



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Kapitel 3:

Technische Schwachstellen und Angriffe

1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

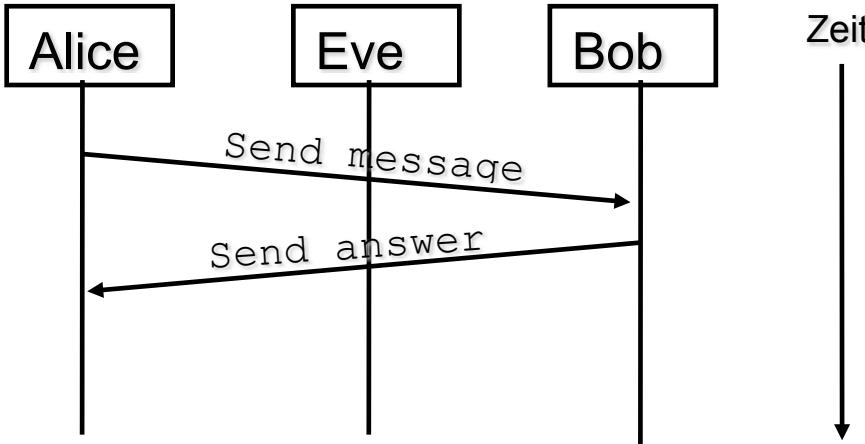
Aktueller Vorfall:

Cisco IOS XE: Aktive Ausnutzung einer Zero-Day-Schwachstelle in Web-UI



- Cisco warnt am 16.10.23: CVE-2023-20198 mit CVSS-Score von 10/10
- Angreifer kann über Web-Benutzeroberfläche vollständige Kontrolle erlangen.
- (Noch) Kein Patch verfügbar
- Bericht 19.10.: Schwachstelle auf 40.000 Geräten ausgenutzt und Backdoors implementiert
 - Neustart entfernt Backdoor ABER Angreifer legt Kennung mit höchsten Berechtigungen an
- Patch steht am 22.10.23 zur Verfügung
- LRZ: Wie schaut es bei uns aus?
 - Web-UI auf allen Routern deaktiviert! 😞
- LRZ nicht aber Nutzer im MWN
 - Compromised Website Report der [Shadowserver Foundation](#) liefert Hinweis
 - Test zeigt Infektion -> Web-UI wurde deaktiviert

Handelnde Personen

- Um Sicherheitsprobleme und -protokolle zu erläutern, werden häufig die folgenden Personen verwendet:
- Die „Guten“:
 - **Alice (A)**
Initiator eines Protokolls
 - **Bob (B)**
antwortet auf Anfragen von Alice
 - **Carol (C) und Dave (D)**
sind ggf. weitere gutartige Teilnehmer
 - **Trent (T)**
Vertrauenswürdiger Dritter
(Trusted third party)
 - **Walter (W)**
Wächter (Warden),
bewacht insb. Alice und Bob
- Die „Bösen“:
 - **Eve (E)**
(Eavesdropper)
Abhörender / passiver Angreifer
 - **Mallory, Mallet (M)**
(Malicious attacker)
Aktiver Angreifer
- Bsp.: Abhören der Kommunikation zwischen A und B
(UML Sequence Diagram)
 

```

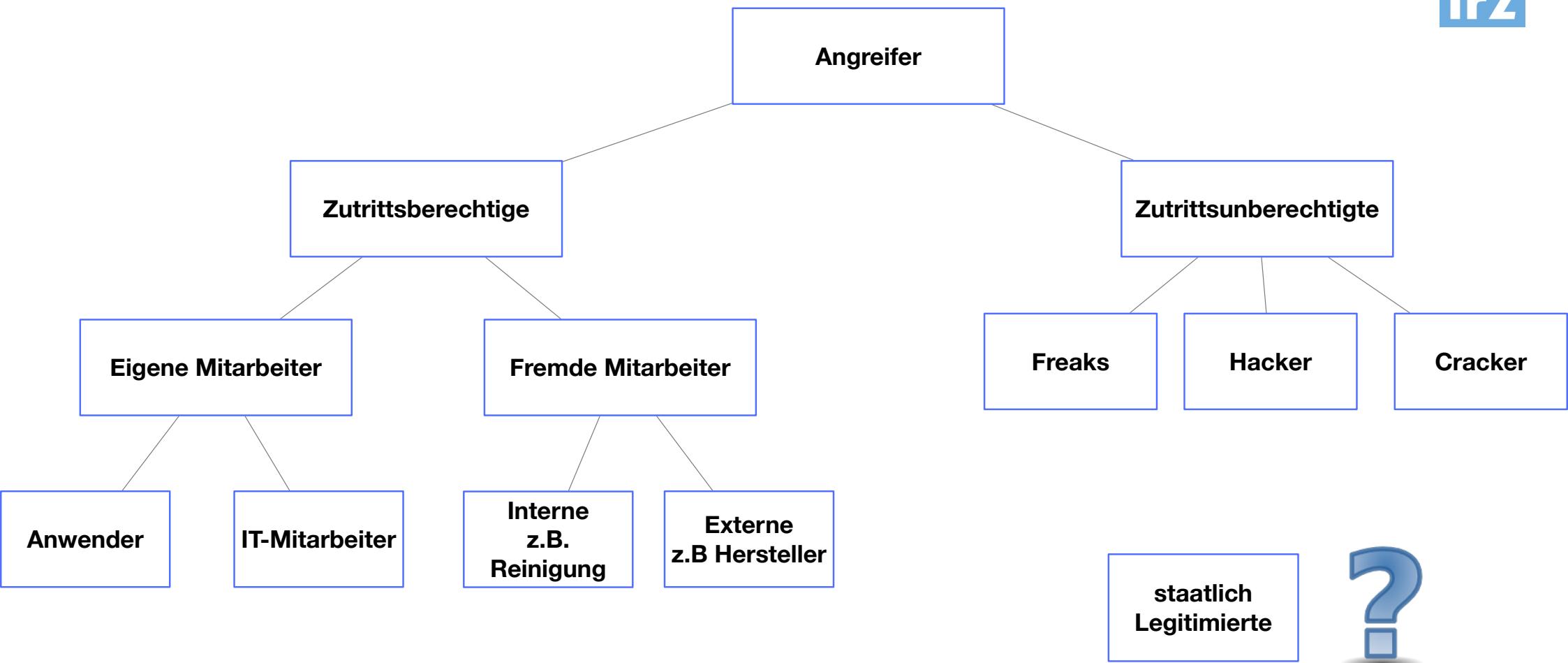
sequenceDiagram
    participant Alice
    participant Eve
    participant Bob
    Alice->>Bob: Send message
    Bob-->>Alice: Send answer
    Eve-->>Bob: 
  
```

The diagram illustrates a sequence of interactions between three participants: Alice, Eve, and Bob. Alice initiates a message exchange by sending a "Send message" to Bob. Bob responds by sending a "Send answer" back to Alice. Eve is positioned between Alice and Bob, indicating her role as an eavesdropper who can intercept the communication between them.

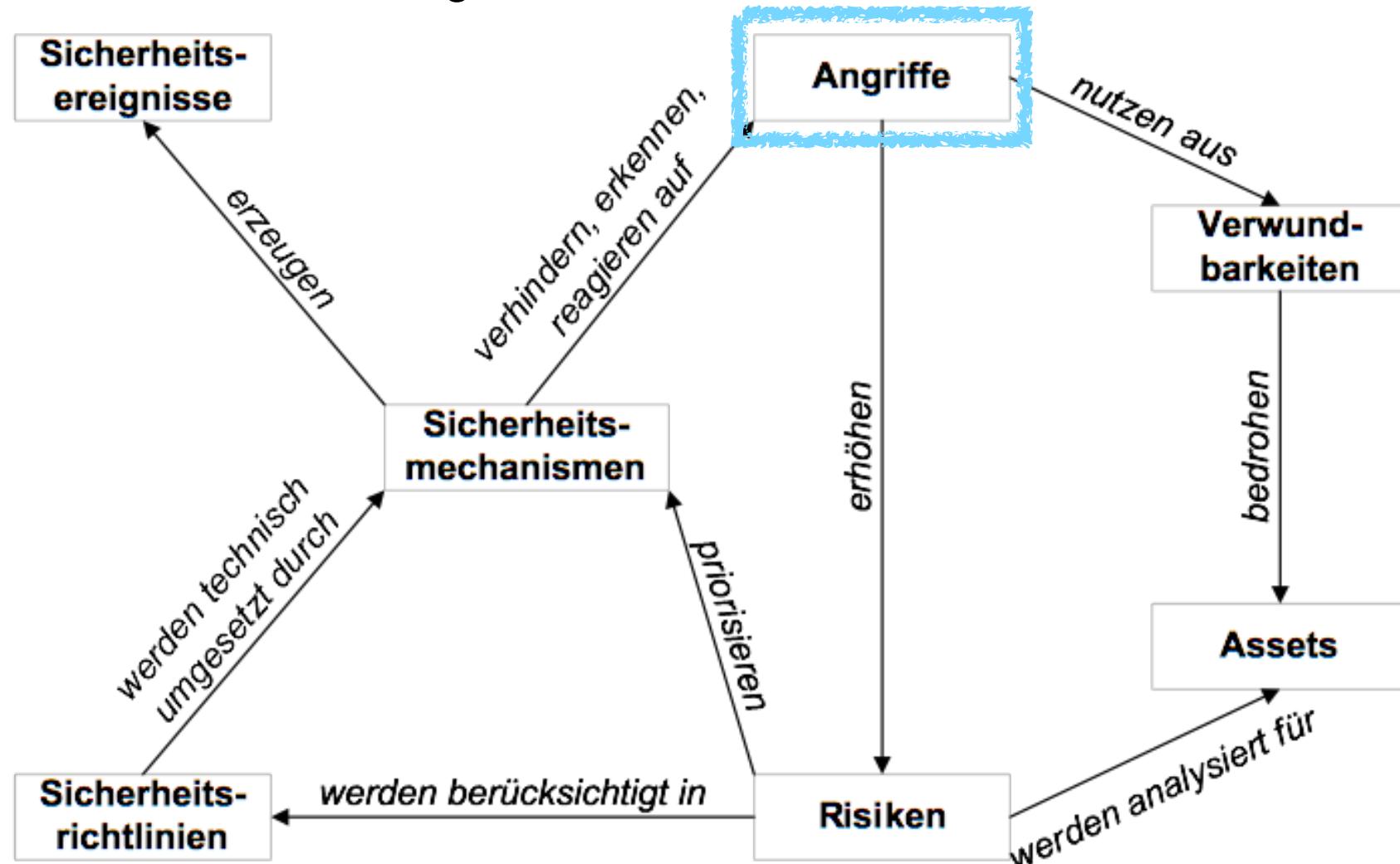
Angreifermodelle

- Antwort auf: Was können/machen Eve, Mallory und Mallet?
- Angreifermodell umfasst insbesondere Angaben zu
 - **Position des Angreifers**
 - Innentäter
 - Besucher, Einbrecher, ...
 - Internet / extern
 - **Fähigkeiten des Angreifers** (= Wissen + finanzielle Möglichkeiten), z.B. bei
 - experimentierfreudigen Schülern und Studierenden :-)
 - Fachleuten mit praktischer Erfahrung
 - erfahrenen Industriespionen / Geheimdiensten
 - **Motivation bzw. Zielsetzung des Angreifers**, z.B.
 - Spieltrieb, Geltungsbedürfnis, Vandalismus
 - Geld
 - Politischer oder religiöser Fanatismus, vermeintlicher Patriotismus
 - Spezifische **Charakteristika durchgeföhrter Angriffe**, z.B.
 - passives Abhören des Netzverkehrs vs.
 - aktive Eingriffe in die Kommunikation

Tätertypisierung



Begriffe und Zusammenhänge



1. Grundlegendes zur Angriffsanalyse

- Notation von Sicherheitsproblemen
- Angreifermodelle
- Begriffe und Zusammenhänge

2. Ausgewählte technische Angriffsvarianten

- Denial of Service (DoS und DDoS)
- Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
- E-Mail-Security (Spam)
- Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
- Web-basierte Angriffe (XSS, ...)
- Netzbasierte Angriffe (Sniffing, Portscans, ...)

3. Bewertung von Schwachstellen

- Common Vulnerability Scoring System (CVSS)
- Zero Day Exploits

Angriffsarten

- Erfolgreiche Angriffe haben negative Auswirkungen auf die
 - **Vertraulichkeit** (unberechtigter Zugriff auf Daten) und/oder
 - **Integrität** (Modifikation von Daten) und/oder
 - **Verfügbarkeit** (Löschen von Daten, Stören von Diensten)
- Eigenschaften zur Differenzierung von Angriffen sind z.B.:
 - **Ziel des Angriffs:** C, I und/oder A?
 - **Aktiv oder passiv** (z.B. remote exploit vs. sniffing)
 - **Direkt oder indirekt** (z.B. Manipulation einer Datenbank betrifft WebApp)
 - **Ein- oder mehrstufig** (z.B. kompromittierter Webserver als Sprungbrett)
- Angriffe sind unterschiedlich elegant und schwierig:
 - DDoS-Angriff zum Abschießen eines kleinen Webservers = trivial
 - Aufspüren und Ausnutzen bislang unbekannter Schwachstellen in Anwendungen = aufwendig

1. Grundlegendes zur Angriffsanalyse

- Notation von Sicherheitsproblemen
- Angreifermodelle
- Begriffe und Zusammenhänge

2. Ausgewählte technische Angriffsvarianten

- Denial of Service (DoS und DDoS)
- Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
- E-Mail-Security (Spam)
- Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
- Web-basierte Angriffe (XSS, ...)
- Netzbasierte Angriffe (Sniffing, Portscans, ...)

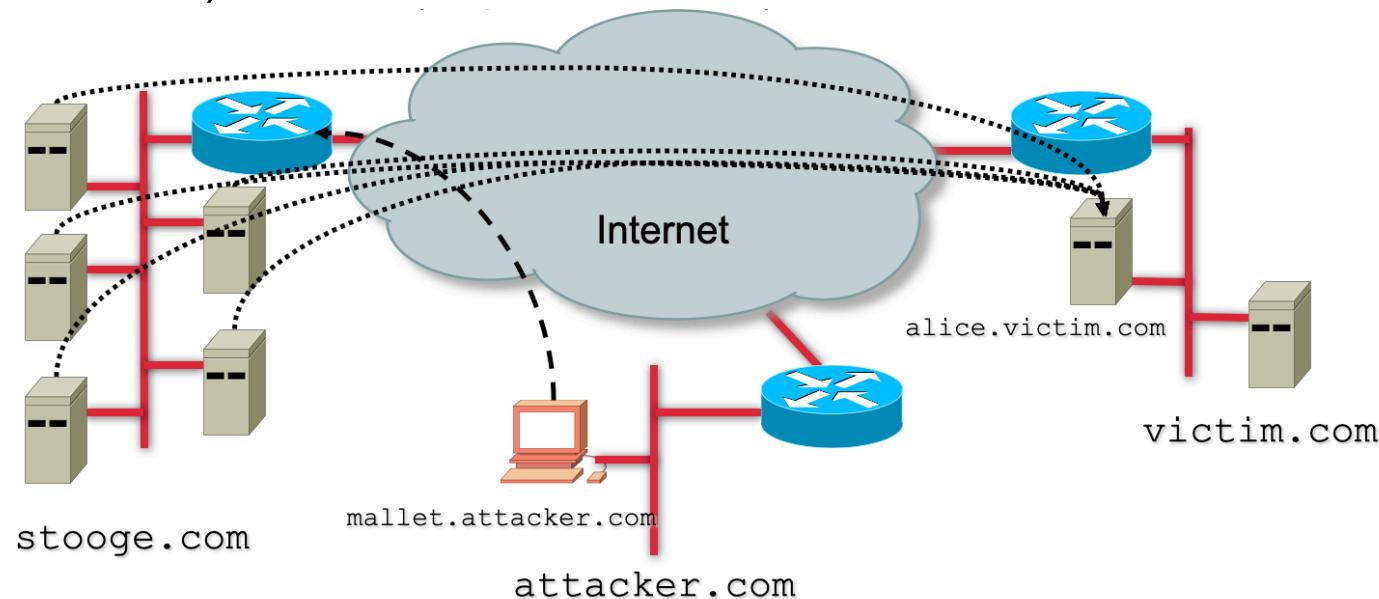
3. Bewertung von Schwachstellen

- Common Vulnerability Scoring System (CVSS)
- Zero Day Exploits

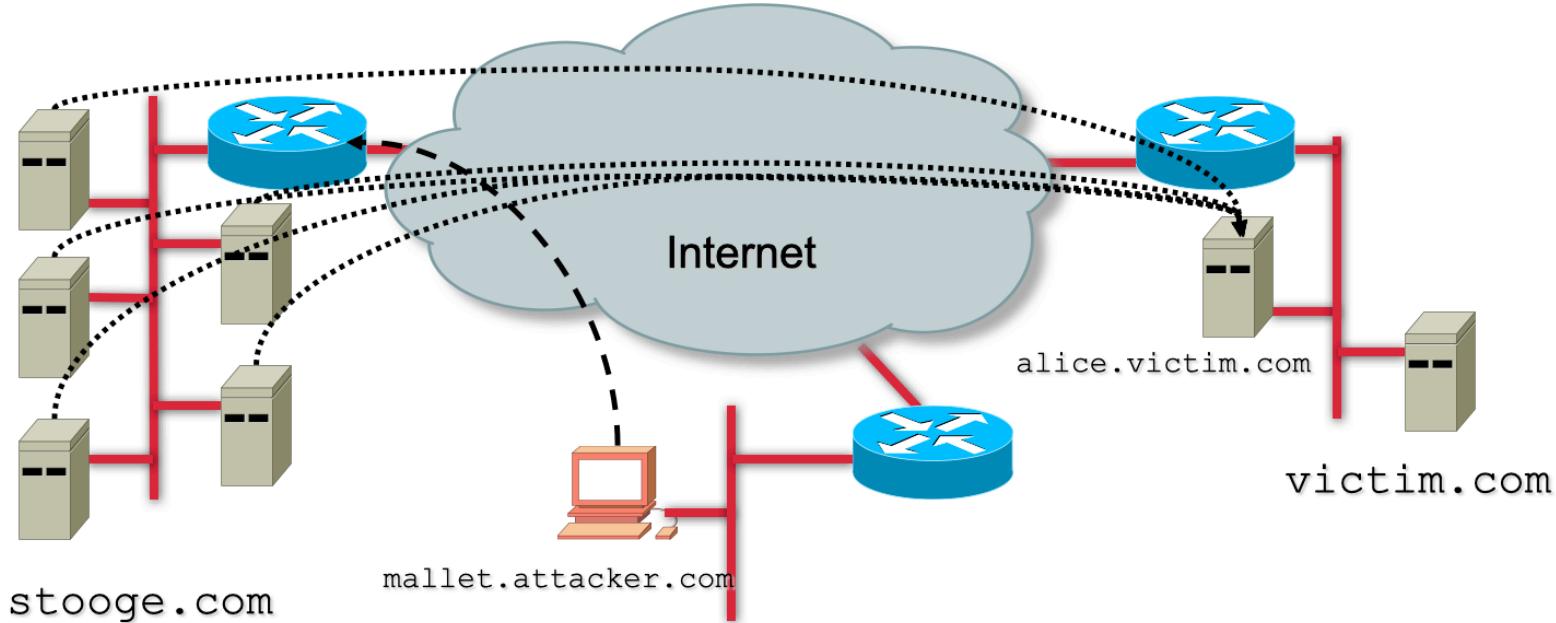
Denial of Service (DoS) und DDoS

- Angriff versucht, das Zielsystem oder Netz für berechtigte Anwender unbenutzbar zu machen, z.B. durch:
 - Überlastung
 - Herbeiführen einer Fehlersituation
 - Ausnutzung von Programmierfehlern oder Protokollschwächen, die z.B. zum Absturz führen
- Häufige Arten von DoS-Angriffen
 - Anforderung bzw. Nutzung beschränkter oder unteilbarer Ressourcen des OS (z.B. CPU-Zeit, Plattenplatz, Bandbreite,...)
 - Zerstörung oder Veränderung der Konfiguration
 - Physische Zerstörung oder Beschädigung
- Beispiel:
 - Überlasten eines Web-Servers durch massive Anfragen

- Angreifer sendet Strom von ping Paketen (ICMP) mit gefälschter Absender-Adresse (`alice.victim.com`) (Adressfälschung wird auch als IP-Spoofing bezeichnet) an IP-Broadcast Adresse von `stooge.com`
- Alle Rechner aus dem Netz von `stooge.com` antworten an `alice.victim.com` (Amplification attack)



Gegenmaßnahmen?



