Juventus Technikerschule HF

Hacking Exposed

LAN-Security & Kryptographische Grundlagen

Ahmet Inci

Inhalt Lan Security

Network Scanning

Machine-In-The-Middle Attacke

ARP-Spoofing

DNS-Spoofing

Abwehrmassnahmen

Ziele

Sie kennen Techniken zur Netzwerk Scanning und Host Discovery

Sie wissen was eine MITM-Attacke ist und können deren Eigenschaften beschreiben und diese im LAN durchführen.

Sie kennen ARP- und DNS-Spoofing

#01 Network Scanning

Host Discovery

- Ein Port-Scan ist teuer und aufwendig und unnötig, wenn ein Zielsystem gar nicht online ist.
- Host Discovery erfüllt die Aufgabe, herauszufinden ob ein Zielsystem überhaupt erreichbar ist.
- Findet vor dem Port-Scan statt.

Nmap Default Host Discovery – vor Port Scan

Nmap führt standardmässig ein Host Discovery durch um Anzahl zu scannender Host auf interessante zu reduzieren:

1)ICMP Echo Request (Ping)

2)TCP SYN Paket auf Port 443

3)TCP ACK Paket auf Port 80

4)ICMP Timestamp Request

- Weitere Techniken und Informationen in der Manual Page:

–man nmap

```
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
```

Host Discovery im LAN

- Schnell alle Geräte die Online sind im eigenen Netzwerk finden:
  - $ sudo nmap -sn 192.168.1.0/24
  - Ping Scan ohne Port Scan
  - Im lokalen Netzwerk wird ein ARP scan verwendet!

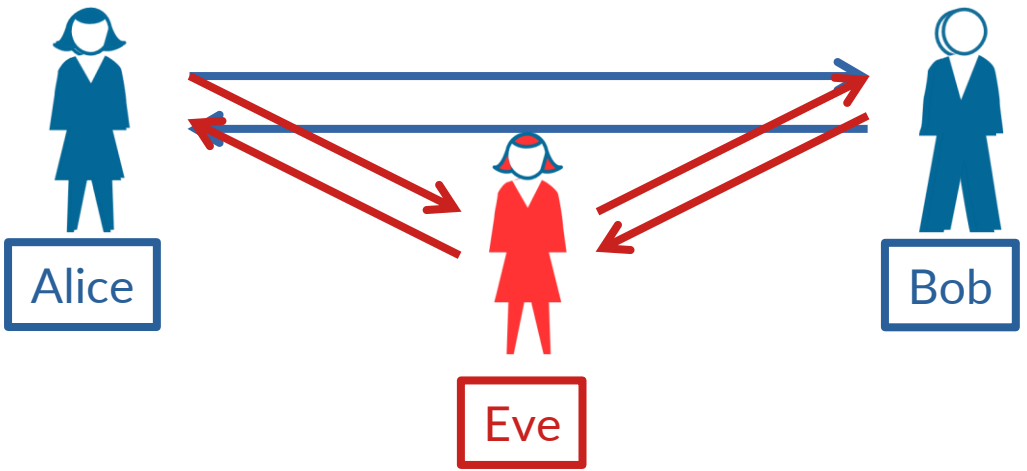
```
pascal@0x470:~$ nmap -sn 192.168.105.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-23 21:25 CEST
Nmap scan report for _gateway (192.168.105.1)
Host is up (0.0034s latency).
Nmap scan report for pihole (192.168.105.53)
Host is up (0.0084s latency).
Nmap scan report for 192.168.105.114
Host is up (0.11s latency).
Nmap scan report for 0x470.home (192.168.105.118)
Host is up (0.00013s latency).
Nmap scan report for sonos_wohnzimmer.home (192.168.105.150)
Host is up (0.0037s latency).
Nmap scan report for sonos_buero.home (192.168.105.151)
Host is up (0.059s latency).
Nmap scan report for sonos_bad.home (192.168.105.152)
Host is up (0.057s latency).
Nmap scan report for sonos_kueche.home (192.168.105.153)
Host is up (0.061s latency).
Nmap done: 256 IP addresses (8 hosts up) scanned in 4.78 seconds
```

#02 MITM-Attacke

Machine-in-the-Middle

MITM

- Jede Form von Angriff, in der Netzwerkverkehr zwischen Alice und Bob von Eve mitgelesen und/oder verändert werden kann.



Alice, Bob, Eve und Mallory

- Alice und Bob sind zwei Personen die miteinander kommunizieren.
- Eve ist eine dritte Person, welche die Kommunikation zwischen Alice und Bob passiv belauscht. (engl. eavesdropper)
- Mallory ist wie Eve, verändert aber aktiv die Kommunikation. (engl. malicious)

https://de.wikipedia.org/wiki/Alice_und_Bob

#03 ARP-Spoofing

Recap: MAC-Adresse

- Rechneradressierung auf Sicherungsschicht (OSI Layer 2)
- Besteht aus 6 Oktetts à 8 Bit (48 Bit)
  - Organisationally Unique Identifier (OUI) Byte 0 bis 2
- https://www.wireshark.org/tools/oui-lookup.html
  - Network Interface Controller (NIC) Byte 3 bis 5

