
where(name=? password=?);');
$stmt->bind_param('ss', $user, $pass);
```
- Whitelisting der Inputs
- Sanitizing der Inputs  
Bsp. Löschen von Zeichen wie ';' oder Ersetzen durch 'sichere' Zeichen wie '\';\'
- Rechte des technischen Benutzers auf der DB einschränken
- Verwenden eines sicheren APIs

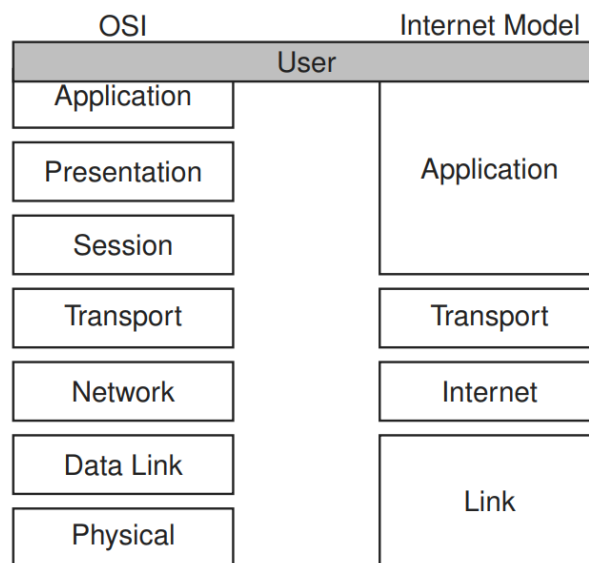
**Layer 8**

Abbildung 34: Layer 8

**Social Engineering: Was ist das?** Zwischenmenschliche Beeinflussungen mit dem Ziel, bei Personen bestimmte Verhaltensweisen hervorzurufen, sie zum Beispiel zur Preisgabe von vertraulichen Informationen, zum Kauf eines Produktes oder zur Freigabe von Finanzmitteln zu bewegen.

**Social Engineering: Was kann man dagegen tun?**

- Benutzer schulen ('awareness')
- Für den Benutzer verständliche Abläufe sicherstellen
- Benutzer nicht zum Umgehen von Sicherheitsmassnahmen verleiten
- Fraud Detection-Massnahmen
- Technische Massnahmen welche den Angriff verhindern, z.B. Vereinzelungsanlage, nicht vorlesbare Codes etc.

TODO

**Teil IV****Management (SW 07-09)****7 Standards & Frameworks, ISMS**



## Sie wissen, was ein ISMS ist und wie man damit umgeht

**ISMS** Ein *Information Security Management System (ISMS)* (auf Deutsch: Managementsystem für die Informationssicherheit) definiert Regeln, Methoden und Abläufe, um die IT-Sicherheit in einem Unternehmen zu gewährleisten, zu steuern, zu kontrollieren und zu optimieren.

### Zweck

- Die (durch die IT verursachte) Risiken sollen identifizierbar und beherrschbar werden.
- Sicherheit erhalten, dass teure Informationen und Daten der Unternehmung angemessen geschützt sind.
- Rechtliche (Datenschutz- oder Berufsgesetz bei Ärzten / Anwälte) und auch Marktanforderung erfüllen (wenn morgen in den Medien publik wird, dass bei der UBS Bank «gehakt» und Millionen gestohlen wurde, dann würden die Kunden nicht länger ihr Vermögen bei der UBS deponieren).

### Vorgehen

- Man sollte einen Prozess unterhalten, mit dem die Risiken der Informationssicherheit identifiziert und bewertet werden können. Dazu sollen Kontrollen bestimmt, eingeführt und stetig verbessert werden können.
- Davor muss zuerst der Schutzbedarf von Vermögenswerten bestimmt und Schutzmassnahmen eingeführt werden.