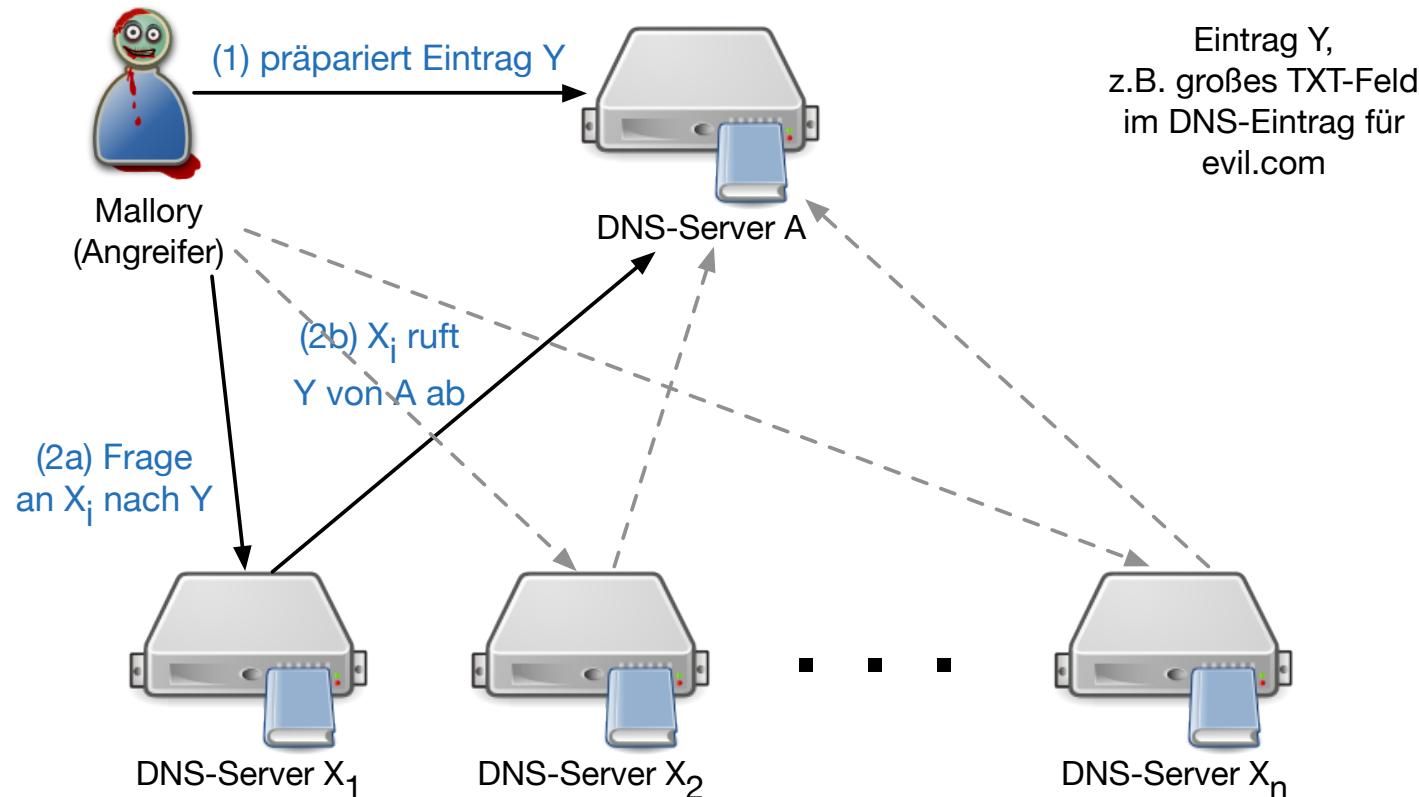
- Überkompensation:
ICMP oder IP-Broadcast am Router komplett deaktivieren
- Besser:
 - Server so konfigurieren, dass sie nicht auf Broadcast-Pings antworten
 - Router so konfigurieren, dass sie von außen an die Broadcast-Adresse gerichtete Pakete nicht weiterleiten

DNS Amplification Attack

- Begriffsbildung:
 - Domain Name System (Zuordnung von Namen zu IP-Adressen)
 - Kleines Paket des Angreifers führt zu großen Paket an Opfersystem
- Grundprinzip:
 - Sehr **kleines UDP-Paket zur Abfrage** des DNS-Servers (ca. 60 Byte)
 - Gefälschte Absenderadresse (i.A. die des DoS-Opfers)
 - **Antwort kann sehr groß werden** (bis theor. 3000 Byte)
 - Verstärkungsfaktor 50
 - Schmalbandiger Uplink reicht aus, um Multi-Gigabit Traffic zu erzeugen
- Historie:
 - Angriffe auf DNS-Root-Nameserver 2006
 - Seit Frühjahr 2012 häufige Scans nach DNS-Servern, wachsende Anzahl an Vorfällen; inzwischen größtenteils behoben, aber gallische Dörfer bleiben.
- Bsp: <http://blog.cloudflare.com/65gbps-ddos-no-problem>

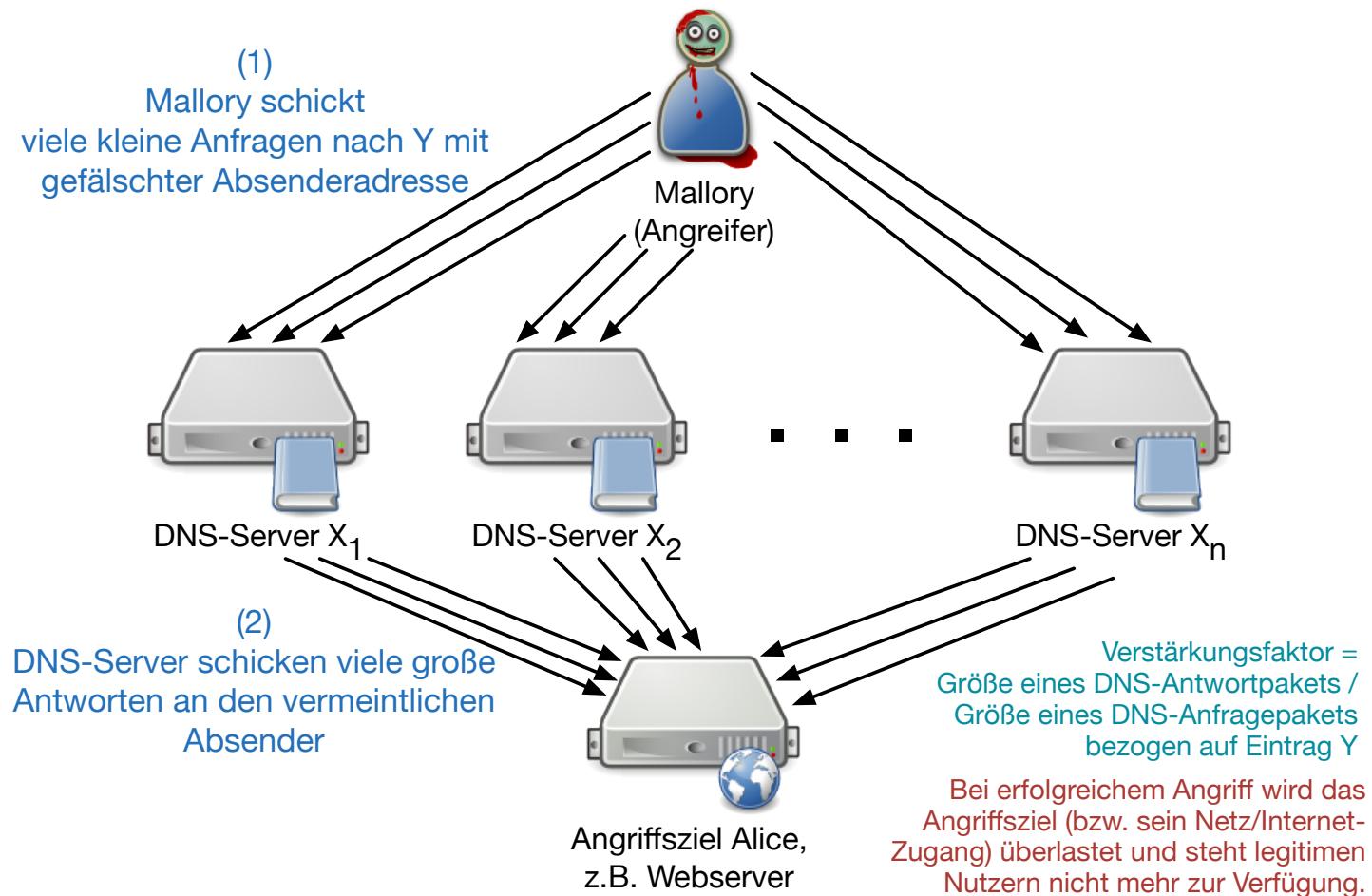
DNS Amplification Attack – Ablauf

Vorbereitung



Ergebnis: DNS-Server X_i haben Eintrag Y in ihrem Cache und liefern ihn auf Anfrage aus

DNS Amplification Attack – Ablauf Ausführung

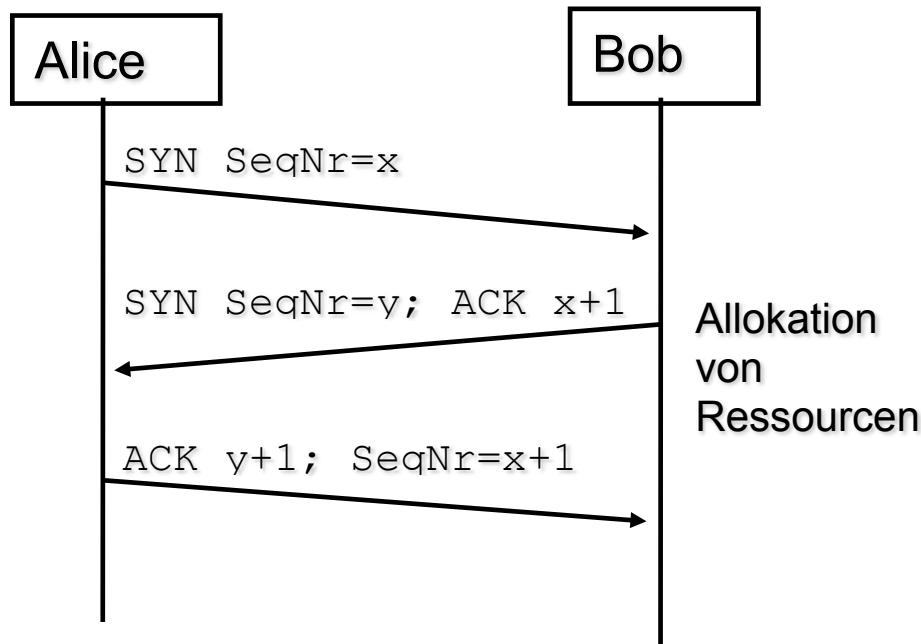


Diskussion und Gegenmaßnahmen

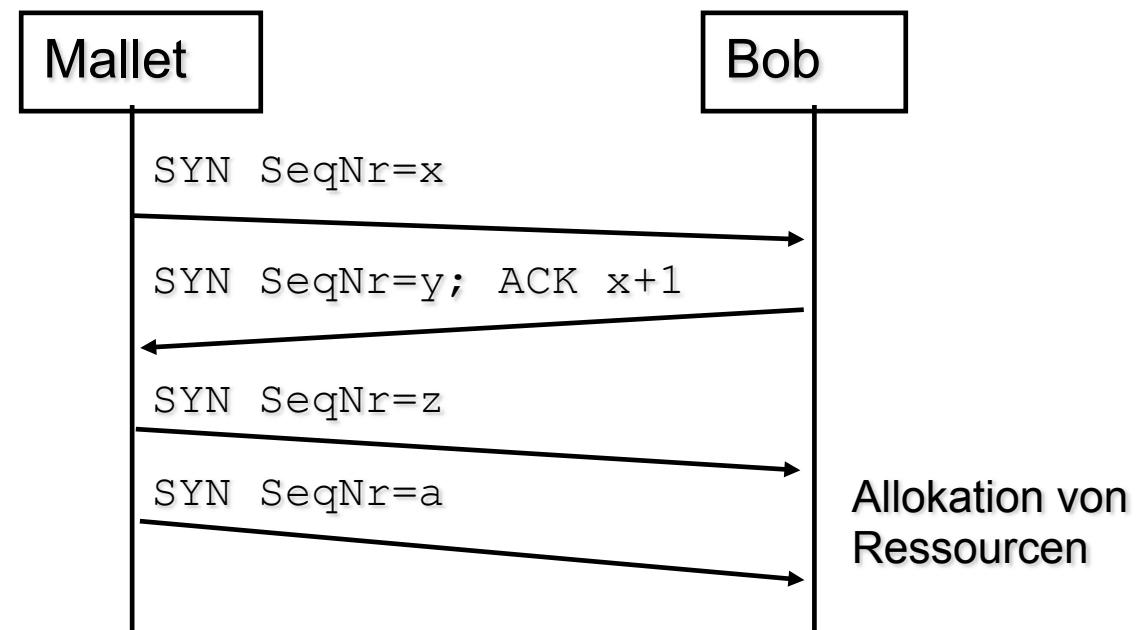
- DNS Server Xn beantworten rekursive Anfragen aus dem Internet
- Ablauf (vgl. vorherige Folien):
 - Angreifer sucht oder präpariert DNS-Server A mit langen Feldern (z.B. TXT-Feld oder DNSSEC-Key-Feld) eines Eintrages Y
 - Anfrage nach Eintrag auf Server A an Server Xi
 - Xi fragt A und schreibt Ergebnis Y in seinen Cache
 - Danach viele Anfragen nach Y an die Server Xn mit gefälschter Absenderadresse von Alice
 - Folge: Alice wird mit DNS-Antworten überflutet
- Gegenmaßnahme:
 - Keine rekursiven Anfragen von extern beantworten
 - [Schwellenwerte für identische Anfragen desselben vermeintlichen Clients]
- MWN im September 2012:
 - 58 weltweit erreichbare DNS-Server
 - 26 beantworten Anfragen rekursiv

SYN Flooding

- TCP 3-Way-Handshake zum Verbindungsauftbau



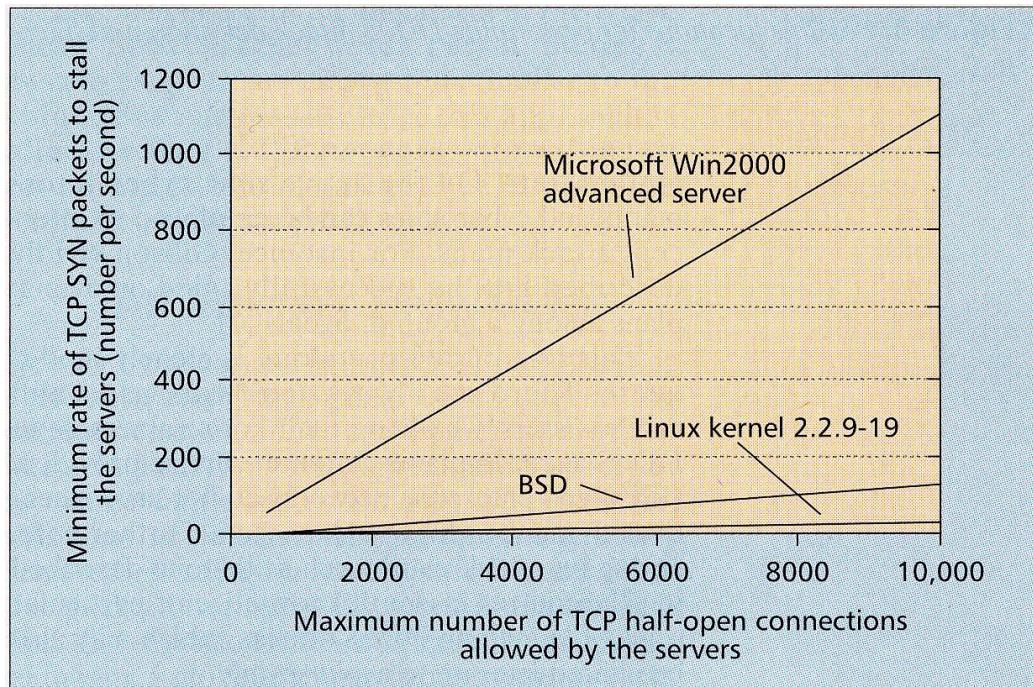
- SYN Flooding



- „Halboffene“ TCP-Verbindungen so lange aufbauen, bis Ressourcen von Bob erschöpft
- Bob kann dann keine weiteren Netzverbindungen mehr aufbauen.

Reaktion der Betriebssysteme

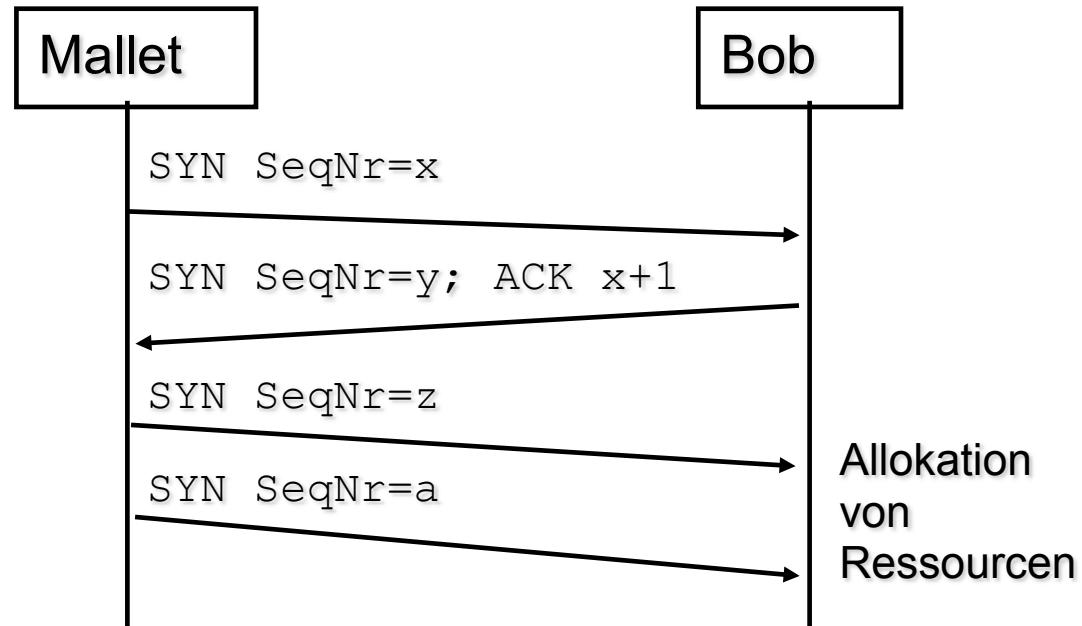
- Minimale Anzahl von SYN-Paketen für erfolgreichen DoS
Quelle: [Chang 02]



- Wiederholung von „verlorenen“ SYN-Paketen:
 - Exponential Backoff zur Berechnung der Wartezeit
 - Linux und W2K
(3s, 6s, 12s, 24s,...)
 - BSD
(6s, 24s, 48s,)
 - Abbruch des Retransmit
 - W2K
nach 2 Versuchen (d.h. nach 9 Sekunden)
 - Linux
nach 7 Versuchen (d.h. nach 381 Sekunden)
 - BSD
nach 75 Sekunden

Gegenmaßnahmen?

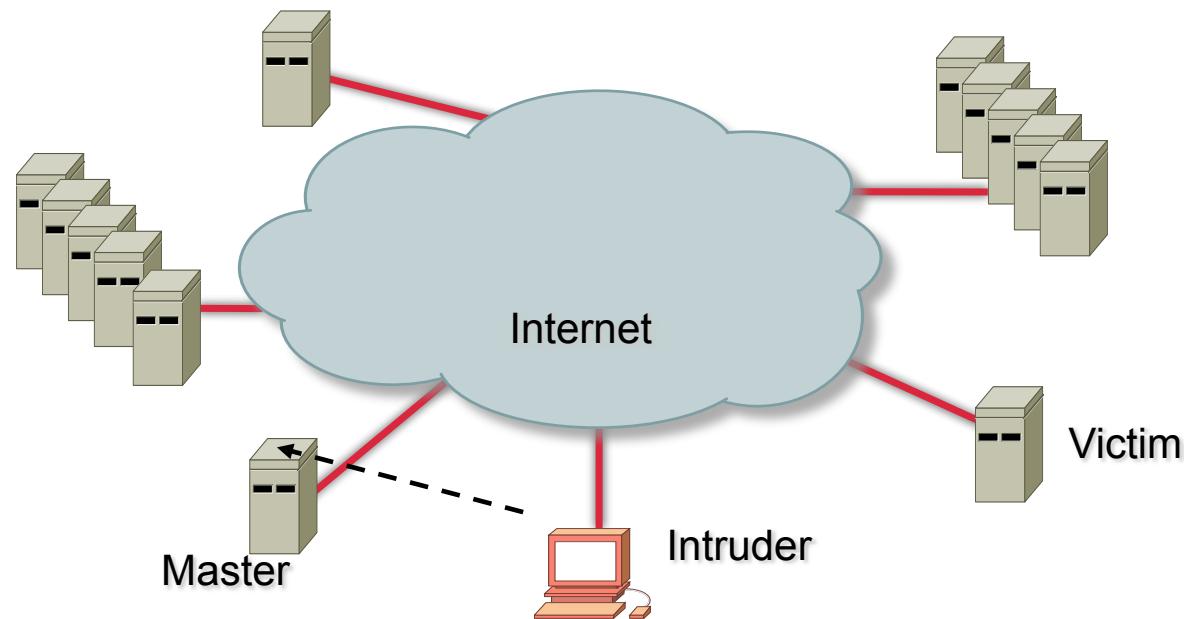
- Timer definieren:
Falls ACK nicht innerhalb dieser Zeitspanne erfolgt, Ressourcen wieder freigeben.
 - ✓ Nutzt nur bedingt
- Falls alle Ressourcen belegt: Zufällig eine halboffene Verbindung schliessen
 - ✓ Nutzt nur bedingt
- Maximale Anzahl gleichzeitig halboffener Verbindungen pro Quell-Adresse festlegen
 - ✓ Immer noch Problem bei DDoS
- SYN Cookies (Bernstein 1996):
Seq.Nr. y von Bob „kodiert“ Adressinfo von Mallet. Ressourcen werden erst reserviert, wenn tatsächliches ACK y+1 von Mallet eingeht.
 - ✓ Legitime Verbindung kommt nicht zustande, wenn das ACK-Paket von Alice verloren geht und Alice im Protokollablauf zunächst Daten von Bob erwartet.