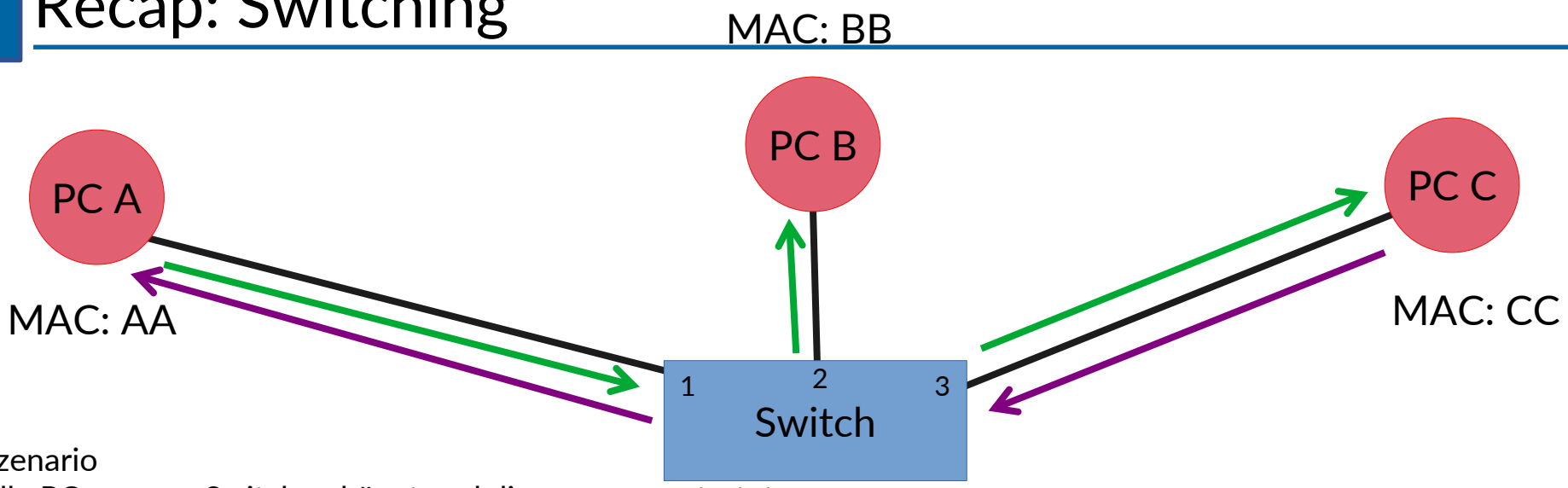
Recap: Kommunikation im LAN

- Client IP
  - 192.168.42.23/24
- Subnetz
  - 255.255.255.0
- Server IP
  - 192.168.42.45
- Sind erste 24 Bit identisch?

- Client IP
  - 11000000.10101000.00101010.00010111
- Subnetz
  - 11111111.11111111.11111111.00000000
- Server IP
  - 11000000.10101000.00101010.00101101

IP Routing Explained:
https://www.youtube.com/watch?v=8qtKpZGoNdI (11 min.)

Recap: Switching



Szenario
Alle PCs neu an Switch gehängt und diesen neu gestartet

1) PC A sendet PC C eine Nachricht:
MAC Src: AA | MAC Dst: CC

2) Switch weiss nicht wo MAC Dst: CC ist und schickt Message an alle Ports

3) PC B prüft MAC Dst der Message mit eigener MAC und verwirft

4) PC C prüft MAC Dst der Message mit eigener MAC und antwortet
MAC Src: CC | MAC Dst: AA

5) Switch prüft Tabelle, findet MAC Dst auf Port 1 und stellt Message an PC A zu

Port	MAC
1	AA
3	CC

Switching Fundamentals
https://www.youtube.com/watch?v=chlBlRfsnq0 (8 Min.)

Woher weiss PC A die MAC-Adresse von PC B?

- Über eine flüchtige Tabelle die IP-Adressen auf MAC-Adressen mappt.
- Das ARP-Protokoll löst IP-Adressen nach MAC-Adressen auf
- Damit diese Namensauflösung nicht immer neu gemacht werden muss, werden die Ergebnisse in der ARP-Tabelle des PCs gecached

Beispiel 1 ARP via Ping im lokalen Netz

```
pascal@0x470:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.105.1            ether   d8:58:d7:00:8f:ca   C                     wlp4s0
192.168.105.53           ether   f2:c2:f4:12:ef:d8   C                     wlp4s0
```

```
pascal@0x470:~$ ping 192.168.105.151 -c 1
PING 192.168.105.151 (192.168.105.151) 56(84) bytes of data.
64 bytes from 192.168.105.151: icmp_seq=1 ttl=64 time=5.39 ms

--- 192.168.105.151 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.391/5.391/5.391/0.000 ms
```

```
pascal@0x470:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.105.1            ether   d8:58:d7:00:8f:ca   C                     wlp4s0
192.168.105.53           ether   f2:c2:f4:12:ef:d8   C                     wlp4s0
192.168.105.151          ether   48:a6:b8:12:22:50   C                     wlp4s0
```

Beispiel 2 ARP via Ping ins Internet

```
pascal@0x470:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.105.1            ether   d8:58:d7:00:8f:ca   C                     wlp4s0
192.168.105.53           ether   f2:c2:f4:12:ef:d8   C                     wlp4s0
192.168.105.151          ether   48:a6:b8:12:22:50   C                     wlp4s0
```

```
pascal@0x470:~$ ping 8.8.8.8 -c 1
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=107 time=2.47 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.466/2.466/2.466/0.000 ms
```

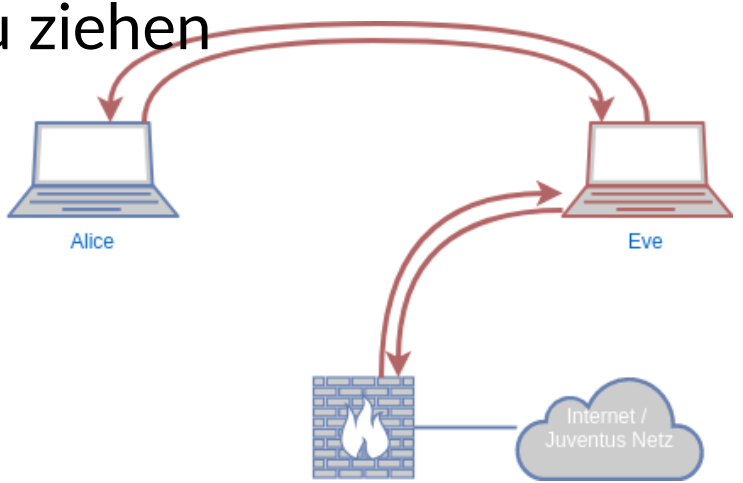
```
pascal@0x470:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.105.1            ether   d8:58:d7:00:8f:ca   C                     wlp4s0
192.168.105.53           ether   f2:c2:f4:12:ef:d8   C                     wlp4s0
192.168.105.151          ether   48:a6:b8:12:22:50   C                     wlp4s0
```

ARP Protokoll – im eigenen Subnetz

- ARP ist ein Broadcasting Namenssystem
  - Self-Managed: kein Master-Node notwendig
  - Einfach zu implementieren
  - Lokationsunabhängig
  - Skaliert nicht: Kommunikation wächst mit der Anzahl Teilnehmer
  - Einfach auszunutzen
- Broadcast Nachricht: «Was ist die MAC-Adresse zu der IP-Adresse x? Antwort bitte an y»