## Sie kennen die wichtigsten Standards der Informationssicherheit

### Standards

- ISO 27000: ISMS – Overview and vocabulary (Überblick / Index)
- ISO 27001: ISMS – Requirements (Anforderungskatalog)
- ISO 27002: Code of practice for information security controls (Analog: Kochbuch; darin steht drin, welche Massnahmen ich tätigen muss)
- ISO 27003: implementation guidance (wie ich die Anforderung umsetze)
- ISO 27004: Information security management – Measurement (Ziele müssen messbar sein, z.B. Jahresziele beim Mitarbeiter Gespräch; Ende Periode kann überprüft werden, ob die Ziele erreicht wurden)
- ISO 27005: Information security risk management (Risiko Bewältigung)

## Sie finden sich in den Standards ISO 27001 und 27002 zurecht

TODO

## Sie verstehen die Grundzüge der BSI-Standards (BSI=Bundesamt für Sicherheit in der Informationstechnik, Deutschland)

TODO

## Sie kennen die Struktur und Grundziele des NIST CyberSecurityFrameworks

TODO

## 8 Risiko-Management und IT-Grundschutz

### Das Risikoanalyse-Verfahren verstehen

TODO

### Die Unterschiede zum Grundschutzverfahren kennen

TODO

### Eine einfache Risikoanalyse durchführen können

TODO

## Sie verstehen die Idee, die Ziele und die Konzepte des IT-Grundschutz-Vorgehens

TODO

Sie kennen den Aufbau der IT-Grundschutz-Kataloge und deren Anwendungsweise

TODO

Sie können die Teilschritte zum Aufbau eines Sicherheitskonzeptes nach IT-Grundschutz durchführen, kombinierte Risikoanalyse

TODO

## 9 Awareness

Sie verstehen die Wichtigkeit der «Awareness »

TODO

Sie kennen verschiedene Prozesse und Vorgehensweisen für die Initiierung, Durchführung und Erfolgsprüfung einer Awareness-Kampagne und können diese anwenden

TODO

Sie kennen die relevanten Erfolgsfaktoren der Mitarbeiter-Sensibilisierung und -Schulung und können diese in einer Kampagne umsetzen

TODO

## Teil V

# Access Control (SW 10)

## 10 Access Control

Sie kennen verschiedene Arten der Authentisierung, wissen wie diese technisch ablaufen und was deren Vor- und Nachteile sind

TODO

Sie wissen wie verschiedene Authentisierungstoken technisch funktionieren, was deren Vor- und Nachteile sind und wie sie beim Login oder bei der Transaktionsbestätigung im e-Banking eingesetzt werden

TODO

Sie wissen was Authentisierung, Autorisierung ist, warum diese wichtig sind und wie Angriffe darauf ablaufen

TODO

## Teil VI

# Multi-Party-Computation (SW 11)

## 11 Cryptographic Protocols

Sie kennen einfache Beispiele von verteilten sicheren Berechnungen und verstehen wie die entsprechenden Protokolle ablaufen

TODO

## 12 Secret Sharing

Sie kennen Arten von Sicherheit von verteilten sicheren Berechnungen und wie diese angegriffen werden können

TODO

Sie wissen welche Eigenschaften elektronisches Geld ausmachen und kennen die technischen Grundlagen von Bitcoin

TODO

## 13 Zero Knowledge Proof

Sie wissen was Zero-Knowledge-Proofs sind und wie diese ablaufen

TODO

## Teil VII

# Quantum (SW 12)

## 14 Quantum Computing and Quantum Cryptography

Sie wissen was ein Quantencomputer ist und was ihn von einem „klassischen“ Computer unterscheidet

TODO

Sie verstehen welchen Einfluss die Existenz eines Quantencomputers auf die Kryptographie hat

TODO

Sie verstehen wie Quantenschlüsselaustausch funktioniert

TODO

## Teil VIII

# WAF, Federations (SW 13)

## 15 Firewalls

Sie wissen was die Aufgaben einer Firewall sind

**Aufgaben einer Firewall** Filtern der ein- und ausgehenden Kommunikation nach

- **Service control:** Z.B. Protokoll, Portnummer, IP-Adresse
- **Direction control:** Wer hat die Verbindung aufgebaut bzw. den Service initiiert?
- **User control:** Welcher Benutzer versucht einen bestimmten Service auszuführen?