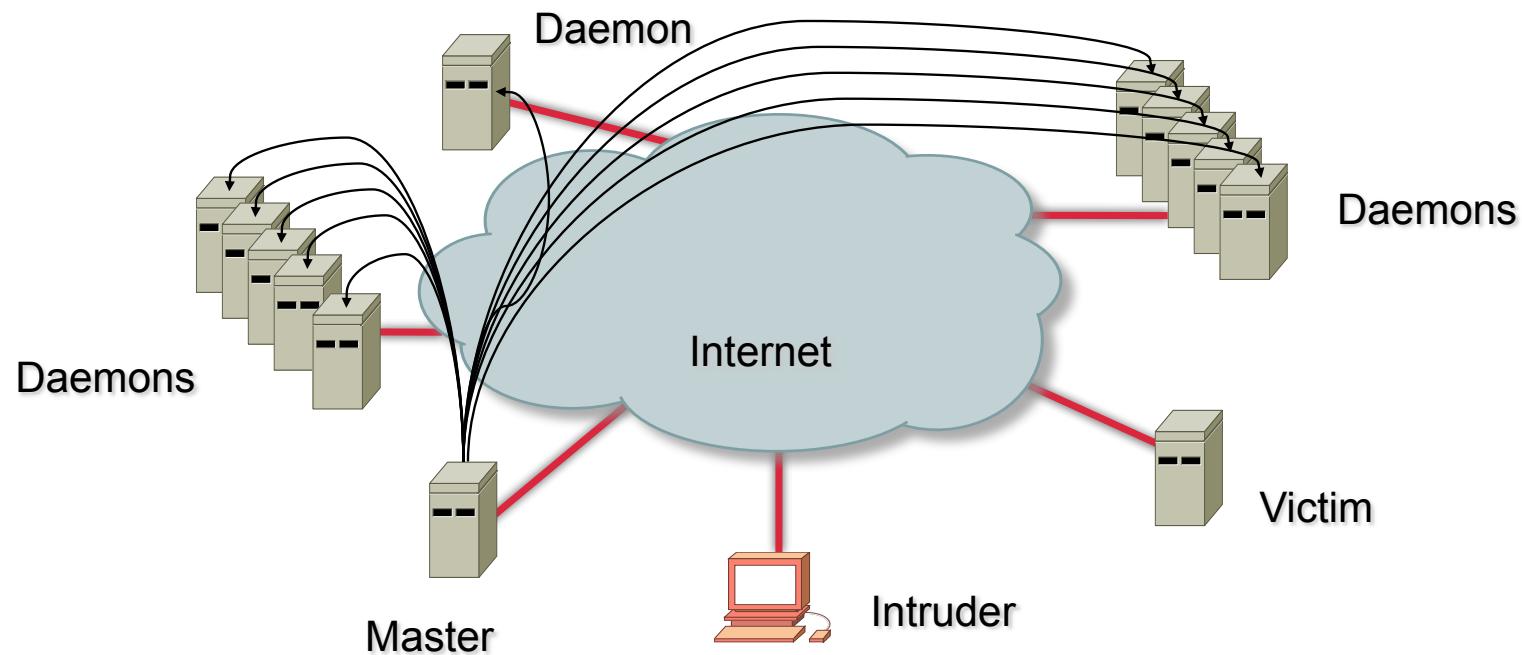
Grundsätzlicher Ablauf - Botnet

■ Dreistufiges Verfahren:

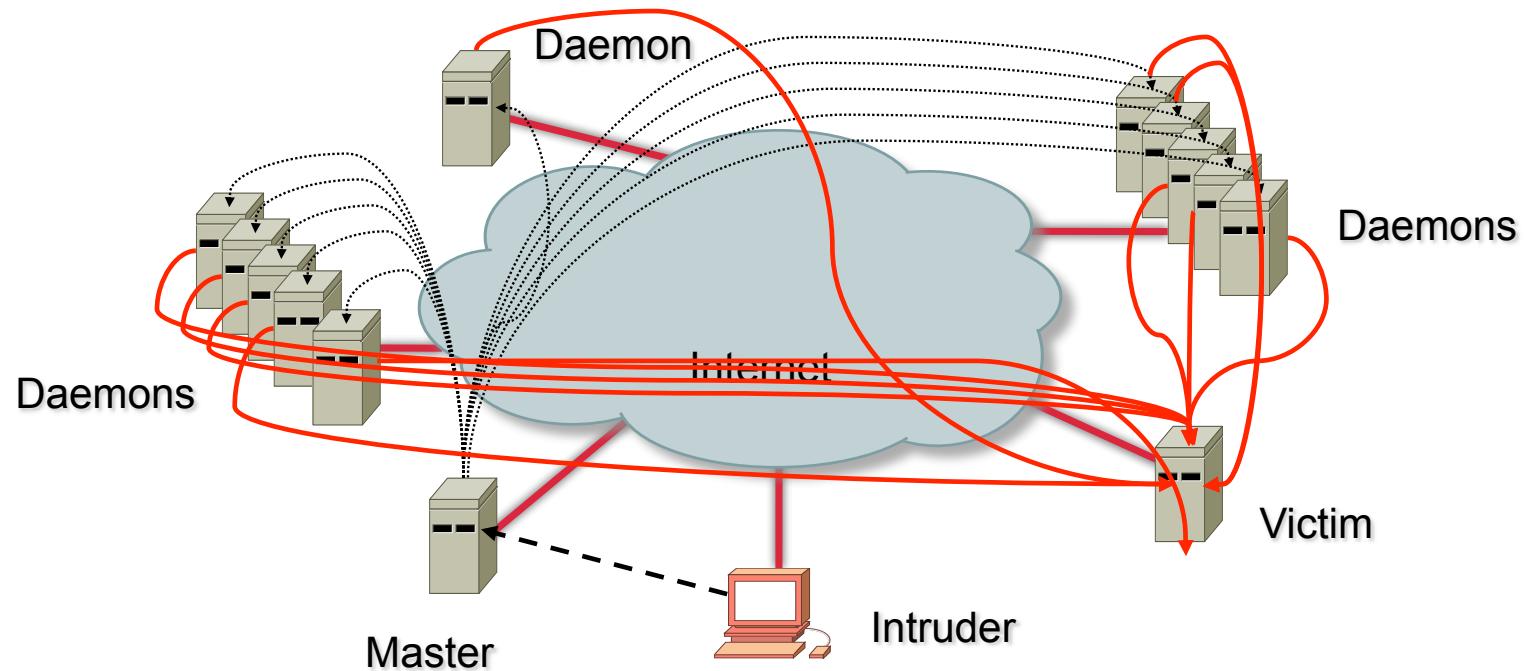
1. Intruder findet Maschine(n), die komromittiert werden können;
Hacking-Werkzeuge, Scanner, Rootkits, DoS/DDoS-Tools werden installiert;
⇒ Maschine wird Master



2. Master versucht automatisiert, weitere Maschinen zu kompromittieren, um DDoS-Software (Daemon) zu installieren, bzw. schiebt anderen Nutzern Malware unter.



3. Intruder startet Programm auf Master, das allen Daemonen mitteilt, wann und gegen wen der Angriff zu starten ist.
Zum vereinbartem Zeitpunkt startet jeder Daemon DoS-Angriff



- IoT (Internet of Things) Botnet (ab 2016)
 - Bots: DSL-Router, WebCams, Digitale Videorekorder, Fernseher, ...
 - Wenig Rechenleistung aber oft ausreichende Bandbreite
 - Kein Sicherheitsbewusstsein bei den Nutzern
- Angriffe
 - Gegen Minecraft Server
 - Webseite des Entwicklers Brian Krebs (beteiligt waren ~1 Mio Bots)
 - Internetzugang des Landes Liberia
 - DSL-Router der Telekom (Nov. 2016)
- Hilfsmittel: shodan.io Suchmaschine für IoT
- Gegenmaßnahmen:
 - Filtern des Mirai Infektionscode mit IDS
 - Patchen der Schwachstellen
 - Abschotten der Geräte, bzw. des Zugangs zum Internet

Schutz- und Gegenmaßnahmen

- Generell:
 - Pauschaler Schutz gegen (D)DoS-Angriffe ist praktisch fast unmöglich
 - Aber:
 - Spezifika einzelner Angriffe erlauben oft gute Schutzmaßnahmen
 - Ggf. temporäres Overprovisioning,
vgl. Spamhaus & DDoS protection provider Cloudflare
- Schutz gegen DoS-Angriffe auf einzelne Vulnerabilities:
 - Software-Updates und Konfigurationsanpassungen
- Schutz gegen Brute-Force-(D)DoS-Angriffe:
 - Firewall-Regeln, ggf. basierend auf Deep-Packet-Inspection
 - Aussperren von Angreifern möglichst schon beim Uplink
 - Zusammenarbeit mit den Internet-Providern der Angriffsquellen
- Allgemeine Ansätze:
 - Anzahl Verbindungen und Datenvolumen überwachen (Anomalieerkennung)
 - Bug- und Sicherheitswarnungen (z.B. CERT) verfolgen

Erpressungsversuch mit DDoS-Drohung

Betreff: DDOS www.zhs-muenchen.de

Datum: Mon, 5 Sep 2011 02:50:02 -0600

Von: <camiliaivgspopek@yahoo.com>

An: <hostmaster@lrz.de>

Your site www.zhs-muenchen.de will be subjected to DDoS attacks 100 Gbit/s.

Pay 100 btc(bitcoin) on the account 17RaBqjGLisGzLRaAUVqdA2YHgspdKD1rJ

Do not reply to this email

- Erpressungsversuche richten sich gegen zahlreiche Firmen und auch mehrere bayerische Hochschuleinrichtungen.
- Bei ausbleibender Zahlung finden tatsächlich DDoS-Angriffe statt; DDoS-Botnet besteht aus ca. 40.000 Maschinen.
- DDoS-Bots senden die folgende Anfrage:
- Filter-Kriterien:
 - Accept-Language *ru* (bei dt./eng. Website)
 - „Host“-Header nicht an erster Stelle

GET / HTTP/1.1

Accept: */*

Accept-Language: ru

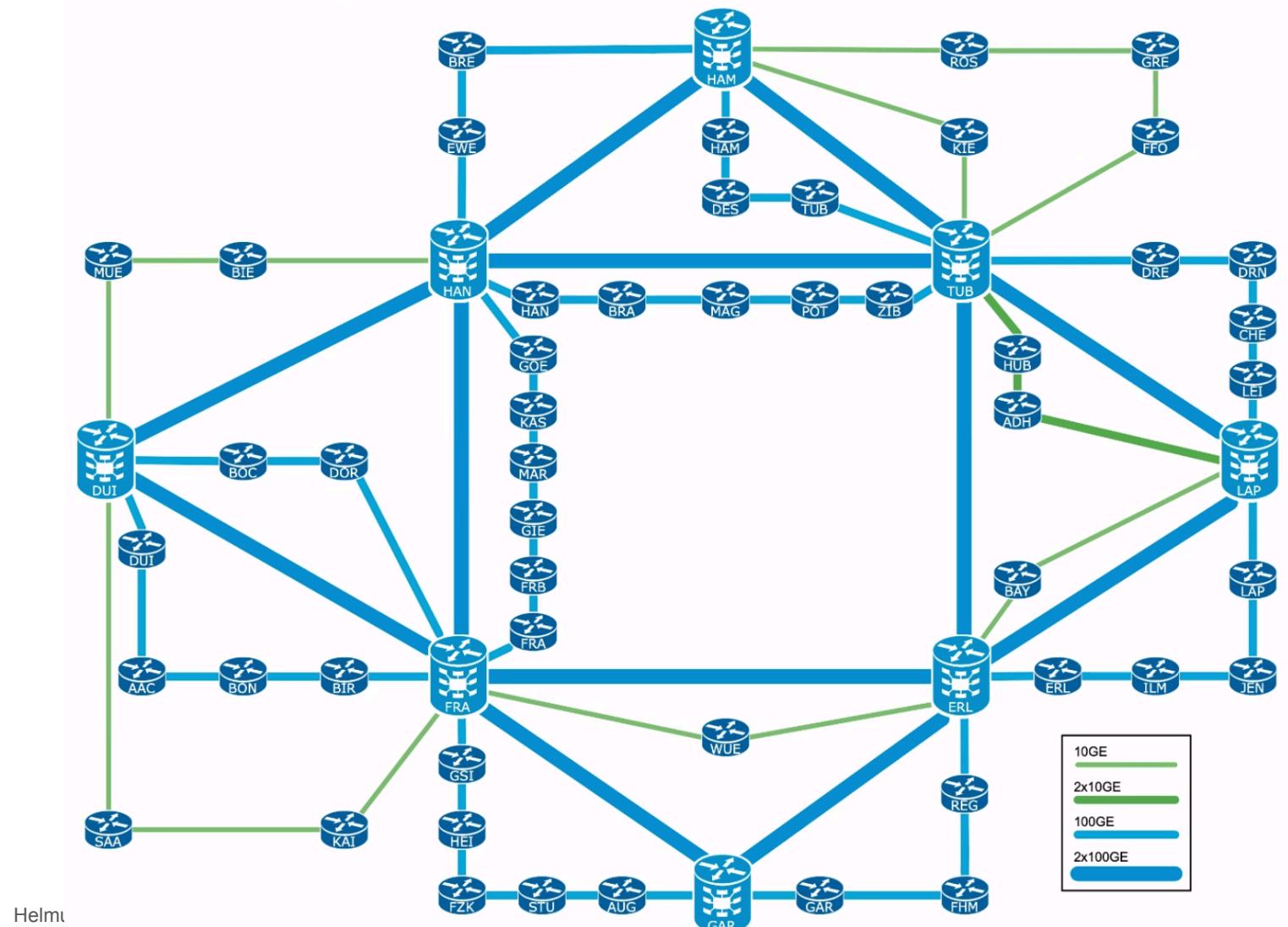
User-Agent: [useragent string]

Accept-Encoding: gzip, deflate

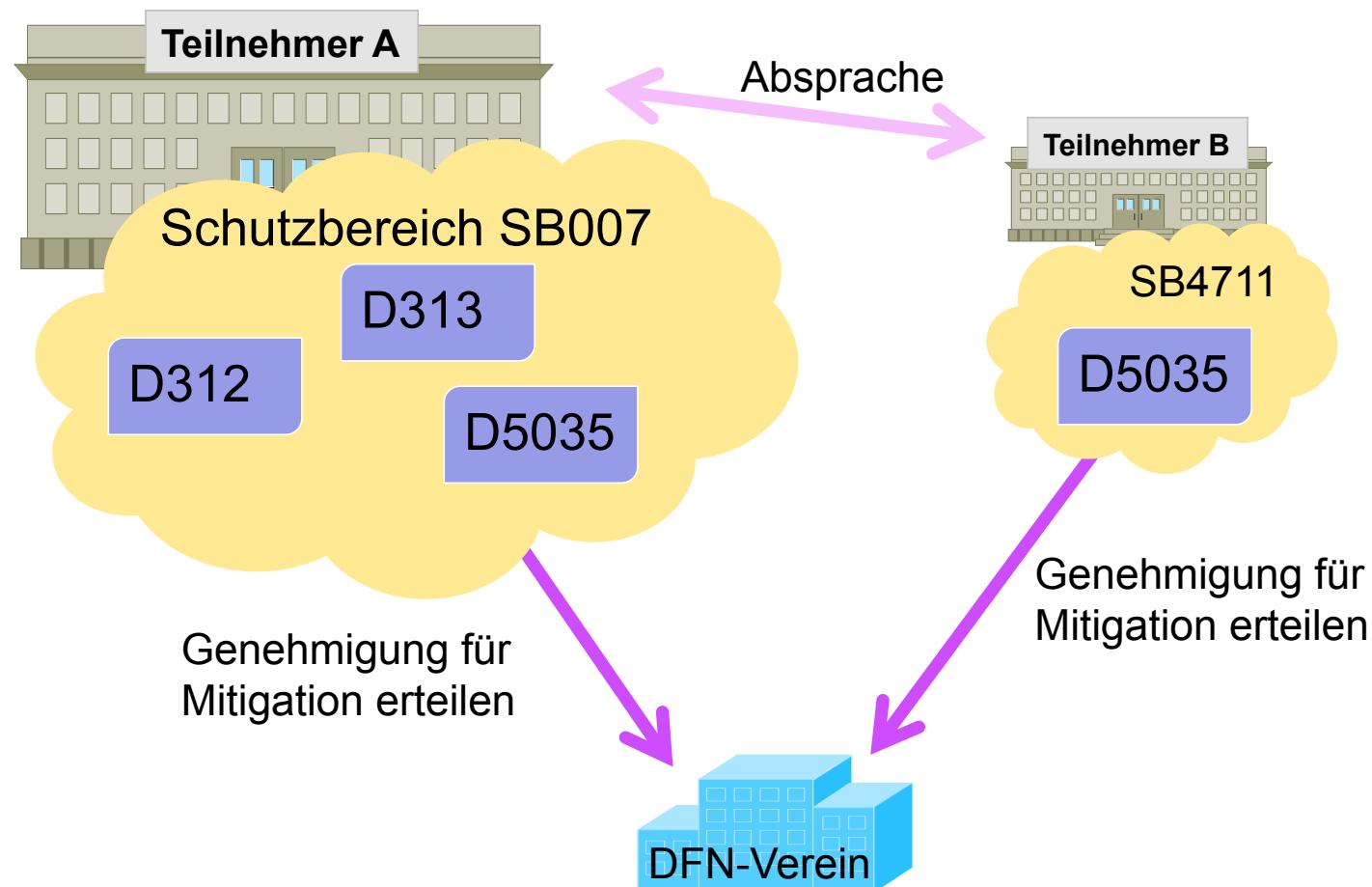
Host: [target domain]

Connection: Keep-Alive

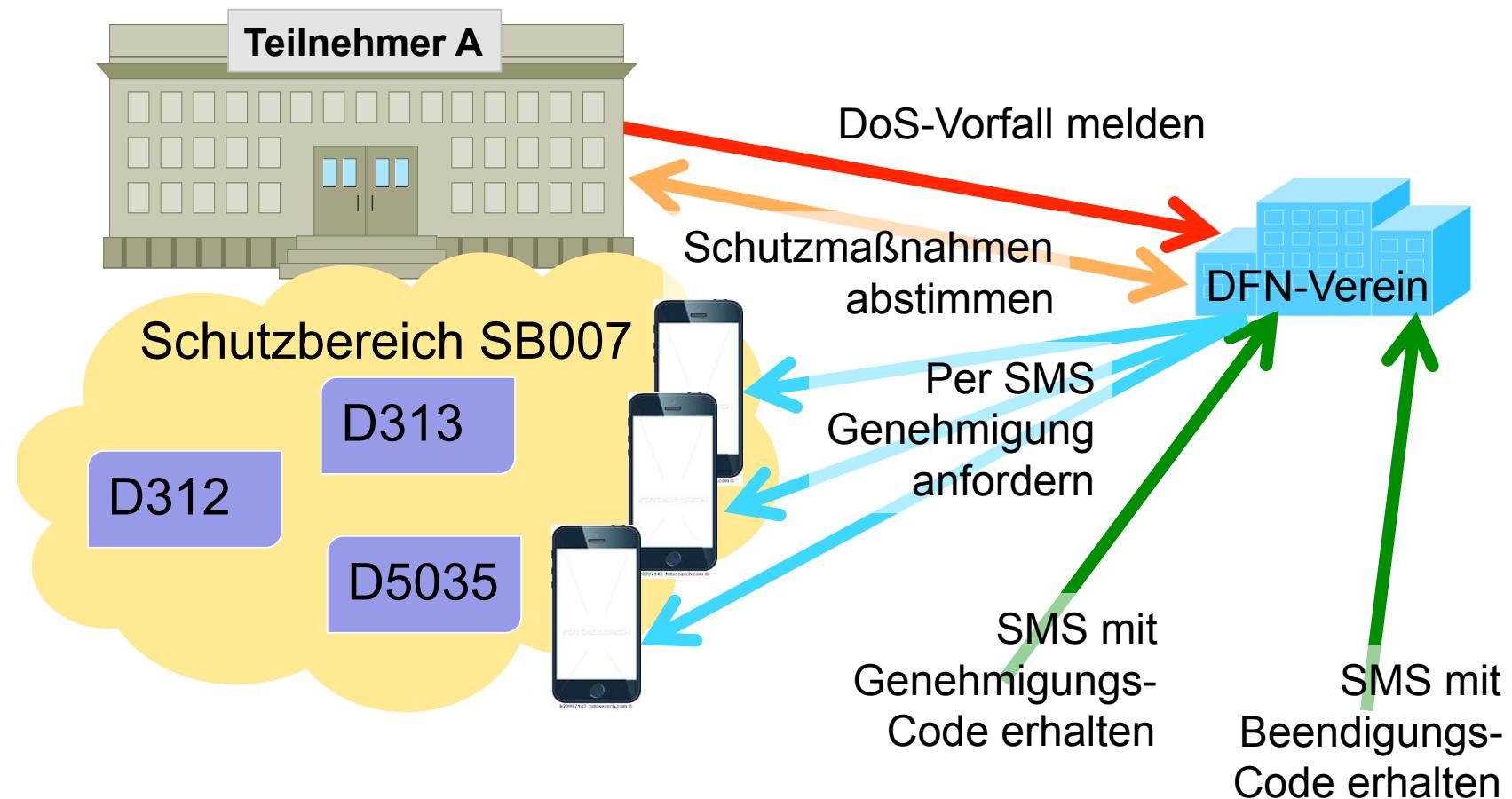
DFN: Deutsches Forschungsnetz Verein e.V.



Registrierungsprozess



Genehmigungsprozess



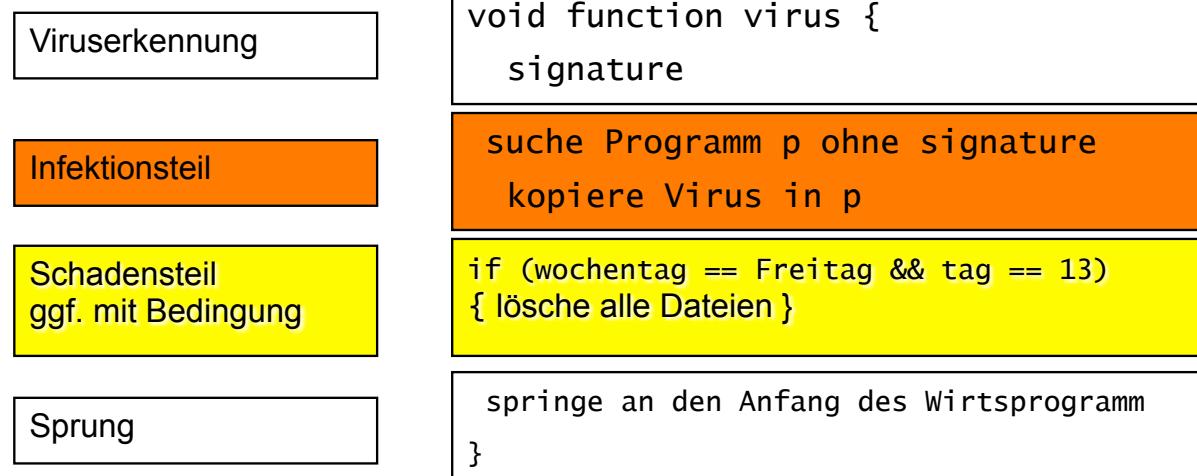
1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

Virus

■ Definition:

- Befehlsfolge; benötigt Wirtsprogramm zur Ausführung
- Kein selbstständig ablaufähiges Programm
- Selbstreplikation (Infektion weiterer Wirte (Programme))

■ Allgemeiner Aufbau:



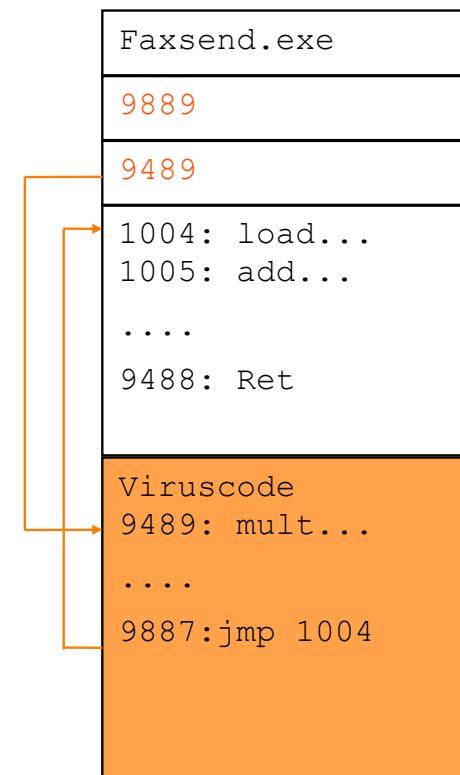
- Daneben ggf. Tarnungsteil (selbstentschlüsselnder Code, Padding, ...)

