Verteidigung gegen Ransomware-Cyberangriffe mit Linux und Open Source

Tübix 2025

Prof. Dipl.-Ing. Klaus Knopper <klaus.knopper@hs-kl.de>
Vizepräsident Digitalisierung Hochschule Kaiserslautern
Software-Engineer

Tübix
› Tübinger Linuxtag

Über Klaus Knopper

- **Dipl.-Ing. Elektrotechnik (→ Uni KL),** erster Kontakt zu Softwaretechnik / IT erst während des Studiums
- Mitgründer **Unix-AG an der Uni-KL**, Mitgründer **→ LinuxTag e.V.** Messe + Konferenz 1997 bis 2005
- **Systemsoftware-Entwickler**, **Open Source** Betriebssysteme, **→ Knoppix™ (Live-Linux)**, **barrierearme User Interfaces** (→ ADRIANE für blinde Computernutzer*innen), **Dozent und IT-Berater**,
- seit 2001 an der **→ HS-KL, Professor für Software Engineering**, **Betriebssysteme**, **Agile Software-Entwicklung**, **IT-Sicherheit**, **3D Prototyping**
- Seit 2022 **Vizepräsident für Digitalisierung** an der HS-KL, **Kompetenzzentrum Digitalisierung und Medien**, **SoliD**-Projekt (Souveränes Lernen im Digitalen)

„Aufgrund der laufenden Ermittlungen weist Klaus Knopper darauf hin, dass in diesen Folien ausschließlich allgemeine Sachverhalte zu Cyberangriffen dargestellt werden, die nicht in Bezug zu Ereignissen an der Hochschule Kaiserslautern stehen.“

https://www.hs-kl.de/hochschule/aktuelles/cyberangriff/aktuelle-meldungen-und-hinweise

Cyberangriffe
Prolog

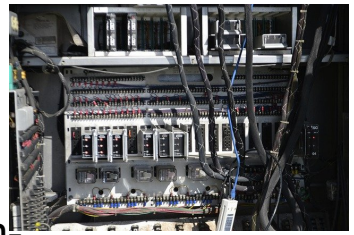
Es ist weniger eine Frage **ob**, sondern **wann** ein Angriffsversuch krimineller Gruppen auf die IT-Infrastruktur ganz oder in Teilen erfolgreich ist. S.a. → Ransomlook.Io

Hierbei werden von den Angreifern nicht nur vernachlässigte Sicherheitseinstellungen, durch → **Phishing** erworbene Zugangscodes, von unachtsamen Nutzern installierte → **Malware**, sondern auch → **aktuelle Schwachstellen** („Zero Day Exploits“) in Systemsoftware, für die es noch gar keine Behebungen gibt, ausgenutzt, um sich zunächst Zugang zu verschaffen und kritische Systeme in „Besitz“ zu nehmen.

Bis dahin sollten im Unternehmen mindestens vorbereitende Maßnahmen und ein Notfallplan (s.a.→ **ISMS**, → BSI-Standards) etabliert sein (s.a. Tipps am Ende).

Cyberangriffe
Ablauf (1)

- Angreifer verschaffen sich (meist zunächst **unprivilegierte**) **Zugänge** zum Netzwerk und einzelnen Rechnern, z.B. durch geleakte bzw. aus erfolgreichen Phishing-Aktionen auch anderer Gruppen erworbene Zugangsdaten, Hintertüren in Softwaresystemen, unbeabsichtigt von Nutzern installierte Trojaner etc.