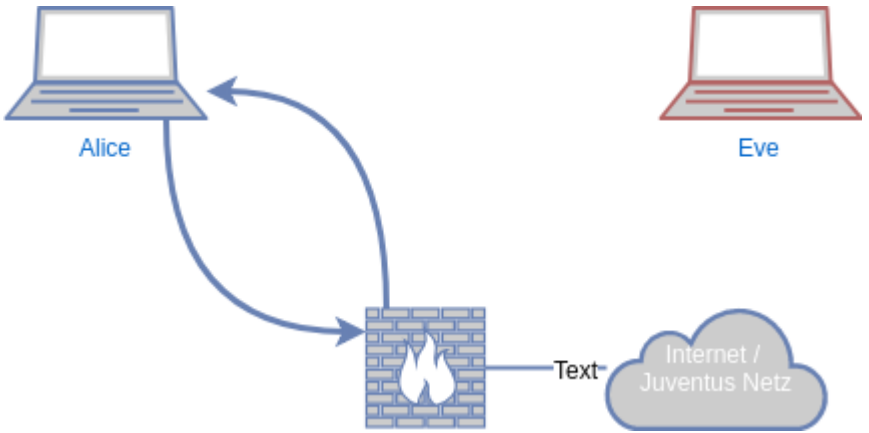
ARP Spoofing

- MITM durch Vergiften von ARP Einträgen bei zwei oder mehreren Rechnersystemen
- Auch ARP-Poisoning genannt
- Ziel ist sämtlichen Datenverkehr an sich zu ziehen
- Einsatzort: LAN
  - Café, Bibliothek, Schule, Zuhause etc.



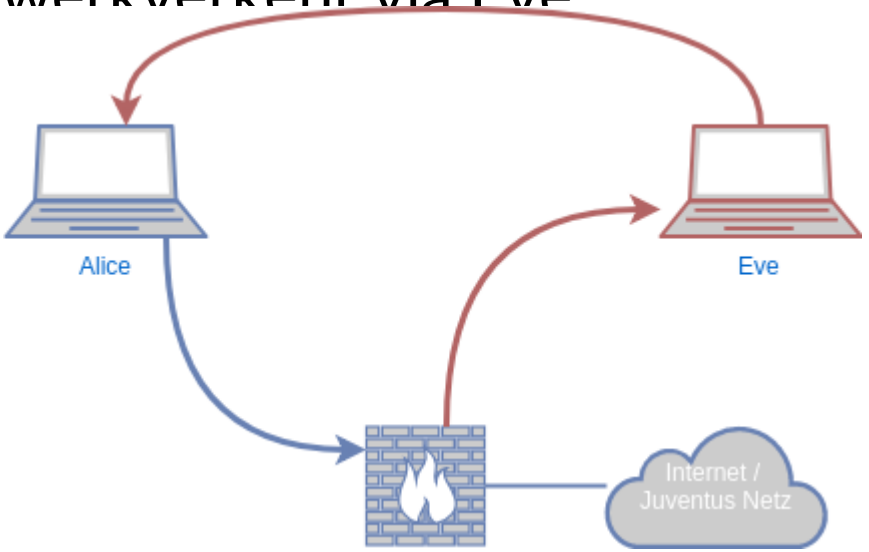
ARP Poisoning

- Angreifer sendet gefälschte ARP-Pakete an Alice und Bob und leitet so den Netzwerkverkehr über sich



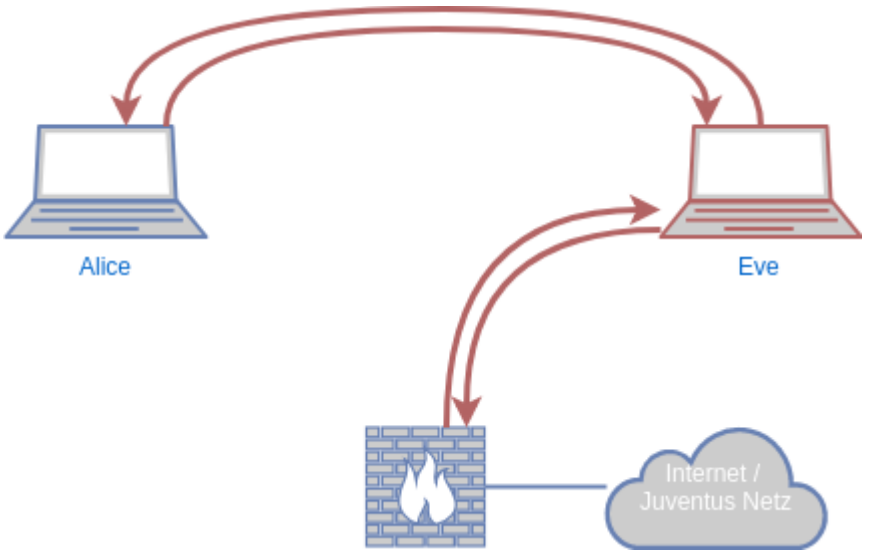
ARP Poisoning Phase 1

- Eve teilt dem Gateway per ARP-Reply die eigene MAC-Adresse für die IP von Alice mit
  - Asynchroner Netzwerkverkehr via Eve



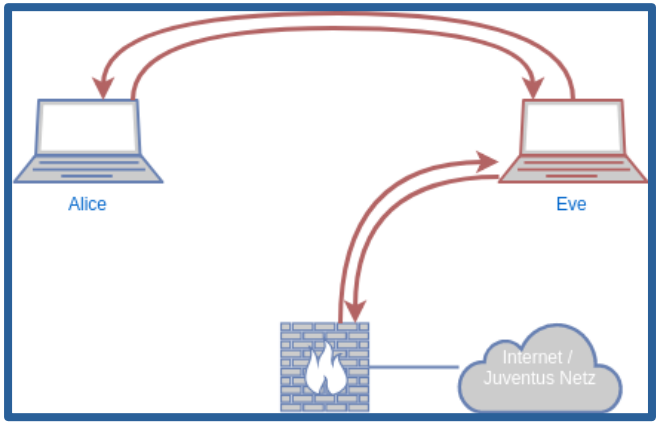
ARP Poisoning Phase 2

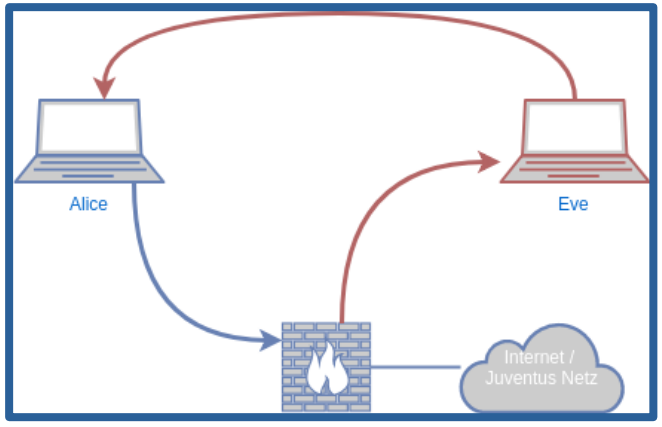
- Eve teilt Alice per ARP-Replay die eigene MAC-Adresse für die IP vom Default-Gateway mit.
  - Synchroner Netzwerkverkehr über Eve → MITM abgeschlossen