■ Dateiformat vor der Infektion (vereinfachtes Beispiel)

Name der Datei
Länge der Datei
Einsprungadresse
Programmcode

Faxsend.exe
9488
1004
1004: load... 1005: add... 9488: Ret

■ Datei nach der Infektion



Manipulierte Virensignaturen

■ Zwei Haupt-Angriffsvektoren:

- Angreifer bringen bekannte Viren-Signaturen in harmlosen Dateien unter und lassen diese über Online-VirensScanner testen
=> Im Worst Case werden z.B. die entsprechenden Files auf eine Blacklist gesetzt und von den Anwendersystemen gelöscht.
- Antivirus-Softwarehersteller erstellt Fake-Signaturen, die von der Konkurrenz ungetestet übernommen werden.

Schwere Vorwürfe gegen Firmenchef Eugene Kaspersky



heise online 15.08.2015 14:38 Uhr – Dorothee Wiegand

vorlesen

Zwei Ex-Mitarbeiter des Antiviren-Herstellers Kaspersky beschuldigen ihren ehemaligen Chef, er habe sie damit beauftragt, Konkurrenzprodukte zu sabotieren.

Zwei ehemalige Mitarbeiter des Antiviren-Herstellers Kaspersky beschuldigen den Firmenchef persönlich. In einem Bericht der amerikanischen Nachrichtenagentur Reuters werden die beiden namentlich nicht genannten Personen zitiert. Demnach habe Kaspersky einige Mitarbeiter damit beauftragt, Konkurrenzprodukte zu sabotieren. Konkret hätten sie den Auftrag bekommen, indirekt Produkte anderer AV-Hersteller so zu manipulieren, dass sie bei harmlosen Dateien Probleme melden, also Fehlalarme hervorrufen – die sogenannten False-Positive-Fälle. Aktionen dieser Art soll es über 10 Jahre gegeben haben.

[http://www.heise.de/
newsticker/meldung/
Schwere-Vorwuerfe-
gegen-Firmenchef-
Eugene-
Kaspersky-2779946.html](http://www.heise.de/newsticker/meldung/Schwere-Vorwuerfe-gegen-Firmenchef-Eugene-Kaspersky-2779946.html)

■ Definition

- Eigenständig lauffähiges Programm - benötigt keinen Wirt!
- Selbstreplikation (z.B. über Netz oder USB-Sticks (mit „Autorun“))
- Einzelne infizierte Maschinen werden als Wurm-Segmente bezeichnet

■ Beispiele:

- Internet-Wurm (1988, vgl. Kap. 1)
- ILOVEYOU (Mai 2000; ausführbares E-Mail-Attachment, verschickt sich an alle im Adressbuch eingetragenen E-Mail-Adressen)
- Code Red (Juli 2001; Defacement von Microsoft IIS Webservern)
- SQL Slammer (2003, vgl. Kap. 1)
- Conficker (November 2008; Windows-Exploits + Wörterbuch-Angriff; infizierte Maschinen formen Botnet, weltweit > 15 Mio. infizierte Rechner)
- Stuxnet (Juni 2010, vgl. Kap. 1)
- Morto (Sommer 2011; Wörterbuch-Angriff via Remote Desktop Protocol)
- NGRBot (Sept. 2012; tarnt sich per Rootkit, späht Daten aus, blockt Updates)
-

■ Definition:

- Ein Programm, dessen Ist-Funktionalität nicht mit der angegebenen Soll-Funktionalität übereinstimmt:
 - Sinnvolle oder attraktive „Nutzfunktionalität“
 - Versteckte (Schad-) Funktionalität
 - Keine selbständige Vervielfältigung

■ Beispiel: Unix Shell Script Trojan [Stoll 89]:

```
echo "WELCOME TO THE LBL UNIX-4 COMPUTER"
echo "LOGIN:"
read account_name
echo "PASSWORD:"
(stty -echo; \
 read password; \
 stty echo; echo ""; \
 echo $account_name $password >> /tmp/.pub)
echo "SORRY, TRY AGAIN."
```

„Staatstrojaner“

- Veröffentlichte Analyse (08.10.2011)
<http://www.ccc.de/system/uploads/76/original/staatstrojaner-report23.pdf>
- Chaos Computer Club (CCC) analysiert zugespielte DLL:
mfc42ul.dll
 - Wird per Registry-Eintrag geladen
 - Klinkt sich bei der Initialisierung in explorer.exe ein
- Funktionen:
 - Screenshots
 - Abhören von Skype- und VoIP-Gesprächen
 - Nachladen weiterer Module
 - Kommunikation mit Command and Control (C&C) Server



Bundestrojaner als Plastik des CCC
Photo: mellowbox/Flickr

■ Kommunikation:

- ❑ Einseitig verschlüsselt zwischen Malware und C&C-Server
- ❑ Mit AES-ECB (Electronic Code Book Mode)
 - Jeder Block wird mit dem identischen Schlüssel verschlüsselt, d.h. gleiche Klartextblöcke ergeben identische Chiffre-Blöcke
 - Schlüssel in allen Varianten identisch
- ❑ „Authentisierung“ über konstanten Banner-String „C3PO-r2d2-POE“
 - Angreifer kann sich als C&C ausgeben
- ❑ Kommando-Kanal (C&C → Malware) unverschlüsselt; keine Authentisierung
 - Malware somit durch Dritte steuerbar
 - Durch Nachladefunktion der Malware kann komplettes System durch Dritten übernommen werden
 - Zielperson kann durch gefälschte Beweise belastet werden
- ❑ Fest kodierte Adresse des C&C Servers: 207.158.22.134
 - Adresse gehört Hosting Provider Web Intellects in Ohio, USA

- Nicht alle Kommandos konnten identifiziert werden
- 18 Befehle: „--“ Kommando wird von Dispatcher nicht behandelt
 - cmd 1, cmd 10, cmd 11, cmd 15: --
 - cmd 2: Client verbindet sich neu und versucht, Daten abzusetzen (ähnlich cmd 13)
 - cmd 3: Screenshot geringer Qualität
 - cmd 4: Registrieren eines Kernelmode-Treibers
 - cmd 5: Installation aller malwarespezifischen Dateien im Dateisystem; Quelle noch nicht geklärt
 - cmd 6: Löschen der Malware aus dem Dateisystem und Reboot
 - cmd 7: Entladen der Malware
 - cmd 8: Liste aller Softwarekomponenten
 - cmd 9: wie cmd 3, nur mit drei Argumenten
 - cmd 12: Setzen irgendwelcher Werte
 - cmd 13: Screenshot von Webbrowser und Skype
 - cmd 14: Nachladen eines Programms und unmittelbare Ausführung

- Bundestag beschließt Gesetz zur Anpassung des Verfassungsschutzrechtes (10.06.21)
 - Quellen-TKÜ (auch von Messenger Diensten) wird erlaubt
 - Nachrichten werden vor Ver- bzw. nach Entschlüsselung auf dem Endgerät ermittelt
 - -> Dazu Software auf dem Endgerät des Überwachten erforderlich
 - Provider werden verpflichtet Verkehr auf Anforderung umzuleiten
- Juli 2021: Pegasus Projekt veröffentlicht
 - Hunderte Journalisten, Menschenrechtler und Politiker werden weltweit mit Spähsoftware Pegasus (Handy-Spähsoftware, Fa. NSO, Israel) überwacht
 - Sept. 21: Bundeskriminalamt soll Pegasus gekauft haben
 - Keinerlei Auskunft wegen staatswohlbegründeten Geheimhaltungsinteressen

Schutz- und Gegenmaßnahmen

- Auf allen Systemen (Desktop + Server):
 - Anti-Viren-Software installieren und aktuell halten
 - Keine Software zweifelhafter Herkunft installieren
 - Getrennt gelagerte, regelmäßig erstellte Daten-Backups
- Auf Desktop-Systemen:
 - Funktionen wie automatische Makro-Ausführung, Autorun etc. deaktivieren
 - Ggf. virtuelle Maschinen zum „Surfen“ und Ausprobieren von Software verwenden (Isolation, Sandboxing)
- (Primär) auf Server-Systemen:
 - Integrity-Checker einsetzen (→ Host Intrusion Detection Systeme)
 - Schreibrechte sehr restriktiv vergeben (Need-to-know-Prinzip)

- Diverse “Apps” für Smartphones und Desktops
 - Vordergründig oft kostenlose, interessante Anwendung
 - Im Hintergrund:
 - Übermitteln des gesamten Adressbuchs an Hersteller
 - Übermitteln der eindeutigen Gerätekennung an Werbenetzwerke
 - Umleiten des Internet-Traffic über Server des Herstellers
 - Mining von Bitcoins o.ähnl.
 - Versand von Premium-SMS o.ähnl.
 - Ohne Analyseumgebung (z.B. Simulator, Netzmonitoring) für Anwender nicht erkennbar
- Hardware-basierte/-nahe Trojanische Pferde
 - Manipulierte Hardware / Firmware, z.B. NSA Supply-Chain Interdiction
 - BadUSB: Z.B. Manipulierte USB Memory-Sticks mit Tastaturemulation zum Absetzen von beliebigen Befehlen

NSA Supply-Chain Interdiction

Die NSA fängt Postsendungen ab

Bild 1 von 3

(TS//SI//NF) Such operations involving **supply-chain interdiction** are some of the most productive operations in TAO, because they pre-position access points into hard target networks around the world.



(TS//SI//NF) Left: Intercepted packages are opened carefully; Right: A “load station” implants a beacon

Blick hinter die Kulissen

So werden Pakete offenbar geöffnet (links) und die enthaltene Technik manipuliert (rechts).

Bild: Glenn Greenwald, "Die totale Überwachung"

Quelle: [http://www.heise.de/
newsticker/meldung/NSA-
manipuliert-per-Post-versandte-US-
Netzwerktechnik-2187858.html](http://www.heise.de/newsticker/meldung/NSA-manipuliert-per-Post-versandte-US-Netzwerktechnik-2187858.html)

Atlassian Confluence Schwachstelle

- Software Hersteller Atlassian veröffentlicht am Abend des 30.10. ein Advisory (d.h. in Europa am 31.10. morgens)
- Improper Authorization-Schwachstelle (CVE-2023-22518), CVSS 9,1 für Confluence Data Center
- Ausnutzung durch entfernten Angreifer, Gefahr von
 - signifikantem Datenverlust
 - Datenveränderung
 - KEIN Auslesen von Information
- Empfehlung: Patch vom 31.10. einspielen

BayernCollab

- LRZ betreibt bayernweite Kollaborationsplattform BayernCollab auf Basis von Confluence Data Center
- Nutzbar für alle bayerischen Universitäten und Hochschulen
- Migrationsphase läuft
 - TUM und LMU sind bereits migriert
 - 4.217 Spaces
 - 98.351 importierte Benutzer
 - 23.489 Gruppen
- Risikoabschätzung?

Vorfallsbearbeitung

- LRZ wird am 31.10. um 5:02 informiert
- Kollegen eröffnen Security Incident (SI-236)
 - ISMS Prozess zur Security Incident Bearbeitung läuft an, bestimmt werden
 - SI-Coordinator,
 - SI-Hotliner
 - Admin
- Fix wird umgehend eingespielt
- SI wird innerhalb von 2 Stunden 36 Minuten abgeschlossen
- Well done!

Ransomware

- Krypto-Erpressungstrojaner
- Malware verschlüsselt Dateisystem und verlangt „Lösegeld“
- WannaCry (Mai 2017)
 - Ausbreitung startet in Russland
 - Krankenhäuser in ganz England betroffen,
 - z.T. wird Betrieb eingestellt, Patienten sollen nicht mehr in Notaufnahme kommen und werden z.T. nach Hause geschickt
 - Nissan Fabrik in Sunderland betroffen
 - Renault stoppt den Betrieb in einigen Fabriken in Frankreich
 - Zuginformationssysteme der Deutschen Bahn
 - Ursache: Schwachstelle in Windows, Veraltete Windows Versionen (NT4, XP, 2000) in Betrieb
 - Gegenmaßnahmen
 - Patch seit März verfügbar
 - Firewall: Port 445/139 und 3389 schließen

- Justus-Liebig-Universität (JLU) Giessen seit So. 8.12.19 offline
 - „Justus Liebig Universität Gießen hat nach einem schwerwiegenden IT-Sicherheitsvorfall ihre Server [...] heruntergefahren“ Twitter: #JLUOffline
 - Mo. 9.12. Ermittler des LKA sowie Fachleute des Darmstädter Forschungszentrum für Cyber-Sicherheit ATHENE treffen ein
 - 11.12. Gießener Anzeiger: „Uni Gießen noch Wochen offline“

- Justus-Liebig-Universität
 - 30.000 Studierende, 5.600 Mitarbeiter
 - 11 Fachbereiche
 - 150 (z.T. internationale) Studiengänge

- Ab Fr. 13.12. Verteilung von USB-Sticks zum Virenskan aller Rechner
 - Verteilung über Fachbereiche, Institute und Professuren
 - Scan lokal und ohne Netzzugang zwingend
 - Geräte die unauffällig sind erhalten grünen Aufkleber, alle anderen einen roten
 - Wegen Komplexität der Schadsoftware ist zweite Scan-Welle erforderlich (in der darauffolgenden Woche)
 - Nur Geräte mit zwei grünen Aufklebern werden zur Benutzung freigegeben

Was bedeutet das?

- Bewerbung zum Sommersemester möglich?
- Kein Internet in Wohnheimen! 
- Fristen und Dokumente für Studierende:
 - Zeugnisse, Urkunden, Scheine, Noten- und Prüfungseinsicht
 - Zugangsvoraussetzungen für Prüfungen o.ä.
 - Erasmus-Bescheinigungen
 - Immatrikulationsbescheinigungen (z.B. für Visa-Verlängerungen)
- Finden Vorlesungen statt? Wie kommen Studierende an digitale Lerninhalte?
- Spitzenforschung? Sind Ergebnisse oder Deadlines in Gefahr?
- Werden Gehälter bezahlt?
- Wie können Rechnungen bezahlt werden?



- Infektion mit Verschlüsselungstrojaner: Emotet/Trickbot
 - 2014 entwickelt als Online-Banking Trojaner, danach mehrere Evolutionsstufen
 - Adaptiert für massenhaften und automatisierten Einsatz
- Eigenschaften
 - Kann auf infizierten Systemen E-Mails und Adressbücher auslesen und daraus Spam-Mails generieren; Absender ist eine bekannte Adresse
 - Text bezieht sich auf eine frühere Mail des Empfängers
 - Signatur ist echt/authentisch
 - Enthält oft Word oder Excel-Dateien oder Link auf Office365 Dokumente
 - Versteckt sich vor Anti-Viren Software, deshalb kaum zu entfernen
- Modular aufgebaut: lädt Schad-Code nach, um „in die Breite“ zu infizieren einmal geklickt - ganzes Subnetz infiziert 😞

Schadensabschätzung



- 38.000 Accounts neu setzen - persönliches Erscheinen
- 3 Wochen keine IT-basierte Tätigkeit
- weitere 3 Wochen kein direkter Zugriff auf Daten
- Vollständige Wiederherstellung der Daten aus Backups dauert 2-3 Monate
- Dienste werden nach Wichtigkeit wieder hergestellt, nach 1,5 Jahren nicht alle Dienst online
- Direkte Kosten: 1,7 Mio. € (RZ: 1,1 Mio.; andere Einrichtungen: 600 k€)

Schadensabschätzung: indirekte Kosten

Kennzahlen zur Kostenschätzung	
Betroffene Mitarbeiter	5.000
Durchschnittliche Personalkosten pro Jahr	50.000 €
Kosten pro Arbeitstag (bei 250 Jahresarbeitstagen)	200 €

Sachverhalt	Zeitraum in Arbeitstagen	Anteil des Arbeitsausfalls	Gesamtschaden
Kein reguläres Arbeiten möglich (drei Wochen vor Weihnachten)	15	100%	15.000.000 €
stark eingeschränktes Arbeiten, kein Zugriff auf Daten (drei Wochen nach den Weihnachtsferien)	15	50%	7.500.000 €
Stark eingeschränkter Datenzugriff (2-3 Monate)	52	20%	10.400.000 €
Gesamtkosten			32.900.000 €

Frankfurter Allgemeine

COMPUTERVIRUS

Hacker-Angriff schränkt Betrieb im Klinikum Fürth ein

AKTUALISIERT AM 13.12.2019 - 14:29

HACKER-ANGRIFF
Ruhr-Uni: Hacker wollten von Hochschule Lösegeld erpressen **NRZ +**
Christopher Onkelbach 29.05.2020 - 14:59 Uhr

Hacker fordern Lösegeld



Cyber-Attacke lähmt Krauss Maffei: Kommen die Hacker aus Nordkorea oder Russland?

Aktualisiert: 04.01.19 - 09:31

Sicherheitsvorfälle - Übersicht

- Weltweite Cyberangriffe:
 - <https://konbriefing.com/de-topics/cyberangriffe.html>
- Angriffe gegen Universitäten (Ransomware, DDoS, Datendiebstahl)
 - <https://konbriefing.com/de-topics/cyber-angriffe-universitaeten.html>



30. Oktober 2023

Cyberangriff auf eine Fachhochschule in Niedersachsen, Deutschland

Hochschule Hannover (HsH) - Hannover, Niedersachsen, Deutschland

Die Hochschule Hannover ist von einem Cyberangriff betroffen

<https://www.hs-hannover.de/ueber-uns/org...>



Oktober 2023 ?

Cyberangriff auf die Polizei einer Universität in Kalifornien

Stanford University Department of Public Safety - Stanford, Kalifornien, USA (Santa Clara County)

[Stanford statement on Department of Public Safety cybersecurity incident](#)

<https://news.stanford.edu/report/2023/10...>

[Stanford University investigating cyberattack after ransomware claims](#)

<https://therecord.media/stanford-investi...>



Oktober 2023 ?

Cyberangriff auf einen Universitätsinstitut in Bremen

Universität Bremen, Institut für Didaktik der Naturwissenschaften - Bremen, Deutschland

[Hackerangriff auf unseren Server und Zugang zu Unterrichtsmaterial](#)

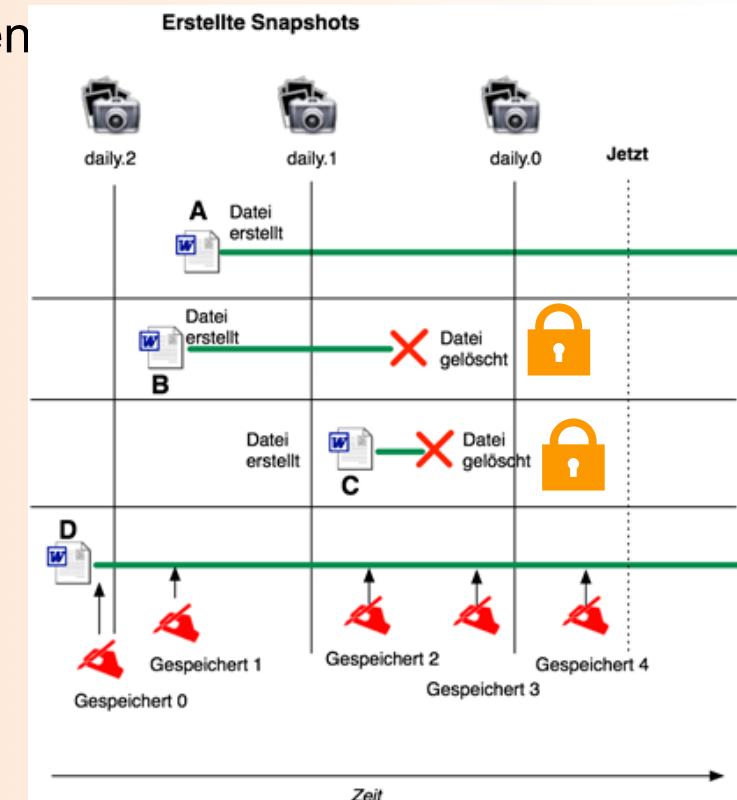
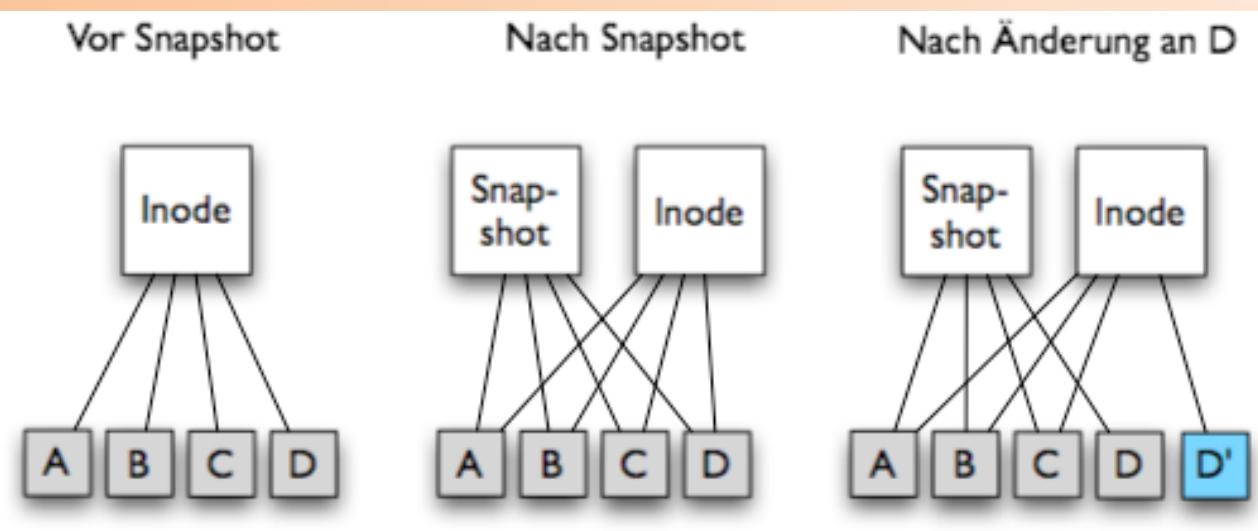
<https://chemiedidaktik.uni-bremen.de/hac...>

- Updates und Patches installieren
- Backups anlegen
 - andere Medien (Bänder)
 - Dateisysteme, Netzlaufwerke nicht dauernd angebunden lassen
- Schutzsoftware (VirensScanner) installieren

- „*Nur E-Mails und Anhänge von bekannten Absendern öffnen*“
 - Absender können sehr einfach gefälscht werden
 - Rechner des Absenders kann kompromittiert sein
 - Ggf. über anderen Kanal beim Absender nachfragen

Snapshots: Admin's Best Friends

- Snapshots sind eine Art von Sicherungskopie zu definierten
- Schedule: stündlich, täglich, wöchentlich
- Snapshots sind Read-Only!!!
 - Können nicht verschlüsselt werden
- Keine Performance-Einbußen (abhängig vom Filter)
- Zusätzlicher Speicherplatz für Snapshots nötig!