- → **Privilege Escalation**: Über Schwachstellen privilegierter Software oder erratene / geknackte Admin-Accounts verschaffen sich die Angreifer Privilegien auf Systemebene auf kritischen Systemen, d.h. sie erhalten **Zugriff auf sämtliche Daten und Einstellungen** und können das System-verhalten beeinflussen, **Scanner** zum Aufdecken weiterer Systemschnittstellen sowie weitere **Backdoors** installieren, um den **Zugang auch beim Schließen einer Schwachstelle aufrecht zu erhalte**n.
- Oft kommen hierbei “schlafende” Programme (Daemons) zum Einsatz, die eine Verbindung zu → **Command & Control Servern** aufrecht erhalten, die dann zu einem bestimmten Zeitpunkt aktiv werden, um koordinierte **Aktionen auf den kompromittierten Rechnern zu starten.**

Cyberangriffe
Ablauf (2)

- Sobald privilegierter Zugang zu zentralen Systemen besteht, die „interessante“ (aus Sicht der Angreifer) Dateien enthalten, werden diese **Daten** kopiert und **ausgeleitet**.

Bis zu diesem Zeitpunkt bleiben die Angreifer im Regelfall unbemerkt, sofern keine Intrusion Detection Mechanismen (→ **IDS**) auf den betroffenen Systemen, herstellerseitig oder selbst installiert, existieren.

- Während der Zeit, in der die Angreifer **unentdeckt im Hintergrund arbeiten**, werden ggf. auch **weitere Systeme „gekapert“**, ggf. werden auch bereits kompromittierte Rechner als Teil eines **Botnetz**es verwendet, um weitere Rechner, auch anderer Institutionen, anzugreifen oder dies sogar als Dienstleistung zu verkaufen.

Cyberangriffe
Ablauf (3)

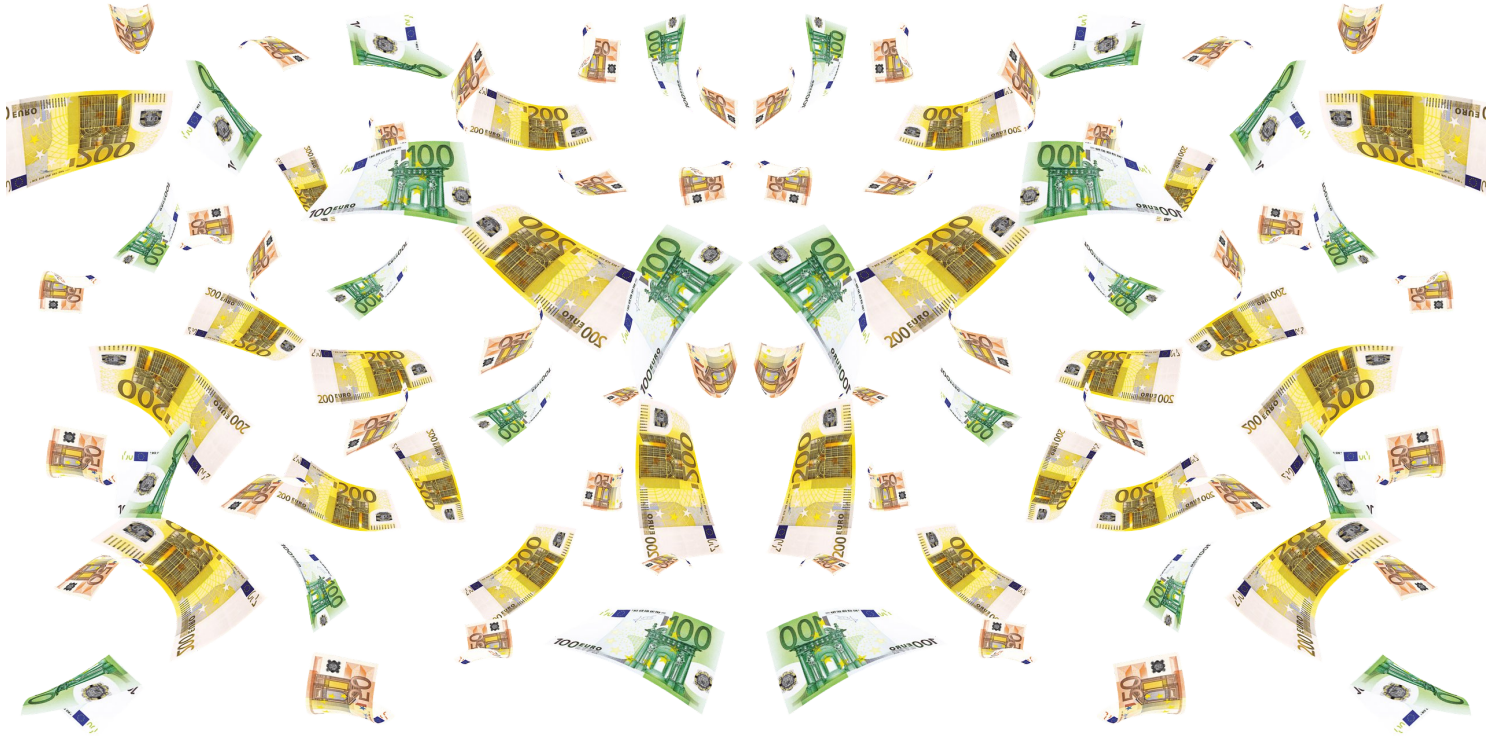
(Vermuteter) **Entdeckung des Angriffs**, oder **nach einer festen Zeitspanne**, kommt die nächste Stufe des Angriffs zum Einsatz, bei der