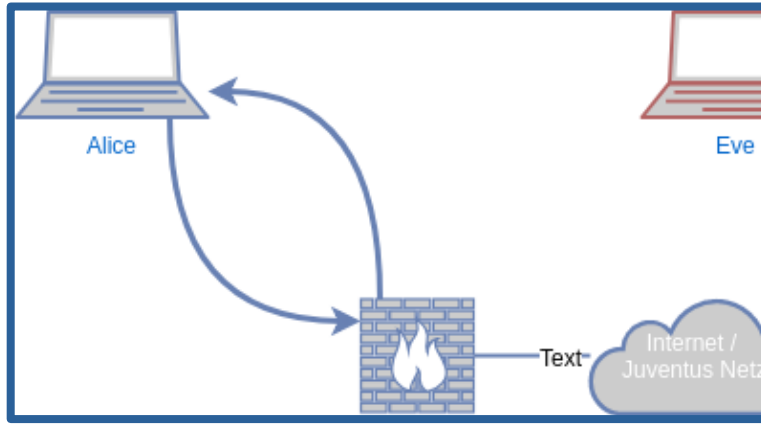


ARP Detoxing

- Um möglichst unbemerkt zu bleiben, die ursprüngliche ARP Tabelle wiederherstellen.







ARP Spoofing erkennen

- Manuell lässt sich ARP Spoofing anhand der ARP-Tabelle erkennen.

– $ arp -n

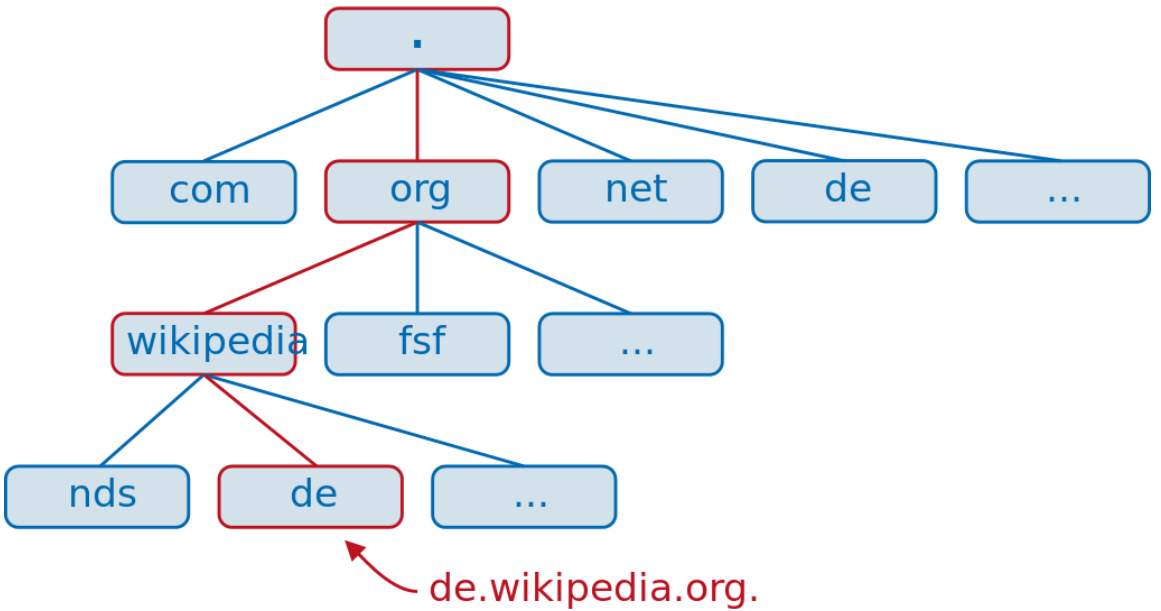
```
Address                  HWtype  HWaddress           Flags Mask            Iface
35.222.85.5                      (incomplete)                              enp0s31f6
192.168.1.100            ether   88:1f:a1:40:c8:18   C                     wlp4s0
192.168.1.1              ether   08:00:27:1c:ef:3e   C                     wlp4s0
192.168.1.116            ether   08:00:27:1c:ef:3e   C                     wlp4s0
35.224.99.156                    (incomplete)                              enp0s31f6
```

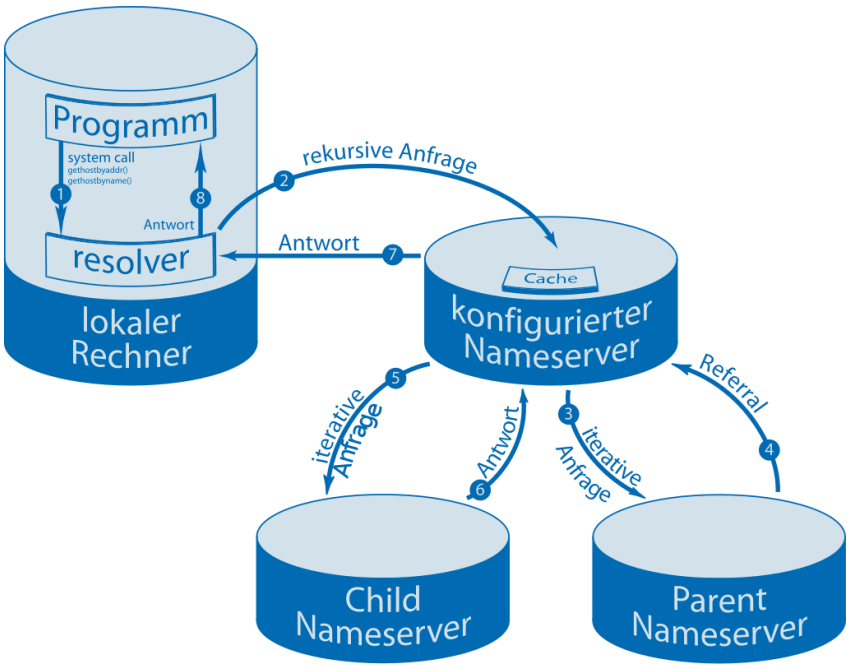
#04 DNS Spoofing

Weitere Attacken im LAN

Namensauflösung im Internet

- DNS hierarchische Namensauflösung





DNS Protokoll

- UDP basiertes Protokoll
  - Kein Verbindungsaufbau → anfällig für Spoofing-Attacken
- Client erhält via DHCP die IP-Adresse des lokalen DNS-Resolvers
- Sobald ein Domainname in eine IP übersetzt werden muss, wird eine Anfrage an den Resolver geschickt