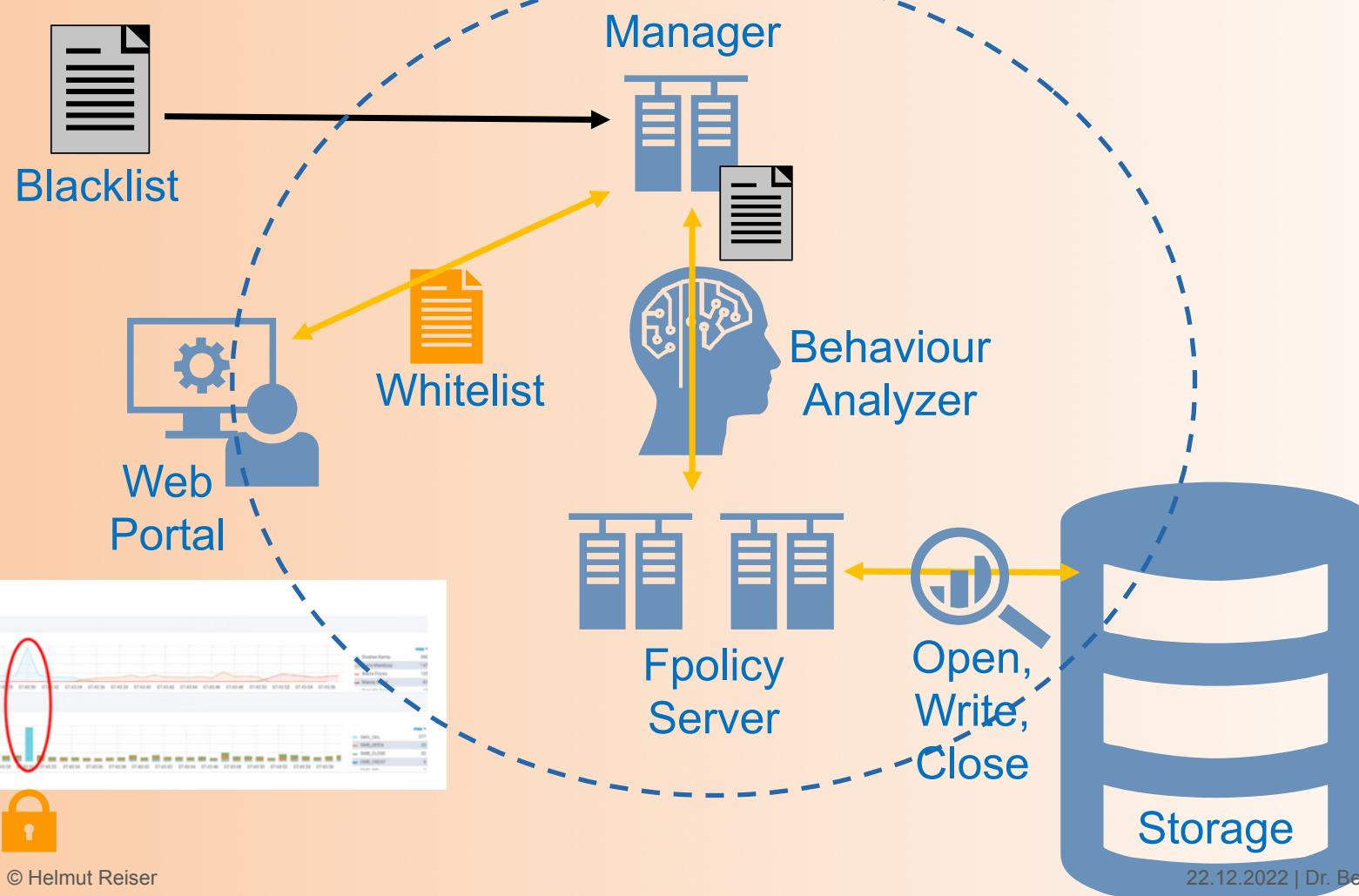
Snapshots sind gut, aber

- Wie erkenne ich einen Verschlüsselungsvorgang?
 - Je Früher desto besser
- Wie blockiere ich die weitere Verschlüsselung?
 - Welcher User verschlüsselt gerade?
 - Welcher Rechner ist involviert?
 - Welche Daten wurden/werden verschlüsselt?
 - Wie blockiere ich auch in der Nacht und am Wochenende?
- Habe ich immer einen aktuellen Snapshot parat?
 - Snapshot beim ersten Anzeichen einer Verschlüsselung erstellen
- Zielgerichteter Restore der verschlüsselten Dateien
 - Möglichst nicht komplette Fakultät/Lehrstuhl auf alten Stand zurücksetzen

- Erkennungen durch Whitelist, bzw. Blacklist
 - Nicht mehr zielführend, da Ransomware-Angriffe zunehmend „intelligenter“ werden. Schadcodes ändern nicht mehr die Dateiendungen in Werte wie .crypto oder .locky.
 - Funktioniert nicht im universitären Umfeld. Sehr viele Dateiendungen vorhanden, die in Blacklist stehen.
 - Automatischer Update der Listen führen zu vielen „false positive“ Meldungen.
- Erkennung durch Behaviour Analyzer
 - Überwacht Verhaltensmuster der Benutzer in Bezug auf alle Zugriffe.
 - Jede Transaktion wird in Echtzeit analysiert, ohne merkbare Performance-Einbußen.
 - Bei einer Anomalie wird Alarm ausgelöst und blockiert den Angreifer, um eine weitere Ausbreitung zu verhindern.
 - Der geblockte User hat keinen Zugriff mehr.
 - Alle anderen User arbeiten ohne jegliche Unterbrechung weiter.
 - Es ist Fine-Tuning des Algorithmus notwendig um nicht zu viele „false positive“ Meldungen zu haben.

- CryptoSpike liefert dem Administrator alle relevanten Informationen
 - User und Rechner
 - Pfad und Anzahl der betroffenen Dateien
 - Administrator kann Transaktionen analysieren
 - Die SW unterstützt mittels Integration in Snapshots den single file restore
 - Nur die betroffenen/manipulierten Dateien werden wiederhergestellt

Ransomware Protection - CryptoSpike



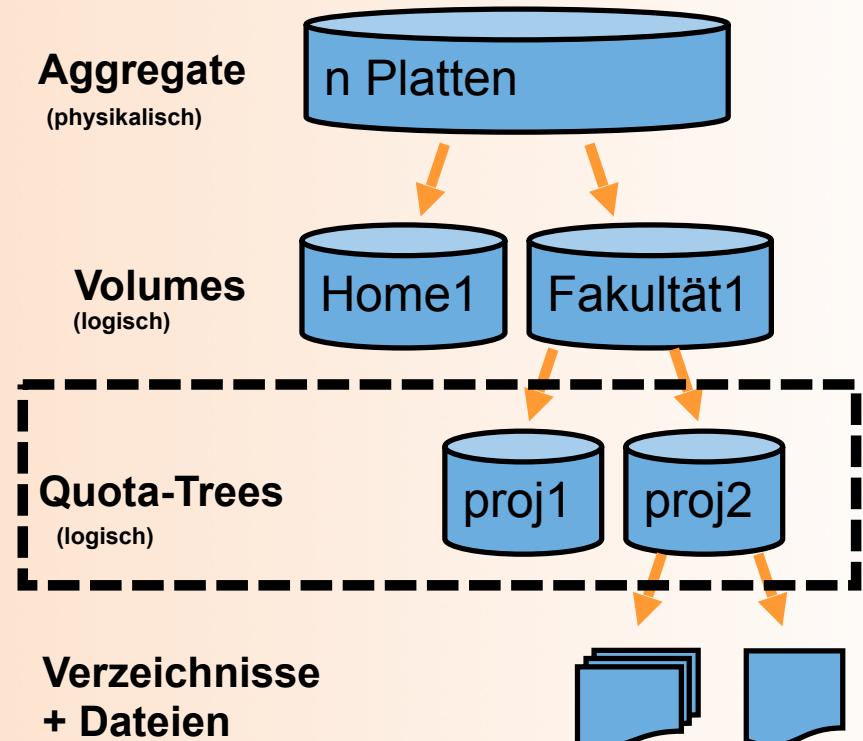
Ransomware Schutzziele für Speichersysteme

- Möglichst einen Befall vermeiden ☺
 - Alleine 2021 fanden ca. 623 Millionen Angriffe durch Ransomware statt
 - (Fast) nicht möglich → Eher eine Frage von WANN, WIE OFT bzw. WIE INTENSIV trifft es mich.
- Befall möglichst einschränken und Ausbreitung verhindern
- Befallene Daten wiederherstellen können
 - Snapshots erstellen
 - Spiegelung auf Sekundärsystem inkl. Backup auf Tape (Medienbruch)



Schutzziel - Ausbreitung einschränken

- Speicherbereiche begrenzen
 - Je kleiner, desto weniger wird verschlüsselt
 - Backup/Restore hängt von Größe ab
- Organisationsstruktur setzen Grenzen
 - Personal Storage
 - Institutional Storage
 - Fakultäten
 - Lehrstühle
 - weitere Projekte
- Zugriffsrechte
 - Je enger, desto besser!
- Snapshots (Read-Only!!!)
 - Können nicht verschlüsselt werden
- Software zum Schutz gegen Ransomware
 - Erkennt und unterbindet Verschlüsselung



1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

Spam-E-Mail

- Unerwünschte Werbemails (unsolicited commercial e-mail, UCE)
- Begriff SPAM
 - SPAM eingetragenes Warenzeichen von Hormel Food
 - „Spam“-Sketch aus Monty Python's Flying Circus
- E-Mail-Spam-Aufkommen
 - Am Beispiel LRZ, ein Tag im Oktober 2008
 - Zustellversuche für 14.556.000 Mails
 - Spam und Viren-Mails: 14.436.000 (~99,18 %)
 - Abgelehnte Mails: 14.400.000 (~99 %)
 - Als Spam markiert: 35.000 (~0,24 %)
 - Viren-Mails: 1.000 (~0,01 %)
 - Gewünschte Mails („Ham“): 120.000 (~0,82 %)
- Probleme:
 - Eingangs-Mailbox wird mit Spam überflutet
 - Extrem störend, oft „gefährlicher“ Inhalt
 - Zusätzlicher Aufwand (Speicherplatz, Arbeitszeit)
 - Zusätzliche Kosten (Infrastruktur, Übertragung, Personal,...)



Beispiel

Zielgruppenorientierter Spam

Subject: UNIVERSITY DIPLOMAS

Date: Tue, 08 Aug 1996 18:47:06 -0400 (EDT)

Obtain a prosperous future and secure the admiration of all for as little as \$125.

Diplomas from prestigious non-accredited universities based on your life experience.

No tests, no classes, no interviews.
All diplomas available including bachelors, masters, and doctorates (PhD's).

No one is turned down.

Your diploma puts a University Job Placement Counselor at your disposal.

Confidentiality assured.

CALL NOW to receive your diploma within days!!!

1-603-623-0033, Extension 307

Open Every Day Including Sundays and Holidays

Willkommen! Wir haben festgestellt, dass Ihr Werdegang und Ihr Lebenslauf von mehreren Online-Personalvermittlungsagenturen empfohlen wurden. Deshalb möchten wir Ihnen einen Teilzeitjob anbieten, den Sie in Ihrer Freizeit ausüben können. Unsere Aufgabe ist einfach: Wir bewerten einfach Ihre Lieblingshotels. Es gibt keine zeitliche Begrenzung und Sie können die Beurteilung zu Hause durchführen. Der Tageslohn liegt zwischen 300 und 1000 Euro, alle Löhne werden am selben Tag ausgezahlt. Wenn Sie teilnehmen möchten, kontaktieren Sie uns bitte per WhatsApp:

Phishing

Information Regarding Your account:

Dear PayPal Member!

Attention! Your PayPal account has been violated!

Someone with ip address 86.34.211.83 tried to access your personal account!

Please click the link below and enter your account information to confirm that you are not currently away. You have 3 days to confirm account information or your account will be locked.

[**Click here to activate your account**](#)

You can also confirm your email address by logging into your PayPal account at
<http://www.paypal.com/> Click on the "Confirm email" link in the Activate Account box and then enter this confirmation number:
1099-81971-4441-9833-3990

Thank you for using PayPal!
The PayPal Team

Please do not reply to this e-mail. Mail sent to this address cannot be answered. For assistance,



PayPal Email ID PP391

Protect Your Account Info

Make sure you never provide your password to fraudulent websites.

To safely and securely access the PayPal website or your account, open a new web browser (e.g. Internet Explorer or Netscape) and type in the PayPal login page (<http://paypal.com/>) to be sure you are on the real PayPal site.

PayPal will never ask you to enter your password in an email.

For more information on protecting yourself from fraud, please review our Security Tips at
<https://www.paypal.com/us/securitytips>

Protect Your Password

You should never give your PayPal password to anyone.

Arten von Phishing

- Clone phishing (“Update” echter E-Mails)
- Spear phishing (personalisiertes, zielgerichtetes Phishing)
- Whaling (Phishing z.B. gegen hochrangigen Mitarbeiter)
- CEO Fraud (Manipulation zur Überweisung von Geld)
- IBAN Fraud (manipulierte PDF-Rechnung mit geänderter Kontoverbindung)
- Vishing (Voice Phishing; Ziel: Opfer ruft Angreifer an)

LRZ SI-192: Phishing with IBAN/BIC modification



From: Leibniz-Rechenzentrum <invoice@accounts-financial.com>

Sent: Monday, 21 November 2022 13:27

To: info [REDACTED]

Subject: Attn.: Financial department - [REDACTED]

NOTE: This e-mail originates from outside [REDACTED] exercise caution with any instructions in the content, attachments or links.

Good Afternoon ,

We have checked your account and saw there are no outstanding invoice statements;
Do we have any outstanding invoice?

Best regards,

Veronika

Leibniz-Rechenzentrum



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Veronika Lauss | Finance & Administration

Leibniz-Rechenzentrum

Boltzmannstraße 1

Garching 85748

Germany

SI-192: Anzeige bei der Polizei

Staatsanwaltschaft München I



lrz

Staatsanwaltschaft München I,
80097 München

Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften
Boltzmannstraße 1
85748 Garching b. München

Frau Staatsanwältin D'hein
Telefon: 089/5597-4709
Telefax: 089-55974131

Ihr Zeichen, Ihre Nachricht vom

Bitte bei Antwort angeben
Akten - / Geschäftszeichen
251 UJs 742382/22

bri
Datum
23.12.2022

Ermittlungsverfahren gegen Unbekannt
wegen versuchten Betruges

das Ermittlungsverfahren wurde eingestellt, weil der Täter bisher nicht ermittelt werden konnte.

Sollte der Täter im Verlauf weiterer Ermittlungen bekannt werden, so erhalten Sie Mitteilung.

Sollte der Täter im Verlauf weiterer Ermittlungen bekannt werden, so erhalten Sie Mitteilung.

Mit freundlichen Grüßen

gez. D'hein
Staatsanwältin

Dieses Schreiben wurde elektronisch erstellt und enthält deshalb keine Unterschrift, wofür um Verständnis gebeten wird.

Gefälschte Abmahn-Mails fordern Bitcoins (10/2014)

- Verbraucherzentrale Rheinland-Pfalz warnt vor gefälschten Abmahnschreiben
- Als Absender sind reale Anwaltskanzleien angegeben
- Empfänger wird beschuldigt, urheberrechtlich geschütztes Videomaterial abgerufen zu haben
- E-Mail enthält Links auf vermutlich Malware-verseuchte Webseiten
- Forderung nach Entschädigungszahlung in Bitcoins

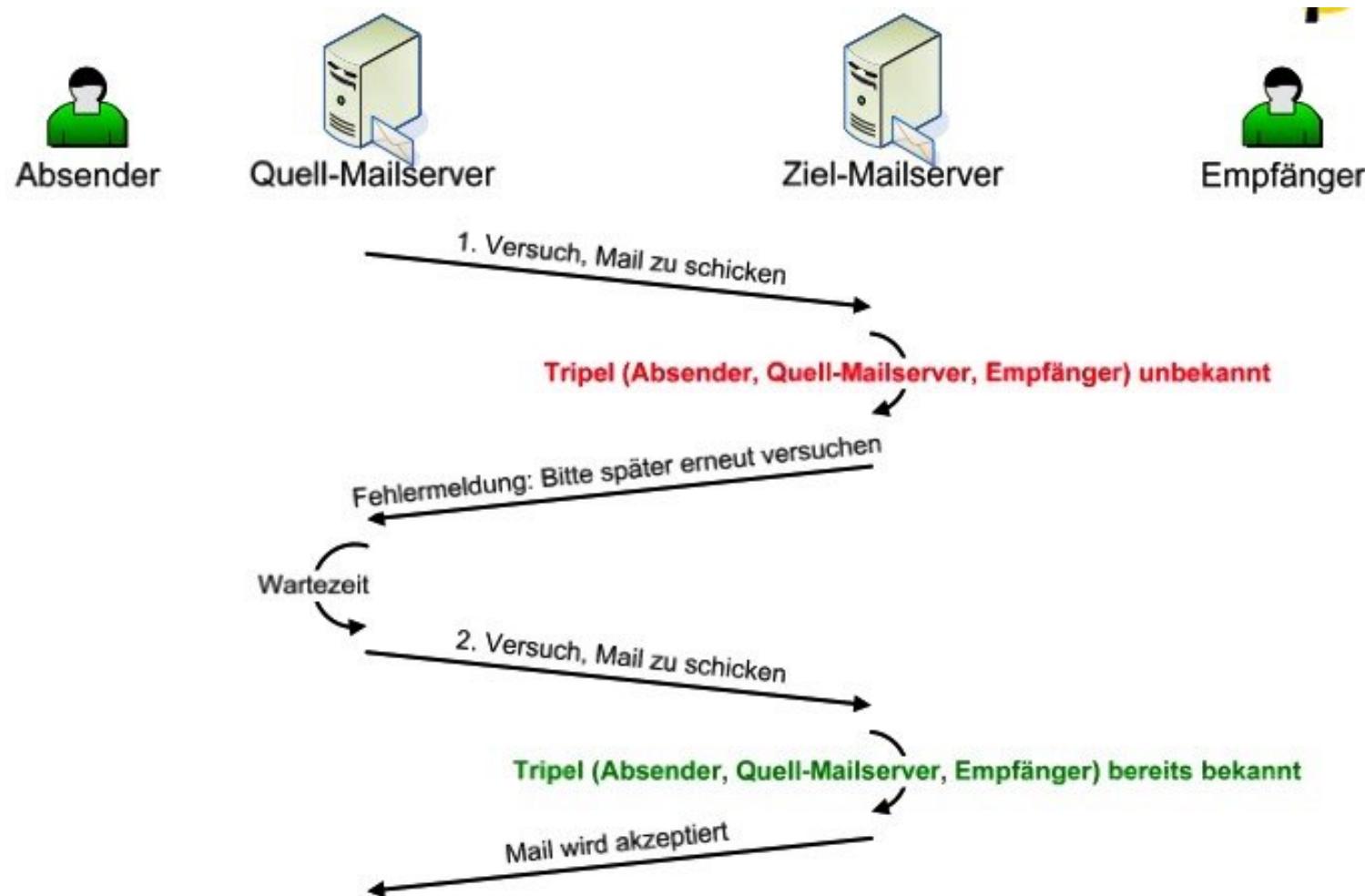
Quelle: <https://www.verbraucherzentrale-rlp.de/porno-phishing-mails>

Klassische Gegenmaßnahmen: Spamfilter

- Software, die eingehende Mails nach Spam durchsucht
- Arten von Spam-Filtern:
 1. Blacklist / Whitelist Ansatz:
Aussperren von Mail-Servern und Mail-Domänen, die üblicherweise von Spammer benutzt werden.
 2. Regelbasiert:
Nachricht wird inhaltlich nach Spam-Merkmalen durchsucht;
sowohl im Header als auch im Body der Mail.
 3. Filtersoftware lernt aus Beispielen:
Neuronale Netze oder Bayes-Filter bewerten Mailinhalte.
- Vor- u. Nachteile dieser Spam-Filter:
 1. Effizient zu implementieren; aber grobgranular, keine inhaltliche Prüfung.
 2. Sehr hohe Erkennungsraten; aber E-Mail muss vollständig entgegen genommen werden,
kontinuierlicher Aufwand für Konfigurationspflege.
 3. Gut in Mail-Clients zu integrieren; aber Erkennungsrate abhängig von Training (NN) bzw.
Modellierung (Bayes).