- **Behaviour control:** Wie wird ein Service verwendet? Z.B. Spam-Filter

Sie verstehen die Funktionsweise einer WAF und wie sie eine Webanwendung vor Angriffen schützen kann

**WAF** Funktionalitäten einer Web Application Firewall

- Terminierung der SSL-Verbindung
- Protokoll-Einschränkungen (Port, HTTP/HTTPS)

- Load Balancing
- DoS-Verhinderung
- Session-Management (Cookie-Store, Timeouts)
- Filter gegen SQL-, HTML-, Code-Injection, XSS
- URL-Verschlüsselung
- Fehlerseiten umschreiben
- Request- und Response-Header setzen, entfernen, blockieren
- CSRF-Token einfügen
- 'Dynamic Value Endorsement'
- Logging und Monitoring

## 16 Federations

Sie verstehen wie Authentisierung mit Identity Federation abläuft, was die Voraussetzungen dafür sind und was die Vor- und Nachteile von Federations sind

TODO

## Teil IX

## Talks (SW 14)

## 17 Malware

Sie verstehen, welche Arten von Malware es gibt, welche Massnahmen gegen Malware sinnvoll sind und wie diese wirken

TODO

## 18 WAF

Sie verstehen wo Machine-Learning in einer WAF eingesetzt werden kann und was eine Machine-Learning-Ansatz vom „herkömmlichen“ Einsatz einer WAF unterscheidet

TODO

Sie kennen Beispiele von Angriffen, welche mittels Machine-Learning auf einer WAF erkannt werden konnten

TODO

# Index

3-Way Handshake, 20

Aktive vs. Passive Angriffe, 19

ARP-Spoofing, 20

Bedrohungen auf OSI-Layern, 18

Chosen plaintext attack, 6

Ciphertext only attack, 6

Code Injection, 23

Cross Site Scripting (XSS), 22

DoS, 21

DRDoS, 21

Eintretenshäufigkeit, 4

Fehler vs. Bugs, 18

Informationstheoretische Sicherheit, 6

Integrität, 3

Kerckhoff's Prinzip, 6

Known plaintext attack, 6

Layer 8, 24

OSI-Modell, 19

Private-Key-Kryptographie, 5

Restrisiko, 4

Risiko, 4

Risikoanalyse, 4

Secret Key, 5

Sensitive Daten, 22

Sicherheit, 4

Social Engineering, 24

SSL, 22

Symmetrische Kryptographie, 5

TCP Verbindungsaufbau, 20

Threat vs. Threat Agent, 19

TLS, 22

Verbindlichkeit, 4

Verfügbarkeit, 3

Vertraulichkeit, 4

Zugangskontrolle, 3

Zugriffskontrolle, 3

Zutrittskontrolle, 3

## Abbildungsverzeichnis

1	Wissenspyramide (Wikipedia) . . . . .	3
2	Zeitpunkt Entdeckung eines Grundziel-Verlustes . . . . .	4
3	Alice verschlüsselt, Bob entschlüsselt mit dem gemeinsamen Schlüssel . . . . .	5
4	Nur Geheimtext . . . . .	6
5	Klartext-Geheimtext-Paare . . . . .	6
6	Klartexte und Geheimtexte . . . . .	6
7	Caesar cipher mit Verschiebung um 3 Stellen . . . . .	7
8	Frequenzanalyse unchiffriert . . . . .	7
9	Frequenz um 10 Stellen verschoben . . . . .	8
10	Vigenère cipher . . . . .	8
11	Funktionsweise des OTP . . . . .	9
12	einfaches Beispiel einer Hashfunktion . . . . .	9
13	Signatur im Detail . . . . .	12
14	Alice verschickt eine signierte Nachricht an Bob . . . . .	13
15	Eve als man in the middle . . . . .	13
16	Alice vertraut Bob durch direkte Überprüfung . . . . .	13
17	Alice vertraut Dave indirekt durch Vertrauensnetz . . . . .	14
18	Hierarchical Trust durch Certificate Authorities . . . . .	14
19	HTTP zustandslos . . . . .	16
20	HTTP Zustand per Request (hidden field) . . . . .	16
21	Einsatz eines Cookies . . . . .	16
22	Cross-Site Request Forgery . . . . .	18
23	OSI-Layers . . . . .	19
24	OSI vs. Internet Reference Model . . . . .	19
25	TCP-Encapsulation . . . . .	20
26	ARP Spoofing . . . . .	20
27	3-Way Handshake . . . . .	21
28	SYN Flood . . . . .	21
29	DRDoS . . . . .	21
30	SSL/TLS . . . . .	22
31	XSS . . . . .	23
32	Code Injection . . . . .	23
33	Beispiel Code Injection . . . . .	23
34	Layer 8 . . . . .	24