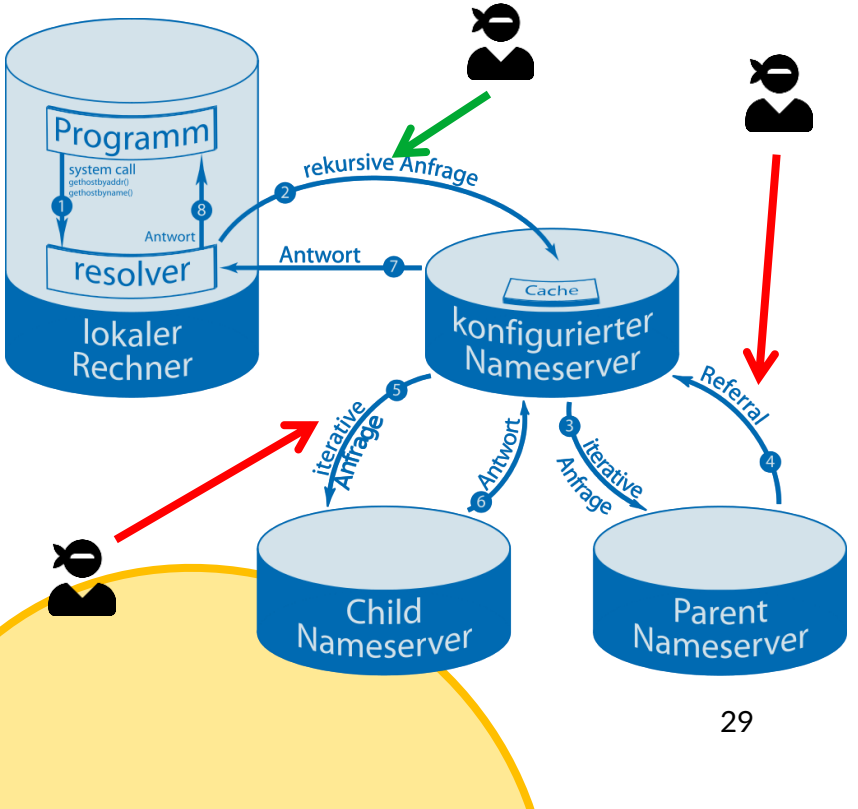
Source	Destination	Protocol	Info
192.168.100.100	8.8.8.8	DNS	Standard query 0x18b2 A www.heise.de OPT
8.8.8.8	192.168.100.100	DNS	Standard query response 0x18b2 A www.heise.de A 193.99.144.85 OPT

DNS Spoofing

- IP-Adresse im DNS-Response wird gefälscht
- Voraussetzung: DNS-Anfragen kommen bei Angreifer vorbei (MITM)
  - ARP-Spoofing
  - DHCP-Spoofing

DNSSEC

- Ist DNSSEC eine Schutzmassnahme gegen DNS-Spoofing-Attacken?
- DNSSEC ist eine wichtige Technologie zur Absicherung von Web- und Internetservices



DNSSEC in der Schweiz
https://www.youtube.com/watch?v=_rSWTQ9LGCE

#05 Abwehrmassnamen

Abwehrmassnamen im LAN

- Einfachste Massnahme ist, den Datenverkehr zu verschlüsseln
  - Transportverschlüsselung
- TLS, VPN, SSH, Tor, moderne Messenger wie Threema, Signal
  - Ende-zu-Ende-Verschlüsselung
- Passwort-ZIP, S/MIME und PGP, moderne Messenger wie Threema, Signal

Videoempfehlungen fürs Selbststudium

- Du kannst alles hacken – du darfst dich nur nicht erwischen lassen. (57 Min)

Inhalt Kryptographische Grundlagen

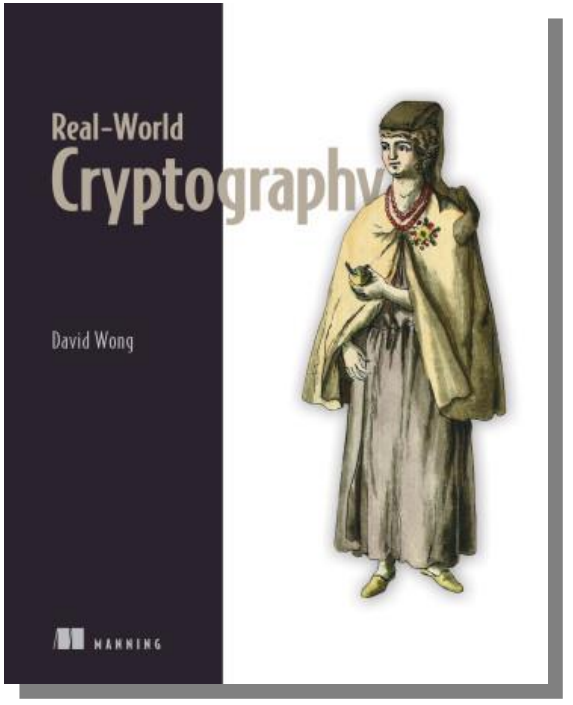
- Passwörter
- Kryptographische Stärke messen
- Kryptographische Grundlagen
  - Symmetrische Verschlüsselung
  - Hash-Funktionen
  - Asymmetrische Verschlüsselung
  - Digitale Signaturen

Ziele

- Sie können die Sicherheit von Passwörtern abschätzen und Empfehlungen für sichere Passwörter abgeben.
- Sie können Aussagen über die stärke von kryptographischen Algorithmen machen.
- Sie kennen die Grundlage symmetrischer und asymmetrischer Kryptographie sowie verschiedene Verfahren.