- Fehlerarten bei der Erkennung
 - Filter, die „automatisch“ Entscheidungen treffen, machen zwei Arten von (systematischen) Fehlern:
 - **Falsch positiv:** Mail wird als Spam erkannt, obwohl sie Ham ist
 - **Falsch negativ:** Mail wird als Ham bewertet, obwohl sie Spam ist
- Welche Fehlerart ist problematischer?
- Policy für Spambehandlung:
 - Spam-Mail löschen und Empfänger ggf. benachrichtigen
 - Spam-Mail markieren und dann ausliefern
 - Welche Variante bevorzugen (unter Beachtung der Fehlerarten)?
 - Vgl. auch Urteil Landgericht Bonn, 15 O 189/13
- Beispiele:
 - SpamAssassin (<http://spamassassin.apache.org/>)
 - Implementiert alle Filterarten (Blacklist, Regelbasis, Bayes-Filter)
 - Zentral und dezentral einsetzbar, fein-granular konfigurierbar
 - Spamfilter als Cloud-Dienst: Mail-Gateway mit Spamfilter bei externem Dienstleister - kein eigener Konfigurationsaufwand, aber “Mitleser”...

Greylisting gegen Spam (1/2)

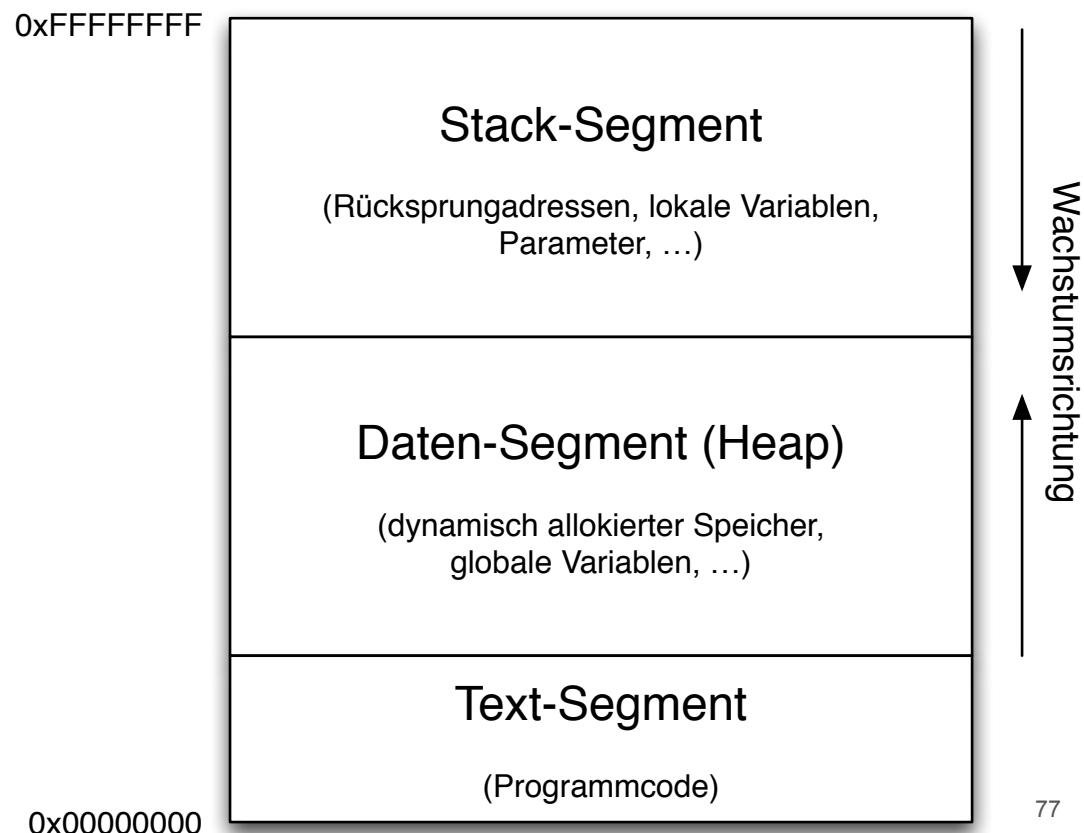


1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

Hier: stack smashing

- Ziel: Ausführen von Code auf fremdem Rechner unter fremden Rechten (z.B. *root*)
- Vorgehen:
 - Auswahl des Ziels:
 - Lokal: Programm, das z.B. mit SUID (Set User ID)-Bit, d.h. mit Rechten des Eigentümers (meist *root*), läuft.
 - Remote: Netzdienst, z.B. Samba-Fileserver
 - Überschreiben interner Programmpuffer, z.B. durch überlange Eingabe
 - Dabei Manipulation z.B. der Rücksprungadresse, dadurch Ausführen von bestimmter Programmsequenz des Angreifers; z.B. Code zum Starten einer Shell

- Speicherabbild eines Programms (am Bsp. Unix)



Anfälliger C-Code

```
1 #include <string.h>
2
3 void kopiere_eingabe (char *eingabe)
4 {
5     char kopie_der_eingabe[128];
6     strcpy(kopie_der_eingabe, eingabe);
7 }
8
9 int main (int argc, char **argv)
10 {
11     kopiere_eingabe(argv[1]);
12 }
```

Hinweis:

Betrifft nicht nur Kommandozeilenparameter, sondern z.B. auch interaktive Eingaben, Datenpakete über Netz, Parsen von Dateien, ...

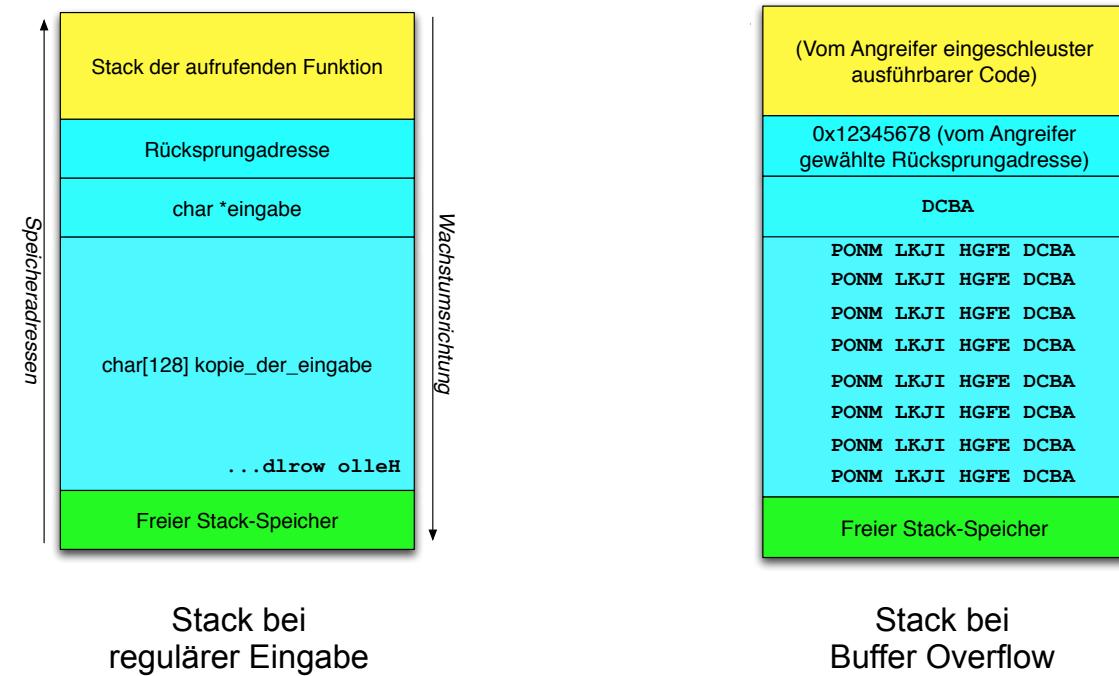
- Kommandozeilenparameter (**argv[1]**) wird vom Angreifer gesteuert.
- Programmierer hat Eingabe < 128 Zeichen angenommen.
- Wenn **strlen(argv[1]) > 127**, dann reicht der reservierte Speicherplatz für die Kopie des Strings nicht aus („buffer overflow“).
- Folge: Andere Stack-Elemente werden überschrieben („stack smashing“).

Ausnutzen von Buffer Overflows in Stack-Segmenten

- Ziel: Stack gezielt überschreiben, so dass
 - Rücksprungadresse auf Angreifer-Code umgebogen wird
 - Angreifer-Code das System kompromittiert (z.B. Starten einer interaktiven Shell oder Nachladen beliebiger Schadprogramme)

```
1 #include <string.h>
2
3 void kopiere_eingabe (char *eingabe)
4 {
5     char kopie_der_eingabe[128];
6     strcpy(kopie_der_eingabe, eingabe);
7 }
8
9 int main (int argc, char **argv)
10 {
11     kopiere_eingabe(argv[1]);
12 }
```

Quelltext



Anmerkung: Darstellung des Stack-Aufbaus vereinfacht!

Kleinere Hürden beim Stack-Smashing

- Rücksprungadresse ist absolut (nicht relativ) anzugeben.
- Lösung: NOPs vor eigentlichem Schadcode:

Rücksprung erfolgt
„irgendwo“ hierhin:

NOP

NOP

NOP

NOP

NOP

NOP

Schadcode beginnt
ab hier:

NOP

NOP

→ mov AH, 1

int 21

...

- Das Stack-Segment bietet nur wenig Speicherplatz für eingeschleusten Code.
- Lösungen: Shellcode kompakt in Assembler programmieren; dynamisches Nachladen von Schadcode.
- Quellcode von proprietärer Software nicht verfügbar.
- Lösung: Fuzzing

- Ziele:
 - Nachbildung des Funktionsaufrufs `system("/bin/sh");`
 - Shellcode darf keine Nullbytes (0x00) enthalten, damit u.a. `strcpy` nicht abbricht.
- Beispiel (Quelle: www.shell-storm.org; Autor: kernel_panik)
`execve ("./bin/sh")`

```
char code[ ] = "\x31\xc9\xf7\xe1\x51\x68\x2f\x2f"
                  "\x73\x68\x68\x2f\x62\x69\x6e\x89"
                  "\xe3\xb0\x0b\xcd\x80";
```
- Größe: 21 Bytes, Plattform: Linux/x86
- Alternative zum Ausführen eigenen Codes: *return-to-libc*, d.h. Einsprung in Standard-Funktionsbibliothek mit eigenen Parametern (z.B. wiederum Aufruf von `system()`).

- Am Besten: Sicheres Programmieren, z.B. `strncpy` statt `strcpy`
 - Unterstützung durch Code-Analyse-Tools, z.B. Splint
- Stack-Guarding:
 - Beim Aufruf einer Unterfunktion wird hinter der Rücksprungadresse ein Kontrollzeichen („Canary“) abgelegt.
 - Vor dem Rücksprung wird geprüft, ob das Kontrollzeichen noch intakt ist.
 - Variante: Mehrere Kopien der Rücksprungadresse.
- Nicht-ausführbare Stacks (non-executable stack)
 - Code auf dem Stack wird vom Betriebssystem generell nicht ausgeführt, damit auch kein eingeschleuster Shellcode.
 - Inzwischen von vielen Prozessoren hardware-unterstützt („NX bit“)
 - Schützt aber weder vor Shellcode auf dem Heap noch vor *return-to-libc*
- Address space layout randomization (ASLR)
 - Speicherbereiche u.a. für Stack werden zufällig gewählt.
 - Angreifer hat es schwerer, die richtige Rücksprungadresse anzugeben.

Weitere Aspekte

- Heap Corruption
 - Überschreiben von programminternen Datenstrukturen mit vom Angreifer vorgegebenen Werten
- Problematisch sind nicht nur String-Operationen
 - int-Überlauf
 - Schleifen mit Abbruchkriterien, die von der Angreifer-Eingabe nicht erfüllt werden
 - Multi-byte character encodings (Unicode)
- Format String Attacks
 - `printf(buffer)` statt `printf("%s", buffer)` bei Benutzereingaben wie "%x"
 - Überschreiben interner Datenstrukturen bei Anwendung z.B. auf `sprintf()`
- Literatur:
 - Buffer Overflow Attacks. Detect, Exploit, Prevent; Syngress Media 2005

Account/Password Cracking

- Passworteingabe ist das am weitesten verbreitete Authentifizierungsverfahren
- Ziel des Angriffs: „Erraten“ von Benutzername und Passwort
- Varianten:
 - Brute-Force Angriff
 - Dictionary Attack (Wörterbuchangriff)
 - Brechen des Hash-/Verschlüsselungsalgorithmus für das Passwort
 - Social Engineering
- Password Cracking am Beispiel älterer UNIX-Systeme:
 - Administrator (`root`) vergibt Benutzernamen
 - Eintrag in `/etc/passwd`
 - Datei für **alle** lesbar
 - Format des Eintrags

`huber:Ad9%y?SmW+zP&:23:17:Herbert Huber:/home/huber:/bin/bash`

`Username:Password:UID:GID:Gecko-String:Home-Verzeichnis:Shell`

UNIX-Authentifikation: User/Password

- Benutzer wählt Passwort
 - Passwort wird mit sich selbst als Schlüssel verschlüsselt und verschlüsselt gespeichert in /etc/passwd:
z.B. :Ad9%y?SmW+zP:<
 - Auch root kennt Passwort **nicht**
- Authentisierung:
 - Eingegebenes Passwort wird mit sich selbst verschlüsselt und mit dem in /etc/passwd verglichen.
- Verschlüsselungsalgorismus crypt (pwd, salt) bekannt
- Dictionary Attack:
 - Angreifer verschlüsselt Wörter aus Wörterbuch und vergleicht verschlüsselte Strings mit Einträgen in /etc/passwd
- Verhinderung der Dictionary Attack
 - Zus. Parameter salt in crypt
 - 12 Bit Zahl: $0 \leq \text{salt} < 4096$
 - Bei Initialisierung zufällig gewählt
 - Die ersten 2 Zeichen im Passwort String sind salt; im Beispiel: Ad
- Brute Force Dictionary Attack:
 - Angreifer muss Wörterbuch für **jeden** Benutzer mit dessen salt verschlüsseln und vergleichen
 - Bei heutiger Rechenleistung kein echtes Problem.
- Verhinderung z.B. durch:
 - Shadow Password System (nur root kann verschl. Passwort lesen)
 - One-Time Passwords
 - Alternativen zu crypt()

Implementierung

Was passiert im Kernal?

- In die Verschlüsselung fließen zwei zufällig gewählte Zeichen ("Salt") ein.
- Salt wird in der Ausgabe im Klartext hinterlegt.
- Angreifer müsste 4096 Werte pro Wörterbuch-Eintrag vorab berechnen.
- (Aus heutiger Sicht kein großer Aufwand mehr)
- Neuerer Ansatz:
 - ❑ Verschlüsselte / gehashte Passwörter in /etc/shadow ausgelagert.
 - ❑ Nur noch „root“ hat überhaupt Lesezugriff, reguläre Benutzer kommen nicht an die verschlüsselten / gehaschten Passwörter heran.
 - ❑ Längeres Salt.
 - ❑ Aufwendigere Hashverfahren, z.B. SHA-512, in mehreren Runden angewandt.
 - ❑ Nutzung von "Slow Hash Functions" wie PBKDF2, bcrypt, scrypt.

```
1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main(void)
5 {
6     char *ergebnisAA, *ergebnisxy;
7
8     ergebnisAA = crypt("GeheimesPasswort", "AA");
9     printf("Salt AA: %s\n", ergebnisAA);
10
11    ergebnisxy = crypt("GeheimesPasswort", "xy");
12    printf("Salt xy: %s\n", ergebnisxy);
13
14    return 0;
15 }
```

Ausgabe: Salt AA: AA3w0THiFXV1A
Salt xy: xyj.4bikXtQ1o

Back Doors, Trap Doors

- Ziel: Angreifer will dauerhaften Zugang (Hintereingang) zu einer bereits kompromittierten Maschine
 - An der Betriebssystem-Authentisierung vorbei
 - Mit speziellen Rechten (z.B. root)
- Mechanismen z.B.:
 - „Verstecktes“ eigenes SUID-root Programm mit „shellcode“.
 - SUID-root Systemprogramm durch eigene Version mit versteckter Funktionalität austauschen.
 - Installation eines “versteckten” Netzdienstes, der zu bestimmten Zeiten einen Netz-Port öffnet und auf Kommandos wartet.
 - Eintrag in `.rhosts`-Datei von root bzw. `authorized_keys` für SSH-Zugang
- Detektion durch Integritäts-Checks:
 - Kryptographische Prüfsummen:
 - aller installierten Programme
 - Konfigurationsdateien
 - regelmäßige Überprüfung
 - Überprüfung der offenen Ports und der aktivierten Netzdienste
 - Suche nach ungewöhnlichen SUID/Sgid-Programmen
- Reaktion bei erkannten Hintertüren:
 - Vollständiges Entfernen der Schadsoftware wirklich möglich?
 - Ggf. Maschine neu bzw. aus „sauberem“ Backup aufsetzen.
 - Verwundbarkeit, die zur Kompromittierung geführt hat, muss behoben werden!

- Begriffsbildung:
 - Zusammensetzung aus *root* (= Administratorkennung unter UNIX/Linux) und *Toolkit* (= Werkzeugkasten)
 - Ursprünglich Bezeichnung für zueinander komplementäre UNIX-Systemprogramme mit eingebauten Backdoors (1. Generation Rootkits)
- Typischer Ablauf:
 - Angreifer kompromittiert Maschine und erlangt root-Berechtigung
 - Angreifer installiert Rootkit
 - Werkzeuge aus dem Rootkit bereinigen Spuren u.a. in Logfiles
 - Backdoors ermöglichen kontinuierlichen root-Zugang für Angreifer
 - Rootkits der 1. Generation bestehen aus eigenen Varianten von Kommandos und Programmen wie *ps*, *ls*, *top*, *du*, *find*, *netstat*, *passwd*, *sshd*, ...
 - Alle ersetzen Systembefehle verstecken Prozesse, Dateien etc. des Angreifers.
- Detektion über Host-IDS und Tools wie *chkrootkit*

Rootkits (Forts.)

- Rootkits der 2. Generation
 - Motivation: Alle Systemprogramme einzeln auszutauschen ist aus Angreifersicht aufwendig und fehleranfällig.
 - Neuer Lösungsansatz: Betriebssystemkern (Kernel) modifizieren
→ Dateien, Prozesse etc. des Angreifers werden vor allen Systemprogrammen versteckt
- LKM-Rootkits unter Linux
 - Loadable Kernel Module → OS-Kern wird zur Laufzeit erweitert
 - Kernelmodul ersetzt Systemfunktionen z.B. zum
 - Auslesen von Verzeichnisinhalten (Verstecken von Dateien)
 - Zugriff auf die Prozessliste (Verstecken von Malware)
 - Ggf. mit Backdoor (spezieller Funktionsaufruf liefert root-Berechtigung)
- Prävention
 - Nachladen von Kernelmodulen komplett deaktivieren
- Detektion
 - „Sauberes“ System nur nach Booten z.B. von USB-Stick oder CD

Moderne Ausprägungen

- Hypervisor-level Rootkits:
 - Rootkit übernimmt das komplette System
 - Ursprüngliches Betriebssystem wird als virtuelle Maschine ausgeführt
 - Beispiel: Blue Pill (2006)
- Bootkits:
 - Angreifer ersetzt Bootloader durch Malware
 - Hebelt auch Schutz durch komplett verschlüsselte Festplatten aus
 - Beispiele: Evil Maid Attack, Stoned Bootkit, Alureon
- Hardware- / Firmware-Rootkits:
 - Rootkit installiert sich z.B. im BIOS oder in der Firmware der Netzwerkkarte (Beispiel: Delugré-NetXtreme Rootkit 2010)
- Zuverlässige Detektion schwierig
 - Timing: Erkennen der rootkit-virtualisierten Umgebung durch veränderte Dauer z.B. von Systemaufrufen. (Problem: zu viele False-Positives)
 - Externe Analyse (Booten von CD)

- Firma RSA Security stellt u.a. weltweit stark verbreitete Token zur Authentifizierung her (RSA SecurID)
- Spear-Phishing Angriff auf RSA-Mitarbeiter: Excel-Attachment „2011 Recruitment Plan.xls“, vermutlich mit Excel 2007 geöffnet.
- Eingebettetes SWF-File nutzt Adobe-Flash-Player-Lücke aus.
- Schadcode (Abwandlung von „poison ivy“) späht Mitarbeiterrechner aus und überträgt u.a. Passwörter an den Angreifer.

- Folgen:
 - SecurID-Quellen und -Seeds werden ausgespäht
 - US-Rüstungsunternehmen Lockheed Martin wird mit „nachgebauten“ SecurID-Token gehackt; zahlreiche weitere Unternehmen betroffen
 - Rund 40 Millionen SecurID-Token werden ausgetauscht

Security-Segen oder -Fluch?

- Browser werden mehr und mehr zum vollwertigen “Betriebssystem”
- Neue Funktionen ..., z.B.:
 - Web Storage API
 - WebSockets API
 - Cross-Origin Resource Sharing
- ... bergen neue Risiken, z.B.:
 - Benutzer stellen Rechenleistung und Speicherplatz zur Verfügung
 - Clients bauen (beliebige) Netzverbindungen auf
- Beispiel: distPaste (Jan-Ole Malchow, FU Berlin)
 - <http://www.dfn-cert.de/dokumente/workshop/2013/FolienMalchow.pdf>
 - Speichert Dateien ggf. verteilt auf mehrere Clients (2,5 MB pro Node)
 - Wer ist verantwortlich für die Inhalte?