1) Die **Spuren des Angriffs verwischt** werden (Löschen / Überschreiben von Log-Dateien, Löschen der verwendeten Angriffs-Tools),

2) **maximaler Schaden** durch Verschlüsselung aller Nutzdaten und zumindest große Teile der für den operativen Betrieb notwendigen Programme und Konfigurationen **produziert wird**.

Im Fall von **Ransomware**: Per Kommando wird ein **Verschlüsselungstool** (s. → Folie 21) **gleichzeitig** auf **allen im Netzwerk erreichbaren Rechnern** gestartet. Auch werden auf die eine oder andere Art deutlich sichtbare „Erpressungsbriefe“ gestreut, die eine **anonymisierte Kontaktaufnahme mit den Angreifern einleiten** sollen.

Motivation der Angreifer?



Ziel des Angriffs

(Ransomware)

Gewinnmaxmimierung (s.a. → **double extortion)**:

Forderung einer **Summe X** (oder **Verhandlung / Versteigerung**) in einer [nicht immer] **schwer zurückverfolgbaren Zahlungsweise** unter **Androhung**

- besonders peinliche / sensible / unternehmenskritische **Daten** zu **verkaufen** bzw. zu **veröffentlichen**,
- massiver und nicht wiederherstellbarer **Verlust aller Arbeitsdaten und Backups** (starke Verschlüsselung),
- **Katastrophale Auswirkungen** auf Reputation / Arbeitsfähigkeit / bis hin zum **Totalverlust der wirtschaftlichen Existenz**.

Die Ausprägung der Versprechungen und Drohungen ist hier je nach Gruppierung unterschiedlich (oft **Spezialisierung** der Gruppe auf **bestimmte Arten von Unternehmen**, öffentlicher Dienst, kritische Infrastrukturen, Gesundheitswesen etc.)

Versteigerung im Darknet

EXCLUSIVE AND UNIQUE DATA ON SALE (200 GB TOTAL, **SAMPLES BELOW**):

TIME LEFT: 6 HRS 20 MINS 5 SECS

SUGGESTED PRICE: 25BTC

Sollte man zahlen?

Nein! Am besten überhaupt keinen Kontakt aufnehmen „tot“ stellen. *)

*) Persönliche Meinung des Autors und Empfehlung der Behörden

Warum nicht?

Eine **Garantie** darauf, **Daten wieder zu erhalten** oder **dauerhaft** gegen die Veröffentlichung gestohlener Daten **geschützt zu sein**, gibt es **weder mit, noch ohne Zahlung**. **Mitunter** sind die Angreifer auch selbst **gar nicht in der Lage, die Daten wieder zu entschlüsseln**, und haben i.d.R. auch Mechanismen für einen **Folge-Angriff auf den kompromittierten Systemen hinterlassen** (den sie an andere Gruppen verkaufen).