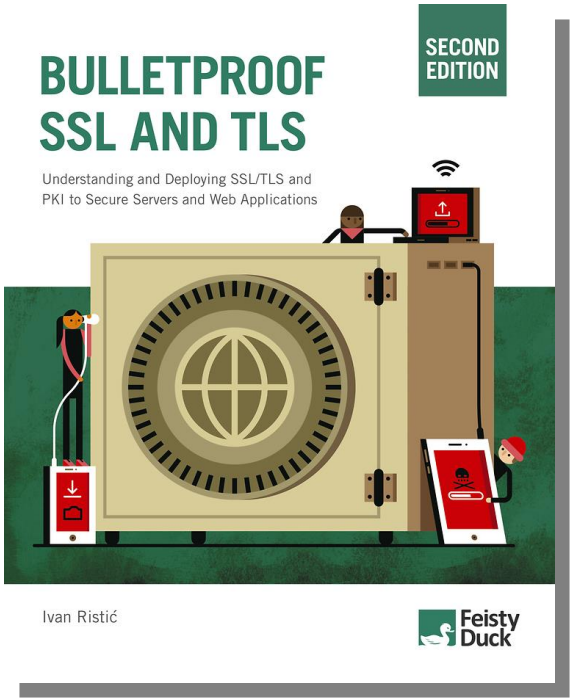
Literaturempfehlung

- Real-World Cryptography
  - David Wong, @cryptodavidw
  - September 2021
  - ISBN: 978-1-61729-671-0



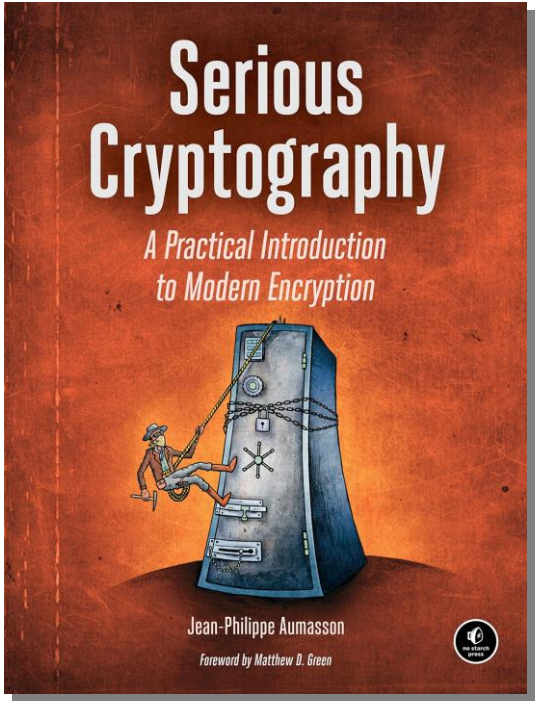
Literaturempfehlung

- Bulletproof SSL and TLS
  - Ivan Ristić, @ivanristic
  - 2. Auflage, Februar 2022
  - ISBN: 978-1907117091
  - Inhaltsverzeichnis



Literaturempfehlung

- Serious Cryptography
  - Jean-Philippe Aumasson, @veorq
  - November 2017
  - ISBN-13: 978-1-59327-826-7
  - Inhaltsverzeichnis



#01 Passwörter

Wie viele Möglichkeiten?

- Zahlenschloss
- Ziffern [0-9] (10)
- Stellen 4
- Kombinationen: $10^4 = 10'000$
- Stärke in Bit: $\lceil \log_2(10^4) \rceil = 14$ Bit

Knacken bei 2 Passwörter / Sekunde:
$10^4 * 0.5\ s / 3600\ s = 1.4\ h$

Wie viele Möglichkeiten?

- Passwort: 8 S R C M
- Ziffern [0-9] (10)
- Buchstaben gross [A-Z] (26)
- Stellen 5
- Kombinationen: $36^5 = 60'466'176$
- Stärke in Bit: $[\log_2(36^5)] = 26$ Bit

Knacken bei 350 Milliarden Passwörter[1] / Sekunde:
$10^4 / 3.5 * 10^{11} = 10^{-7}$ s respektive 100 ns

[1] GPU Cluster aus 2012!

Brute-Force-Attacke

- Durchtesten aller *n* möglichen Kombinationen
- Führt immer zum Erfolg (theoretisch)
  - Ist aber nicht zwingend der schnellste Angriff
- In 50% aller Angriffe, nur n/2 Versuche nötig

Rainbow Table und Salt

- Serverseitig werden Passwörter gespeichert
  - Nie in Klartext!
- Passwort wird gehashed und der Hash davon wird gespeichert.
  - Wieso?
- Dump einer Login-Datenbank kann mit einer Rainbow Table angegriffen werden.
  - Tabelle mit vorgerechneten Hashes von Passwörtern
  - Suchen nach dem Passwort-Hash => Passwort
- Ein Salt schützt die gehashten Passwörter

Sichere Passwörter

- Aus dem Ratgeber Eine kurze Anleitung zur digitalen Selbstverteidigung der Digitalen Gesellschaft:

- *«Ein hinreichend sicheres Passwort ist mindestens fünf zufällige Wörter oder zwölf Zeichen lang, beinhaltet Klein- und Grossbuchstaben, Zahlen und Sonderzeichen und lässt sich nicht herleiten aus personenbezogenen Angaben wie Name, Geburtstag oder Wohnort.»*

- Passwörter nicht mehrfach verwenden

Sichere Passwörter

- Passwort: 8 } / 2 M p x % L @ P }
- Ziffern [0-9] (10)
- Buchstaben gross [A-Z] (26)
- Buchstaben klein [a-z] (26)
- Sonderzeichen [!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~] (33)
- Stellen 12
- Kombinationen: $95^{12} = 5.403600877 \times 10^{23}$
- Stärke in Bit: $\lceil \log_2(95^{12}) \rceil = 79$ Bit

Knacken bei 350 Milliarden Passwörter[1] / Sekunde:
$95^{12} / (3.5 * 10^{11} * 3600 * 24 * 365) = 50194$ Jahre

[1] GPU Cluster aus 2012!

Sicherheitsfragen

- Werden oft zwecks Passwort-Recovery eingesetzt:
- «Wie hiess Ihr Haustier in der Kindheit», «Was ist der Mädchenname Ihrer Mutter», «Was ist Ihre Lieblingsfarbe» etc.
- Von der Verwendung solcher Mechanismen ist abzuraten, denn:
  - *«Der Zoo der Lieblingstiere ist im Zweifel sehr klein!»* - Linus Neumann
- Solche Felder am besten wie Passwortfelder behandeln und im Passwortmanager hinterlegen

Gute Passwörter halten lange, aber nicht ewig.