1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

Cross Site Scripting

- Einbetten von Schadcode in (vertrauenswürdigen) anderen Code
- Beispiel:
 - Alice betreibt eine Webseite mit Gästebuch-Funktion.
 - Mallet hinterlässt einen Gästebuch-Eintrag, der JavaScript enthält.
 - Bob ruft die Gästebuch-Einträge auf der Website von Alice ab und führt dabei den JavaScript-Code von Mallet aus.
- Typisches Ziel bei XSS:
 - Sensible Daten, z.B. Cookies, an den Angreifer übertragen.
 - Mallet kann sich damit als Bob ausgeben
(Identitätsdiebstahl, impersonation attack)
- Häufig im Zusammenhang mit HTML und JavaScript, betrifft aber nicht nur Webbrowser
(z.B. Skype 2011: JavaScript in Profilfeldern)

- Anwendung prüft Benutzereingaben nicht ausreichend
- Im Gästebuch-Beispiel:
 - Gästebuch-Webanwendung sollte Einträge mit (Schad-)Code nicht akzeptieren
 - Client (Bob) kann von Server (Alice) gewünschten Code nicht von bösartigem Code (Mallet) unterscheiden.
- Im Skype-Beispiel:
 - Skype-Client von Mallet erlaubt JS-Code im Feld „Mobiltelefonnummer“.
 - Skype-Client von Bob führt diesen Code ungeprüft aus.
- JavaScript kann u.a. HTML-Formulare automatisch ausfüllen und abschicken; wurde missbraucht z.B. für
 - Sofort-Kauf von Ebay-Angeboten
 - Beleidigende oder Spam-artige Einträge in Internet-Foren
 - Generieren von URLs, die Benutzer auf fremde Webseiten umleiten und dabei sensible Daten als Parameter übergeben.

DOM-basiertes (lokales) XSS

- Lokal bedeutet hier: Ohne Beteiligung eines Webservers
- Auslöser: JavaScript-Funktion prüft übergebene Parameter nicht
- Beispiel:

```
<HTML>
  <TITLE>HTML-Beispieldokument DOM-XSS</TITLE>
  Hallo
  <SCRIPT>
    var pos=document.URL.indexOf("username")+9;
    document.write(document.URL.substring(pos,document.URL.length));
  </SCRIPT>
  <BR/>
  Dies ist ein Beispiel-HTML-Dokument.
</HTML>
```

- Aufruf mit:

```
http://www.example.com/index.html?username=<script>alert("XSS-Problem!")</script>
```

- Als Parameter übergebener Code wird von anfälligen Browsern ausgeführt.

Reflexives (nicht-persistentes) XSS

- Ablauf:
 - Webserver liefert Webseite mit Inhalt aus, der vom Benutzer übergebene (und somit nicht-persistente) Parameter (inkl. JavaScript-Code) enthält.
 - Mallet bringt Alice dazu, einen Link mit entsprechenden Parametern anzuklicken
- Beispiel:
 - Alice klickt folgenden Link an:

[http://suchmaschine.example.com/?suchbegriff=<script type="text/javascript">alert\("Alice, Du hast ein XSS-Problem!"\)</script>](http://suchmaschine.example.com/?suchbegriff=<script%20type%3D%22text/javascript%22%3Ealert(%27Alice,%20Du%20hast%20ein%20XSS-Problem!%27)%3C%2Fscript%3E)

- Webserver liefert folgendes Dokument aus:

```
<HTML>
  <TITLE>Suchmaschine: Ergebnisse</TITLE>
  Ihr Suchbegriff war: <script type="text/javascript">alert(„Alice,
    Du hast ein XSS-Problem!“)</script>
  <BR/>
  Hier sind Ihre Ergebnisse: ...
</HTML>
```

Persistentes (stored) XSS

- Schadcode wird vom Webserver gespeichert und bei jeder Anfrage ausgeliefert.
- Vgl. Gästebuch-Beispiel; Eintrag enthält z.B.

Tolle Webseite! `<script type="text/javascript">alert("Aber mit XSS-Problem!")</script>`

- Dadurch sehr breit gestreuter Angriff
- Besonders problematisch, wenn der „verseuchte“ Webserver als besonders vertrauenswürdig konfiguriert ist
- Gegenmaßnahme:
 - Webapplikation muss Script-Code aus Benutzereingaben entfernen oder „ungefährlich“ machen.
 - Script-Code kann anhand der Meta-Zeichen, z.B. <, erkannt werden.
 - Client-seitig: JavaScript deaktivieren oder Plugins wie NoScript verwenden
 - Content-Security-Policy - Spezifikation vertrauenswürdiger Script-Quellen, alles andere wird nicht ausgeführt

Angriff auf Issue Tracking System von Apache

- apache.org nutzt Atlassian JIRA als Issue Tracking System
- 5. April 2010: Angreifer legt neue Issue (INFRA-2591) an:
ive got this error while browsing some projects in jira <http://tinyurl.com/XXXXXXXXXX> [obscured]
 - tinyurl: Dienst, um URLs zu kürzen und „dauerhaft“ zu machen
 - Lange URL enthält XSS, um Cookies von JIRA-Usern zu stehlen
 - Auch Administratoren von JIRA werden Opfer
 - Gleichzeitig startet Angreifer Brute-force Passwort-Angriff gegen Anmeldeseite login.jsp
- 6. April 2010: Angreifer hat Administrator-Rechte auf JIRA
 - Angreifer deaktiviert Benachrichtigung für ein Projekt
 - Ändert Pfad für den Upload von Attachments; Pfad erlaubt Ausführung von JSP und ist schreibbar für JIRA-User
 - Erzeugen neuer Issues mit einer Vielzahl von Attachments
 - Eines der Attachments ist JSP zum Durchsuchen und Kopieren von Dateisystem-Inhalten

Angriff auf Issue Tracking System von Apache (Forts.)

- 6. April 2010: (Fortsetzung)
 - Angreifer kopiert sich damit viele Home-Directories von JIRA-Usern
 - Weiteres JSP mit Shell-Backdoor für das System wird installiert
- 9. April 2010: Angreifer hat jar-Datei installiert, die alle Benutzernamen und Passwörter protokolliert
 - Eines der Passwörter funktioniert auch für den Rechner brutus.apache.org
 - Für diesen Account ist sudo aktiviert.
 - Damit voller Root-Zugriff auf brutus (Server für JIRA u. Wiki).
 - Angreifer nutzt dies zum Ausspähen von Login-Daten für den Subversion-Server minotaur
 - Angreifer beginnt, JIRA-Passwort-Reset-Mails zu verschicken, damit sich noch mehr Benutzer auf dem kompromittierten System anmelden.
 - Dadurch wird der Angriff schließlich erkannt.

Erwarteter Aufruf

`http://webserver/cgi-bin/find.cgi?ID=42`

Erzeugte SQL-Abfrage

`SELECT autor, text FROM artikel WHERE ID=42`

Aufruf mit SQL-Injektion

`http://webserver/cgi-bin/find.cgi?`

`ID=42;UPDATE+USER+SET+TYPE="admin"+WHERE+ID=2`

Erzeugte SQL-Abfrage: 2 Befehle!

`SELECT autor, text FROM artikel WHERE ID=42; UPDATE
USER SET TYPE="admin" WHERE ID=2`

1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

Sniffer: Abhören des Netzverkehrs

■ Local Area Network (LAN):

- Oft gemeinsam genutztes Medium (shared medium), z.B. WLAN, Ethernet ohne Switches
- Netzwerk-Karten können im Prinzip gesamten Verkehr mithören, aber
- geben nur die an den Rechner adressierten Pakete weiter
- Gefahr: "Promiscuous Mode":
 - Einstellung der Karte
 - Im Promiscuous Mode werden **alle** Pakete gelesen und ans OS durchgereicht

■ Wide Area Network (WAN):

- Jeder Vermittlungsrechner kann Nachrichten „mitlesen“, z.B.
Mirror-Ports an Routern
- "Anzapfen" von Leitungen (z.B. durch Geheimdienste)

■ Tools:

- Übergang zwischen Werkzeug des System- sowie Netzadministrators und Cracker-Tool sind fließend
- `tcpdump`, `ngrep`
Standard-Werkzeuge in vielen UNIX/Linux-Distributionen
- `wireshark`
Packet-Analyzer (Linux, Windows)
-

```
match: nm_ifi.lmu.de
#####
U 129.187.15.116:60267 -> 10.156.33.53:53
.....www.nm_ifi.lmu.de.....
#
U 129.187.15.116:51458 -> 10.156.33.53:53
.....www.nm_ifi.lmu.de.....
#
U 10.156.33.53:53 -> 129.187.15.116:51458
.....www.nm_ifi.lmu.de.....Q...pchege01.....Q...dns0...hostmaster..w..#0.....:Q.
#
U 10.156.33.53:53 -> 129.187.15.116:60267
.....www.nm_ifi.lmu.de.....Q...pchege01.....Q...T.....Q...dns2.lrz.....Q...dns1.....Q...dns1.
\.....Q...dns3.....Q...acheron.....Q...dns0.....Q...T...n.....Q...T.....Q...dns1.....Q...L.....S.
\.....Q...C...W.....Q...L.....S.....Q...{.....Q...0.....#.....Q...L.....Q...L@.....Q...
#####
T 129.187.15.116:50224 -> 141.84.218.31:80 [AP]
GET /itsec HTTP/1.1. Host: www.nm_ifi.lmu.de..User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_2) AppleWebKit/534.51.22 (KHTML, like Gecko) Version/5.1.1 Safari/534.51.22..Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8..Accept-Language: de-de..Accept-Encoding: gzip, deflate..Connection: keep-alive...
##
T 141.84.218.31:80 -> 129.187.15.116:50224 [AP]
HTTP/1.1 302 Found..Date: Thu, 27 Oct 2011 07:38:14 GMT..Server: Apache/2.2.21 (Ubuntu) PHP/5.4.24. X-Powered-By: PHP/5.4.24..Location: http://www.nm_ifi.lmu.de/teaching/Vorlesungen/2011ws/itsec..Keep-Alive: timeout=15, max=100..Connection: Keep-Alive..Transfer-Encoding: chunked..Content-Type: text/html....1 .....0....
##
T 129.187.15.116:50224 -> 141.84.218.31:80 [AP]
GET /teaching/Vorlesungen/2011ws/itsec HTTP/1.1..Host: www.nm_ifi.lmu.de..User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_2) AppleWebKit/534.51.22 (KHTML, like Gecko) Version/5.1.1 Safari/534.51.22..Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8..Accept-Language: de-de..Accept-Encoding: gzip, deflate..Accept-Language: de-de..Connection: keep-alive...
##
T 141.84.218.31:80 -> 129.187.15.116:50224 [AP]
HTTP/1.1 301 Moved Permanently..Date: Thu, 27 Oct 2011 07:38:14 GMT..Server: Apache/2.2.21 (Ubuntu) PHP/5.4.24..Location: http://www.nm_ifi.lmu.de/teaching/Vorlesungen/2011ws/itsec/..Keep-Alive: timeout=15, max=99..Connection: Keep-Alive..Transfer-Encoding: chunked..Content-Type: text/html; charset=iso-8859-1....151..



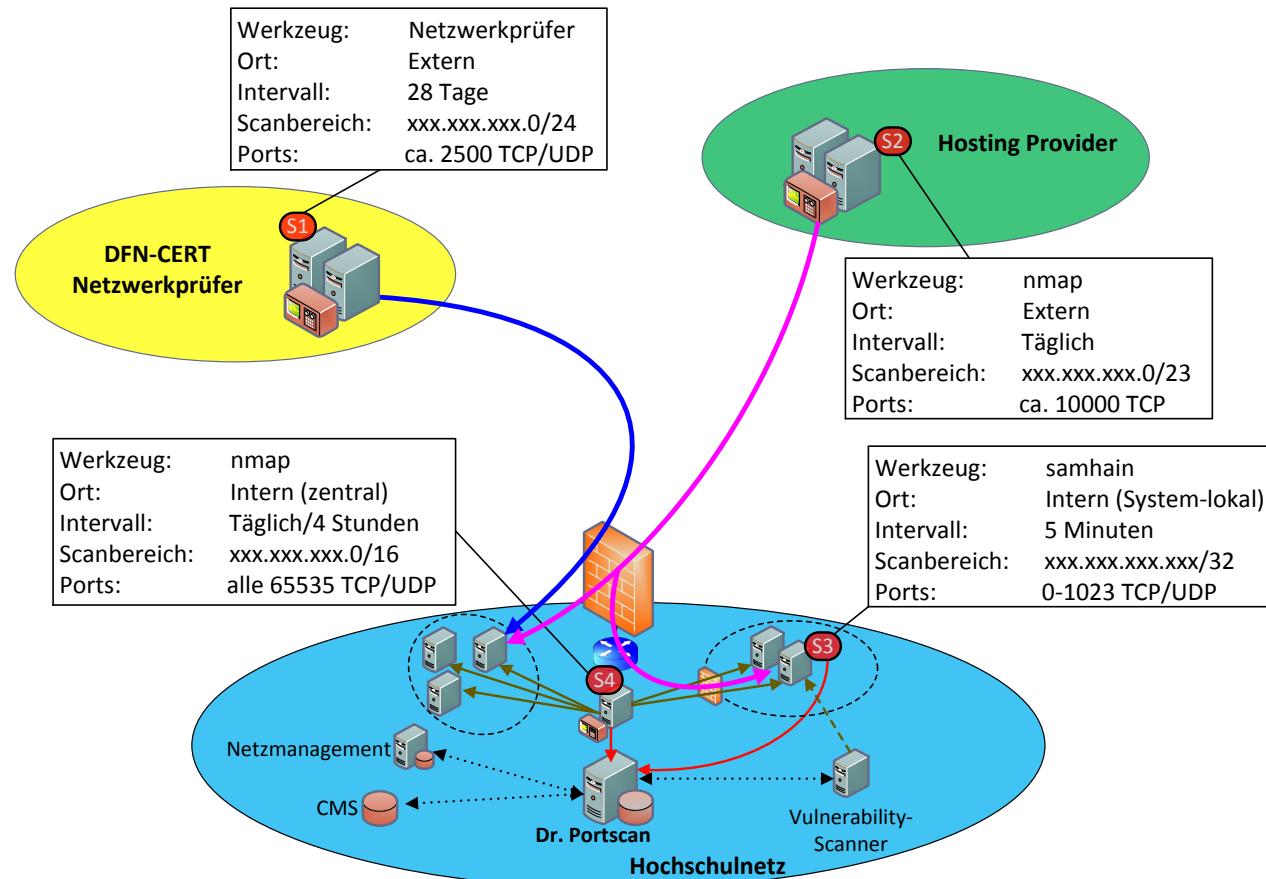
..<TITLE>301 Moved Permanently</TITLE>..<HEAD><BODY>..<H1>Moved Permanently</H1>.The document has moved <A href="http://www.nm_ifi.lmu.de/teaching/Vorlesungen/2011ws/itsec/">here</A>..<P>..<HR>..<ADDRESS>Apache/2.2.21 Server at www.nm_ifi.lmu.de Port 80</ADDRESS>..</BODY>..</HTML>....0....
```

- Suchen auf entferntem Rechner nach „offenen“ Ports
 - Versuch eines Verbindungsaufbau / pro Port
 - Falls erfolgreich: Port ist „offen“
- Damit Identifikation von Diensten
- Gezielte Suche nach Rechnern, die Dienste mit bekannten Schwächen anbieten
- Auch hier ist der Übergang zwischen nützlichem Werkzeug und Cracker Tool fließend
- Port-Scans werden oft als Angriff gewertet und deshalb getarnt durchgeführt
- Beispiel:

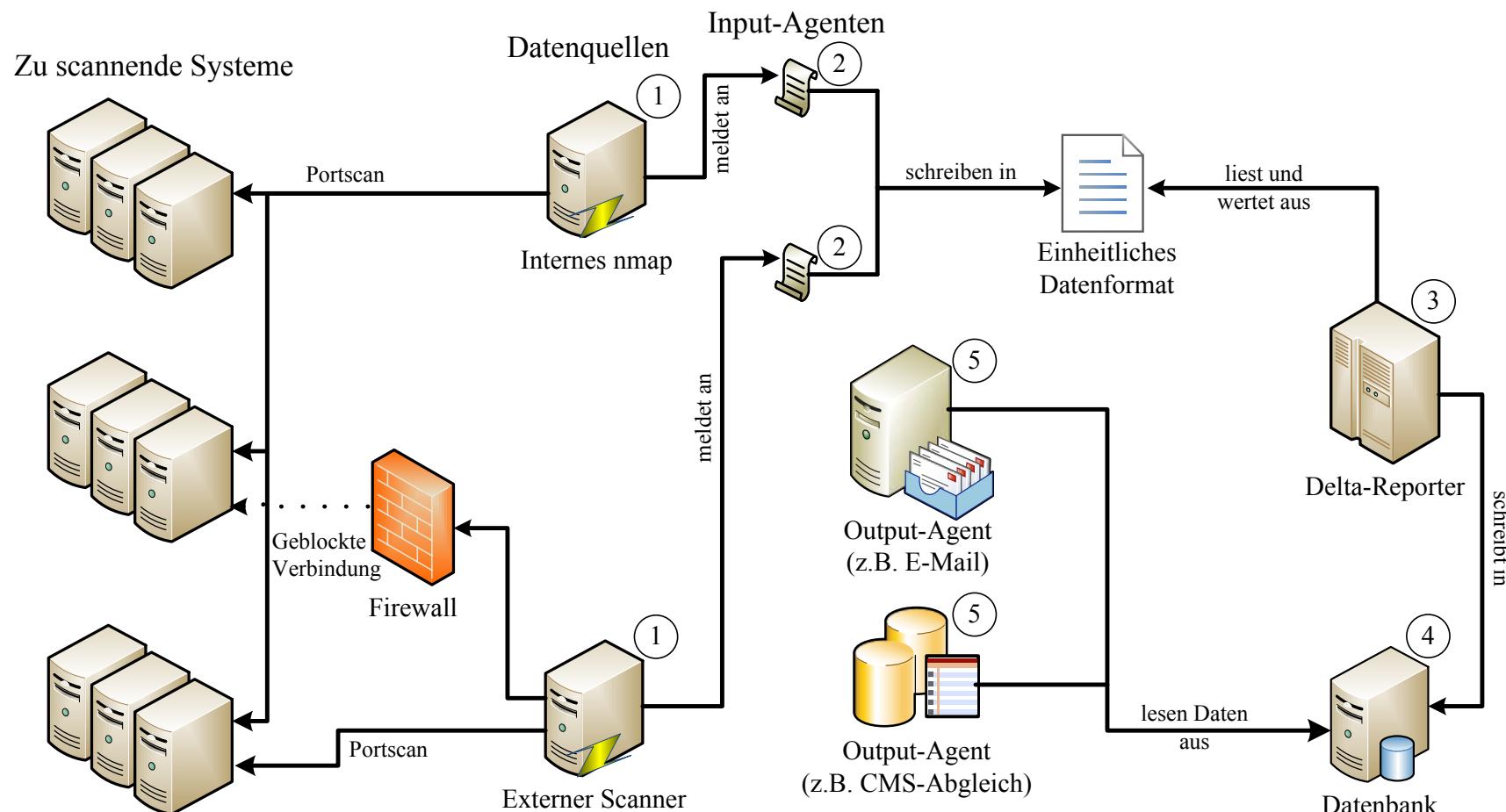
- nmap

```
Nmap scan report for www.nm.ifi.lmu.de (141.84.218.31)
Host is up (0.018s latency).
rDNS record for 141.84.218.31: pchege01.nm.ifi.lmu.de
Not shown: 65532 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.15 - 2.6.27, Linux 2.6.24 - 2.6.26 (Debian), Linux 2.6.27 (Ubuntu 8.10)
Uptime guess: 74.352 days (since Sun Aug 14 01:24:27 2011)
```

Proaktive Netzüberwachung mit Portscans



Auswertung von Portscan-Ergebnissen



1. Grundlegendes zur Angriffsanalyse

- Notation von Sicherheitsproblemen
- Angreifermodelle
- Begriffe und Zusammenhänge

2. Ausgewählte technische Angriffsvarianten

- Denial of Service (DoS und DDoS)
- Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
- E-Mail-Security (Spam)
- Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
- Web-basierte Angriffe (XSS, ...)
- Netzbasierte Angriffe (Sniffing, Portscans, ...)

3. Bewertung von Schwachstellen

- Common Vulnerability Scoring System (CVSS)
- Zero Day Exploits

Motivation für Schwachstellenbewertung

- Hauptziel: Priorisierung
 - Wie wichtig ist es, eine Schwachstelle (schnell) zu beseitigen?
 - Welche von mehreren gleichzeitig bekannten Schwachstellen ist die dringendste?
- Betrifft sowohl die **Entwickler** als auch die **Betreiber** von Software/Systemen.
- Idee: Quantitative Bewertung von Schwachstellen anhand einer definierten Menge verschiedener Charakteristika
=> Jeder Schwachstelle wird ein Zahlenwert zugeordnet.
Problem: Objektivität / Einheitlichkeit.
- CVSS-Ansatz: Dreiteilung in unveränderliche bzw. zeitlich und räumlich variable Charakteristika.