Selbst wenn scheinbar die Möglichkeit einer „einfachen“ Lösung durch Zahlung einer Summe X bestünde, ist die daraus resultierende **Unterstützung krimineller Gruppen im Regelfall illegal**, aufgrund **gesetzlicher Vorgaben oft auch gar nicht durchführbar** und **fördert das falsche Verhalten.**

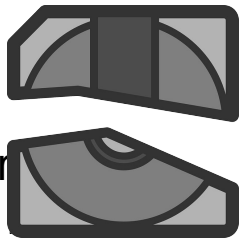
Cyberangriffe
Was ist zu tun?

- gesetzliche Vorgaben (Meldepflicht CERT, Landesdatenschutzbeauftragter, ggf. BSI (bei KRITIS), LKA, Ministerium, ...)
- organisatorische Maßnahmen
- technische Maßnahmen
- abgestimmte Kommunikation nach außen und innen
- sensible Betreuung der betroffenen Mitarbeitenden.

S.a. → Checkliste (12-Punkte) des Bundesamtes für Sicherheit in der Informationstechnik https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Empfehlungen-nach-Angriffszielen/Unternehmen-allgemein/IT-Notfallkarte/TOP-12-Massnahmen/top-12-massnahmen.html

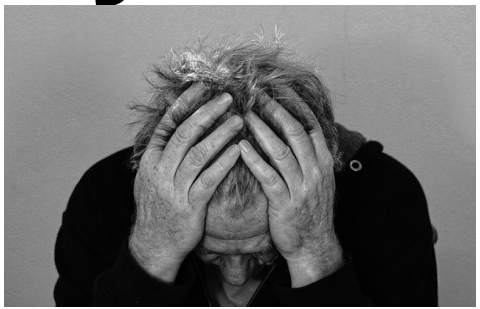
Kommen wir zu den Backups…

- Nach Stillegung und **Neuinstallation aller kompromittierten Rechner freut sich jede*r Admin**, dass er*sie **selbstverständlich tägliche Backups** mit einem **von einer professionellen Firma installierten, "marktführenden Backup-System"** angefertigt hat…
- ...bis sich herausstellt, dass sich **diese Backups auch nicht mehr lesen lassen**. Rein hypothetisches Beispiel: Bei einem ziemlich populären und teuren proprietären Produkt zur **Sicherung von VMs bricht der Restaurierungsvorgang** sofort **ab**, wenn in der Backup-Datei ein **Block mit einem Bit-Fehler** gefunden wird – wenn man Pech hat, ganz am Anfang des Archivs. (Also, fast **immer**.)
- Natürlich **würde** der **Hersteller** der Backup-Software uns doch sicher sofort helfen, den kleinen Fehler zu beheben, und kontaktiert sofort seine **Entwickler**, um eine Lösung zu finden, war schließlich alles sehr teuer, und Backups sind doch schließlich auch für den Worst Case gedacht… – **NICHT**.



Backups

„I’m sorry, this is proprietary information.“*)



*) Die **Frage** eines Kunden nach einer **technischen Spezifikation** des **Backup-Dateiformats, um selbst Skripte zur Datenrettung programmieren zu können** ist wohl irgendwie **naiv**?
Viel Spaß beim Reverse Engineering...

“The Backup is the Problem”

- **Konzeptionell** (seitens Hersteller) kann der **Backup-Server** lesenden **und direkt schreibenden Zugriff** auf die VM-Management-Struktur haben (z.b. über **Network Block-Devices**), um die täglichen blockbasierten inkrementellen Backups und Restauriervorgänge bequem und **zentral gesteuert anstoßen** zu können.