- UNIX-Prominenz wählte Schach-Eröffnung: 39 Jahre alte BSD-Passwörter geknackt Heise, 12.10.2019

#02 Kryptographische Grundlagen

Kryptographischer Algorithmus Designziel

- Kerckhoffs-Prinzip (1883):

– Die Sicherheit eines Verschlüsselungsverfahrens darf nur von der Geheimhaltung des Schlüssels abhängen, nicht von der Geheimhaltung des Algorithmus.

https://de.wikipedia.org/wiki/Kerckhoffs%E2%80%99_Prinzip

Auguste Kerckhoffs

Sicherheit eines kryptographischen Algorithmus

- Ein kryptographischer Algorithmus ist sicher, wenn der effizienteste Angriff nicht wesentlich schneller ist, als ein Brute-Force Angriff.
- Die Sicherheit eines kryptographischen Algorithmus hängt von der Anzahl möglicher Schlüssel ab: Je mehr mögliche Schlüssel desto sicherer.

Sicherheit von kryptographischen Systemen

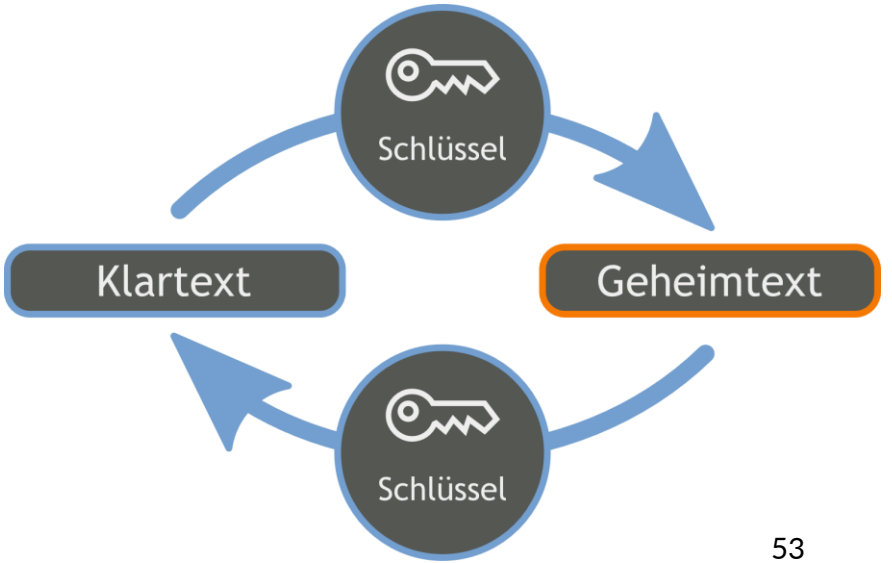
- Ein kryptographisches System ist nur so sicher wie seine schwächste Komponente

Ver- und Entschlüsselung allgemein

#03 Symmetrische Verschlüsselung

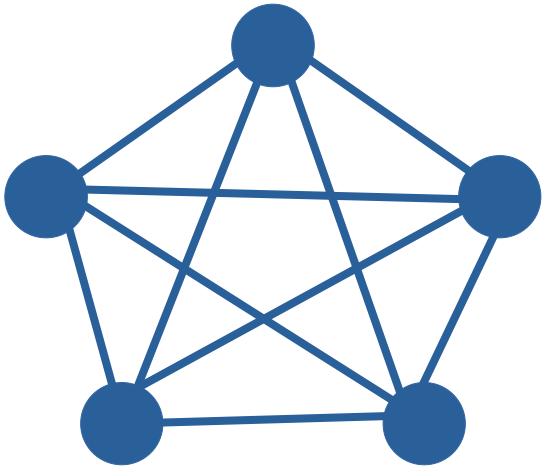
Symmetrische Kryptographie

- Ver- und Entschlüsselung mit demselben Schlüssel
- Sehr schnell
- ZIP-Passwortverschlüsselung

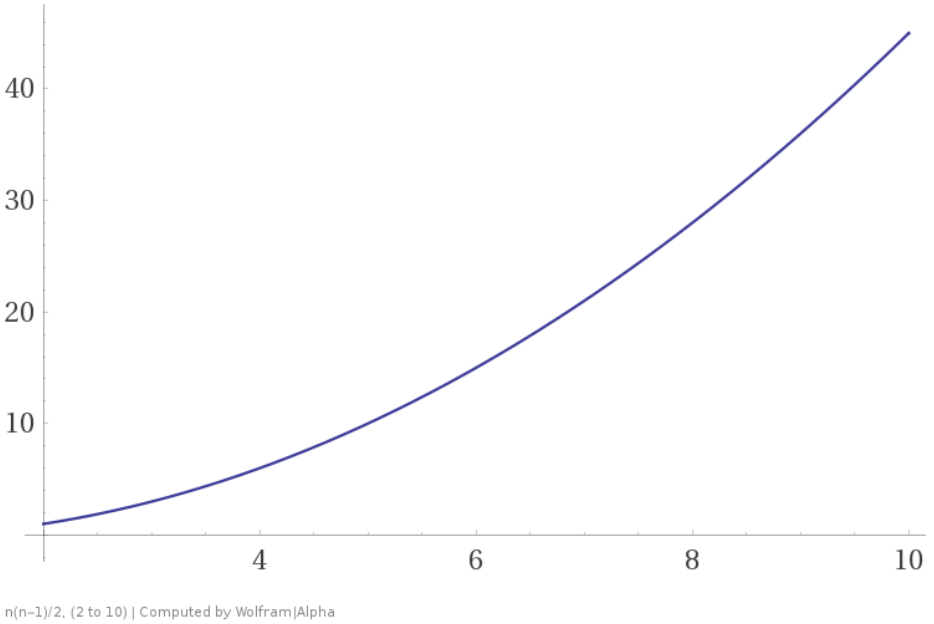


Anzahl Schlüssel in symmetrischem Kryptosystem

- 5 Personen möchten zueinander je einen sicheren Kanal aufbauen. Wie viele symmetrische Schlüssel gibt es in diesem System?
  - a) 32
  - b) 10
  - c) 5
  - d) 25
- Wie lautet die Formel zur Berechnung?
  - $n(n-1)/2$
  - $O(n^2)$



Anzahl Schlüssel im System wächst Quadratisch



https://www.wolframalpha.com/input/?i=n%28n-1%29%2F2%2C+%282+to+10%29

Symmetrische Algorithmen

Algorithmus	Schlüssellänge (Bit)	Stärke (Bit)
DES	56	56
3DES	168	112*
AES	128, 192, 256	128, 192, 256
Camellia	128, 192, 256	128, 192, 256