Common Vulnerability Scoring System v4

- CVSS ist an der CMU (Carnegie Mellon University) entstanden und wird inzwischen von FIRST (Forum of Incident Response and Security Teams) gepflegt. v4 (v4.1 erschienen Juni 2024) V3.1 aktuell de-facto Standard.
- Vier Gruppen von Bewertungskennzahlen:
 - **Base Metrics:** Grundlegende Eigenschaften der Verwundbarkeit
 - **Threat Metric:** Umsetzung - Proof-of-Concept Code oder aktive Exploits
 - **Environmental Metrics:** Anwenderspezifische Eigenschaften der Verwundbarkeit
 - Supplemental Metrics: Extrinsische Attribute um den Kontext zu berücksichtigen

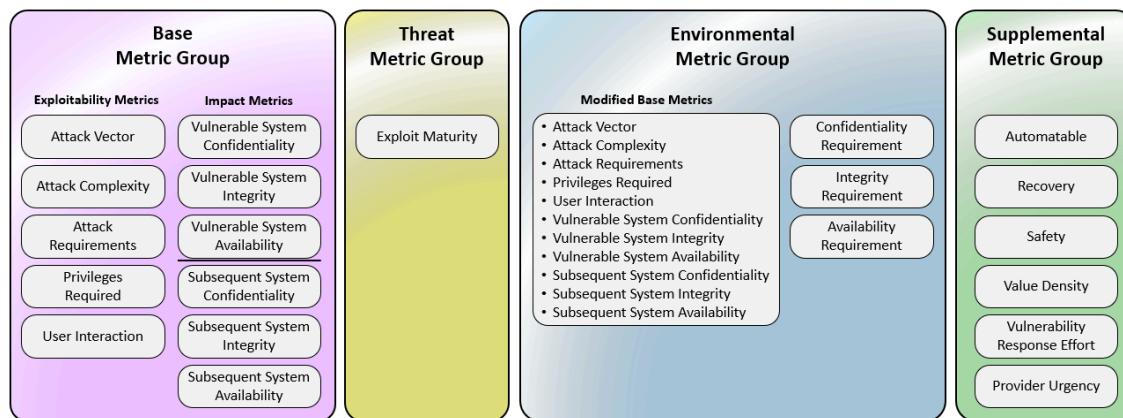


Figure 1: CVSS Metric Groups

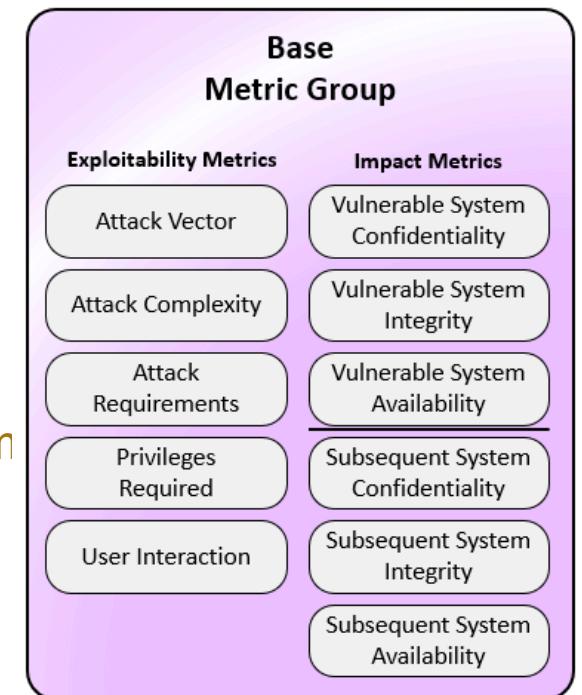
Bildquelle:
<https://www.first.org/cvss/v4-0/cvss-v40-specification.pdf>

- Base Metrics werden oft von Herstellern / Sicherheitsunternehmen veröffentlicht

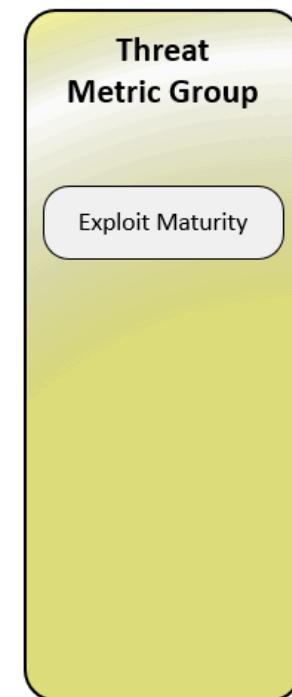
- **Input:**
 - Einfache Bewertung von Schwachstellen durch vorgegebene Fragen und Antwortmöglichkeiten.
- **Outputs:**
 - CVSS-Score (= Zahl) zwischen 0,0 (harmlos) und 10,0 (Katastrophe)
 - CVSS-Vektor = kompakter String, Kurzfassung des gesamten Inputs
- **Scoring-Formel:**
 - Die pro Frage (Metric) gewählte Antwort beeinflusst den Score.
 - Der konkreter Einfluss-Wert ist jeweils empirisch definiert; hierin steckt die “Intelligenz” bzw. Praxiserfahrung von CVSS.
 - Als CVSS-Anwender muss man “nur” die zur Schwachstelle pro Metric passende Antwortmöglichkeit auswählen.
 - Den numerischen CVSS-Score gibt der “CVSS Calculator” aus z.B. <https://www.first.org/cvss/calculator/4.0>

Base Metrics

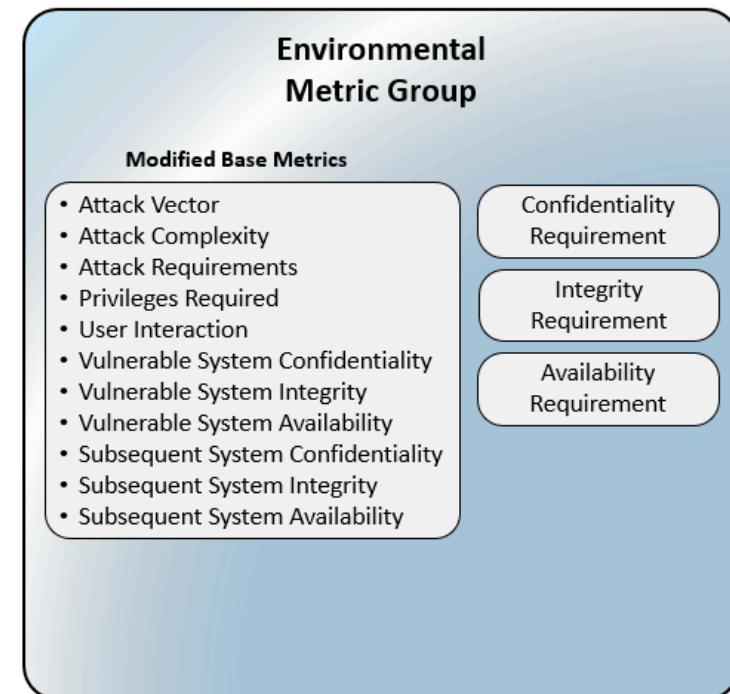
- Die **Base Metrics** bewerten die intrinsischen, konstanten Eigenschaften von Schwachstellen; drei Untergruppen: **Exploitability metrics**, **Impact metrics** und **Scope**.
- **Exploitability metrics** (technische Aspekte der Schwachstelleausnutzung):
 - **Attack vector**: Physisch, auf dem Rechner, vom LAN aus, via Internet?
 - **Attack complexity**: Trivial, anspruchsvoll?
 - **Attack Requirements**: Spezielle Umgebungs- oder Ausführungsbedingungen nötig?
 - **Privileges required**: Jeder, reg. User, Admin?
 - **User interaction**: Angreifer braucht User?
- **Impact metrics** (Auswirkung der Schwachstelle):
 - **Auswirkungen auf C, I, A für verwundbares und Nachfolgesystem**
 - **Jeweils gar nicht, gering oder stark?**



- Die Threat Metrics bewerten den jeweils (zeitlich) aktuellen und damit variablen Stand der Schwachstelle.
- Exploit code maturity?
 - Not defined: nimmt den Worst Case an
 - Attacked: Erfolgreiche Angriffe bekannt
 - Proof of Concept: PoC Code ist öffentlich
 - Unreported: Keine PoC bekannt



- Die **Environmental Metrics** bewerten die Schwachstelle im Hinblick auf das Einsatzgebiet des betroffenen Systems; sie unterscheiden sich also z.B. je nach Organisation.
- **C, I, A Requirements?**
 - Undefined; oder:
 - Abstufung: Gering; mittel; hoch
- **Modified Base Metrics?**
 - Falls Sicherheitsmaßnahmen im Einsatz sind, die sich auf die Base-Metric-Eigenschaften der Schwachstelle auswirken, können diese individuell modifiziert werden.
 - Default: not defined.



- Die **Supplement Metrics** sind optional und bewerten zusätzliche externe Faktoren.
- **Automatable**? Kann der Angreifer automatisiert mehrere Ziele angreifen?
- **Recovery**? Resilienz des Systems. Wie schnell kommt man zu stabilem Zustand?
- **Safety**? Impact des Angriffs auf Safety?
- **Value Density**? „Dichte“ des Angriffs - Angriffsziel hat wenig Ressourcen (z.B. einzelner Email-Client) oder sehr viele (z.B. zentraler Email Server).
- **Vulnerability Response Effort**? Wie hoch ist der Aufwand um auf den Angriff zu reagieren?
- **Provider Urgency**? Provider liefern zusätzliche Einschätzungen des Schweregrades?

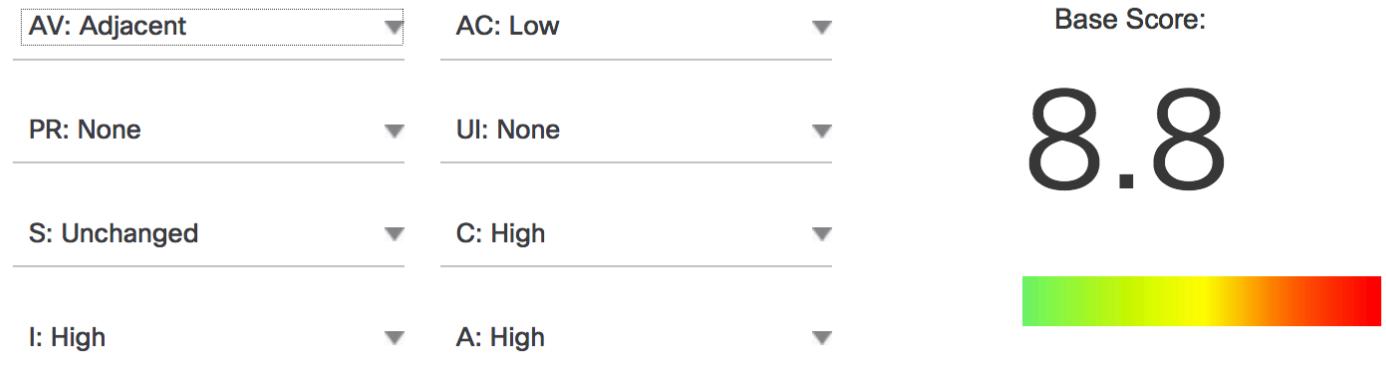


- CVSS v3 Score:
 - Man muss **Base Score**, **Temporal Score** und **Environmental Score** unterscheiden.
 - “CVSS Score” ist oft Synonym für **CVSS Base Score** - nur der ist verpflichtend
- Bei CVSS v2 galt:
 - **Base Score** \geq **Temporal Score** \geq **Environmental Score**
 - Beispiele:
 - Praxisrelevanter Score ist niedriger, wenn offizieller Fix verfügbar ist.
 - Praxisrelevanter Score ist niedriger, wenn CIA Requirements niedrig sind.
- Ab CVSS v3 gilt o.g. Ungleichung nicht mehr zwingend wg. optionaler **Modified Base Metrics**...
- CVSS v4 **Thread Metric Group** realisiert Zeitbezug (Temporal Score)
- CVSS v4 Supplemental Metric Group bilden Kontext und äußere Attribute ab

- CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:X/RL:X/RC:X/CR:X/IR:X/AR:X/MAV:X/MAC:X/MPR:X/MUI:X/MS:X/MC:X/MI:X/MA:X
 - Cisco: CVSS v3 Base Score: 8.8
-

Base Parameters

Once discovered, analyzed, and catalogued, there are certain aspects of a vulnerability that do not change, assuming the initial information is complete and correct. These immutable characteristics will not change over time, nor in different environments. The base metric group captures the access to and impact on the target.



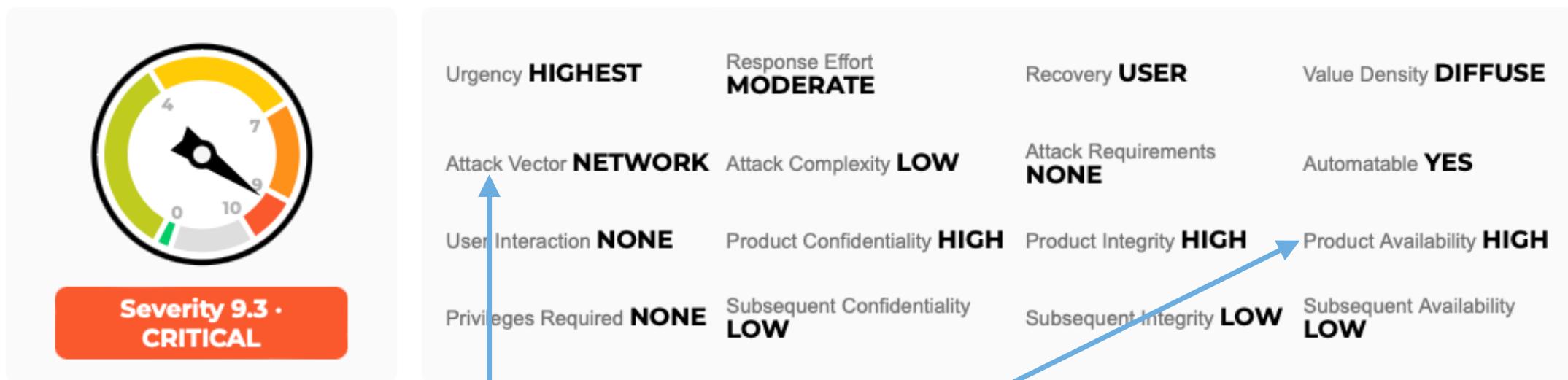
Beispiel:
<https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20181101-ap>

- Exchange Validation Key Remote Code Execution Vulnerability (CVE-2020-0688)
 - ❑ Schwachstelle in der Schlüsselerzeugung (keine unique keys)
 - ❑ Gestohlenes E-Mail Konto erlaubt einschleusen von Code zur Systemkompromittierung
- CVSS v3 Base Score: 8.8 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)

Base Score Metrics		
Exploitability Metrics		Scope (S)*
Attack Vector (AV)*		<input checked="" type="checkbox"/> Unchanged (S:U) <input type="checkbox"/> Changed (S:C)
<input checked="" type="checkbox"/> Network (AV:N) <input type="checkbox"/> Adjacent Network (AV:A) <input type="checkbox"/> Local (AV:L) <input type="checkbox"/> Physical (AV:P)		
Attack Complexity (AC)*		Impact Metrics
<input checked="" type="checkbox"/> Low (AC:L) <input type="checkbox"/> High (AC:H)		Confidentiality Impact (C)*
		<input checked="" type="checkbox"/> None (C:N) <input type="checkbox"/> Low (C:L) <input checked="" type="checkbox"/> High (C:H)
Privileges Required (PR)*		Integrity Impact (I)*
<input checked="" type="checkbox"/> None (PR:N) <input type="checkbox"/> Low (PR:L) <input type="checkbox"/> High (PR:H)		<input checked="" type="checkbox"/> None (I:N) <input type="checkbox"/> Low (I:L) <input checked="" type="checkbox"/> High (I:H)
User Interaction (UI)*		Availability Impact (A)*
<input checked="" type="checkbox"/> None (UI:N) <input type="checkbox"/> Required (UI:R)		<input checked="" type="checkbox"/> None (A:N) <input type="checkbox"/> Low (A:L) <input checked="" type="checkbox"/> High (A:H)

Anwendungsbeispiel - Palo Alto CVE 2024-5910

- Missing Authentication in Palo Alto Networks Expedition leads to Admin Account Takeover
 - <https://security.paloaltonetworks.com/CVE-2024-5910>



CVSSv4 Base Score: 9.3 (CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:L/SI:L/SA:L/AU:Y/R:U/V:D/RE:M/U:Red)

<https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:L/SI:L/SA:L/AU:Y/R:U/V:D/RE:M/U:Red>

1. Grundlegendes zur Angriffsanalyse
 - Notation von Sicherheitsproblemen
 - Angreifermodelle
 - Begriffe und Zusammenhänge
2. Ausgewählte technische Angriffsvarianten
 - Denial of Service (DoS und DDoS)
 - Schadsoftware (Malicious Code - Viren, Würmer, Trojanische Pferde)
 - E-Mail-Security (Spam)
 - Systemnahe Angriffe (Buffer Overflows, Backdoors, Rootkits, ...)
 - Web-basierte Angriffe (XSS, ...)
 - Netzbasierte Angriffe (Sniffing, Portscans, ...)
3. Bewertung von Schwachstellen
 - Common Vulnerability Scoring System (CVSS)
 - Zero Day Exploits

Analyse von „zero-day“ Exploits

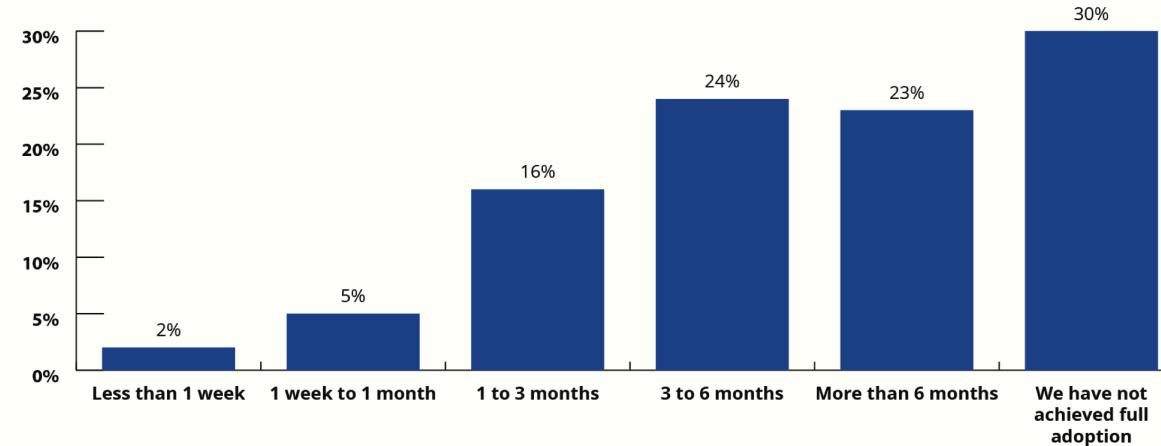
- “Before we knew it - An empirical study of zero-day attacks in the real world”, Bilge/Dumitras, Oktober 2012
http://users.ece.cmu.edu/~tdumitra/public_documents/bilge12_zero_day.pdf
- Wie lange werden Sicherheitslücken ausgenutzt, bevor sie allgemein bekannt (und beseitigt) werden?
 - Untersuchung für 11 Millionen Windows-PCs mit Symantec-Software
 - Dauer schwankt zwischen 19 Tagen und 30 Monaten
 - Durchschnitt liegt bei 312 Tagen (!)
- Google Project Zero
 - 0Day „In the Wild“; Angriffe auf Basis von Zero Day Exploits seit 15.07.14
<https://googleprojectzero.blogspot.com/p/0day.html>
 - Spreadsheet mit „Date discovered“, „Date Patched“ (Patch Available) zwischen 1 und 165 Tagen

Analyse von „zero-day“ Exploits

- Ponemon Institut: State of Endpoint Security Risk -10.2018

<https://cdn2.hubspot.net/hubfs/468115/whitepapers/state-of-endpoint-security-2018.pdf>

Figure 17. How long did it take your organization to achieve full adoption of your EDR?



- Wie wirkt sich die Veröffentlichung einer Sicherheitslücke aus?
 - Anzahl an Malware-Varianten steigt um das bis zu 85.000-fache
 - Anzahl beobachteter Angriffe steigt um das bis zu 100.000-fache

Zusammenfassung

- Angreifermodelle beschreiben Fähigkeiten, Motivation usw.
- Angriffe zielen darauf ab, den individuellen Schutzbedarf (Vertraulichkeit, Integrität, Verfügbarkeit) zu verletzen:
 - Malware-/Rootkit-infizierte Systeme bieten keine Vertraulichkeit mehr
 - Buffer Overflow Exploits zerstören die Integrität von Software
 - DoS-Angriffe stören die Verfügbarkeit
 - ...
- Zu jedem Angriff gibt es mehr oder weniger effektive / kostspielige Gegenmaßnahmen:
 - Ziel ist aber kein Flickenwerk aus einzelnen Maßnahmen, sondern ein von Grund auf sicheres Design (=> Security Engineering).
 - Kenntnis von Angriffsvarianten und -wegen ist Voraussetzung für die Konzeption adäquater Sicherheitsmaßnahmen.

NSA and CISA Share Top Ten Cybersecurity Misconfiguration



- CISA: Cybersecurity & Infrastructure Security Agency - America's Cyber Defense Agency
 - <https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-278a> aus 2023
1. Default configurations of software and applications
 2. Improper separation of user/administrator privilege
 3. Insufficient internal network monitoring
 4. Lack of network segmentation
 5. Poor patch management
 6. Bypass of system access controls
 7. Weak or misconfigured multifactor authentication (MFA) methods
 8. Insufficient access control lists (ACLs) on network shares and services
 9. Poor credential hygiene
 10. Unrestricted code execution

1. Default Configuration of Software and Applications



- Default Credentials
 - Finding Credentials with a simple web search
 - Resetting built-in administrative accounts via predictable forgotten password questions
 - Built-in administrative credentials for web applications and gaining access to the application and its underlying database.
- Default Service Permissions and Configuration Settings
 - Insecure Active Directory Certificate Services
 - Insecure legacy protocols/services
 - Insecure Server Message Block (SMB) service

9. Poor Credential Hygiene



- Easily crackable passwords
 - Guess within a short time using relatively inexpensive computing resources
 - Lack of password length (i.e., shorter than 15 characters) and randomness
 - Coaching hashed passwords with [hashcat](#)
- Cleartext password disclosure
 - Storing passwords in cleartext is a serious security risk
 - Malicious actors search for text files, spreadsheets, documents, and configuration files in hopes of obtaining cleartext passwords.
 - A common tool used for locating cleartext passwords is the open source tool, [Snaffler](#)

- Information Systems Audit and Control Association (ISACA)
- <https://www.isaca.org/resources/news-and-trends/industry-news/2024/the-top-5-cybersecurity-threats-and-how-to-defend-against-them> aus 2024
- Social Engineering
- Third Party Exposure - betrifft Beziehungen zu Externen, z.B. Outsourcing
- Cloud Vulnerabilities
- Ransomware
- Attacks on IoT Devices
- ISACA zeigt auf Verteidigungsstrategien (how to defend) auf ihrer Seite