* Meet-in-the-Middle Attacke

#04 Hash-Funktionen

Hash-Verfahren

- Sind Einwegfunktion mit fixen Outputlängen
- Für beliebiges X erzeugt die Hashfunktion H den Output h
  - H(X) → h
  - sha256sum kali-linux-2018.2-amd64.iso
  - 56f677e2edfb2efcd0b08662ddde824e254c3d53567ebbbcdbbf5c03efd9bc0f56 f677e2edfb2efcd0b08662ddde824e254c3d53567ebbbcdbbf5c03efd9bc0f

Hash-Funktionen in Verwendung

- Einige Beispiele
  - Checksummenberechnung
  - Passwort-Ablage
  - digitale Signaturen

Hash-Verfahren

Hash-Verfahren	Hash Output Länge (Bit)	Stärke (Bit)
SHA-1	160	80
SHA-2, SHA-3	224	112
SHA-2, SHA-3	256	128
SHA-2, SHA-3	512	256

Datenleck: Hacker bietet Daten von zwei Escort-Foren zum Verkauf an → MD5
Golem, 11.10.2019

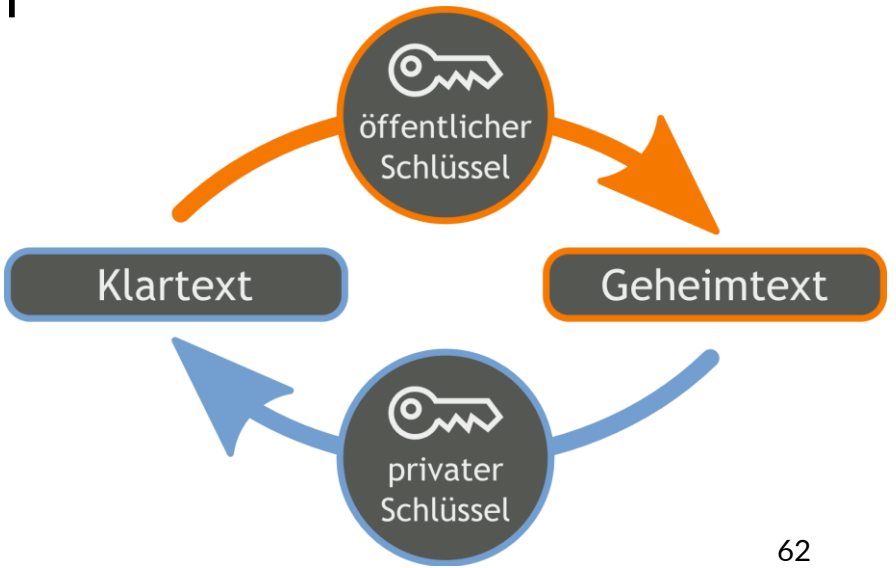
Passwort-Hashing richtig machen

Wie sicher sind gehashte Passwörter?

#05 Asymmetrische Verschlüsselung

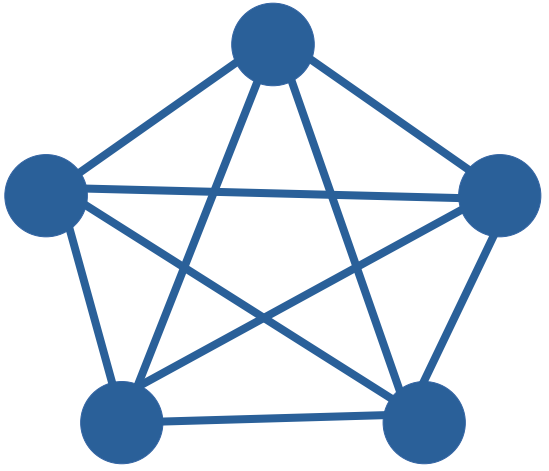
Asymmetrische Kryptographie

- Ver- und Entschlüsselung unterschiedliche Schlüssel
- Öffentlicher Schlüssel ist allen bekannt
- Privater Schlüssel muss geheim bleiben
- Langsamere Ver- und Entschlüsselung
- Email-Verschlüsselung



Anzahl Schlüssel in asymmetrischem Kryptosystem

- 5 Personen möchten zueinander je einen sicheren Kanal aufbauen. Wie viele öffentliche Schlüssel gibt es in diesem System?
  - a) 32
  - b) 10
  - c) 5
  - d) 25
- Wie viele private Schlüssel?



Asymmetrische Algorithmen

Algorithmus	Schlüssellänge (Bit)	Stärke (Bit)
RSA / DSA / Diffie-Hellman	1024	80
	2048	112
	3072	128
	7680	192
	15360	256

Forscher vermelden neuen Rekord beim Knacken von 795-Bit RSA Schlüssel
Heise, 4.12.2019

#06 Digitale Signaturen

Beispiel digitale Signatur erstellen

- Alice möchte eine Nachricht M digital signieren
  - Hashed Nachricht M mit Hashverfahren H zu Hash h
  - Verschlüsselt Hash h mit privatem Schlüssel $A_{PrivKey}$ zu Signatur S_M
- Sendet Nachricht M und Signatur S_M an Bob

Beispiel digitale Signatur prüfen

- Bob möchte Unverändertheit und Authentizität von Nachricht M mit Signatur S_M prüfen
  - Hashed Nachricht M mit Hashverfahren H zu Hash h_1
  - Entschlüsselt Signatur S_M mit öffentlichem Schlüssel A_{PubKey} zu Hash h_2
- $h_1 = h_2$
  - Wahr: Nachricht wurde nicht verändert und stammt von Alice
  - Falsch: Nachricht wurde verändert und/oder stammt nicht von Alice

Übungen & Labor

Übungen: Lan Security / Krypto

Labor: https://github.com/hexposed/Lab

Weiterführende Literaturempfehlung

- Kryptographie in der IT - Empfehlungen zu Verschlüsselung und Verfahren
- Verschlüsselung im Web mit der Web Crypto API
- Verschlüsseln mit elliptischen Kurven

Videoempfehlungen für's Selbststudium

- Immer diese verfluchten Passwörter (47 Min)
- Krypto knacken für Anfänger (45 Min)
- Kryptographie nach Snowden (57 Min)
- Krypto für die Zukunft (31 Min)
- Attacking end-to-end email encryption (60 Min)