- **Folgen dieses Konzepts:** Bei **Kompromittierung** des **Authentifikationssystems (“Active Directory”)** und/oder **des Backup-Servers selbst** haben die **Angreifer** damit auch leichtes Spiel beim **Schreibzugriff auf das VM-Cluster** – obwohl **Linux-Server** (wie bei VMWare und alle anderen Virtualisierungslösungen) i.d.R. **gar nicht das primäre Ziel von Cyberangriffen** und (in diesem Fall nur noch gefühlt) immun sind.

- **Lessons Learned:** Es geht eben doch nichts über **nicht-löschbare** Backups (z.B. auf Tape), z.B. als **second level Backup**. Und ein **“Offline”-Backup-Konzept** (am besten Open Source mit offengelegten Dateiformat).

Und der VM-Server selbst?

- ...

Proprietäre Dateisysteme

(Vorgriff auf die “technischen Folien”)

Während das unter Windows gebräuchlichste NTFS mit Live-Systemen zur Datenrettung hervorragend mit dem Open Source `fuse`-Dateisystem **`ntfs-3g`** unterstützt wird, sieht es bei einigen proprietären schlecht aus mit dem Support und der technischen Dokumentation, hierzu gehören:

ReFS („Resilient Filesystem“ – ist nicht resilient) → Microsoft
VMFS (Virtual Machine File System) → VMWare ESXi

Während ReFS sich (fast) nur unter Windows einbinden lässt, gibt es für VMFS experimentelle Linux → fuse-Module, die das Dateisystem (ggf. **mit eigenen Patches**) read-only einbinden. Das genügt für die Datenrettung, ist aber aufwändig.

Lessons [to be] learned (Part 1)

Strategisch: Im Rahmen des **ISMS**, **Hinterfragen** und **realistische Risikobewertung** auch aller „langjährig erprobten, Industrie-Standard Lösungen, die alle anderen auch haben“

~~Zentrale Domänenverwaltung / Profile~~ ☠

~~VPN mit Allgemeinzugang~~ ☠

~~“Eeeasy to use” ADMIN GUI~~ ☠

(Entgegen dem Werbeversprechen, dass *keinerlei* fortgeschrittene IT-Kenntnisse zur Administration notwendig seien, was dazu führt, dass bei Katastrophen auch niemand weiß, welche Daten wo und wie gespeichert sind)

~~Proprietäre Virtualisierungslösung~~ ☠

~~Proprietäre Backuplösung~~ ☠

~~“Netzlaufwerke verbinden“~~ ☠

…

Lessons [to be] learned (Part 2)

im weiteren Verlauf (nach Angriffsbewältigung)

- **Neue IT-Infrastruktur** mit **Fokus auf Resilienz**, Funktionalität und Nachhaltigkeit **planen** (auch hier sollten Experten hinzugezogen werden), hierbei **nicht mehr konkrete Produkte** („alles wie bisher“), sondern **benötigte Funktionalitäten** und **Risiken** betrachten. *)
- **Heterogenität** und **Abkopplung** von IT-Diensten („Inseln“) fördern **Resilienz**, **zentrale Administration**smechanismen und **“einfache ADMIN-GUIs” mit nicht dokumentierten Zugriffs-Mechanismen auf zentrale Infrastruktur vermeiden**.
- Weiterführung, **Anpassung**, bzw. Etablierung des → ISMS. ☜

*) Problem der Akzeptanz ist nach einem Cyberangriff-Erlebnis nicht mehr prioritär, „Macht der Gewohnheit“ vs. Risikoabwägung: intensiver Austausch mit allen Beteiligten und IT-Sicherheitsbeauftragte sowie Hochschulleitung notwendig.

Empfehlungen aus persönlicher Erfahrung

- **Zentrale/Kritische Infrastruktur-Komponenten** müssen langfristig **digital souverän (Open Source)** sein und **vollständig technisch verstanden werden** (von den Administratoren / in-house Experten, auch wenn man externen Support regelmäßig in Anspruch nimmt)
  Das sind insbesondere
  - **zentrale Authentifikations- und Zugriffsmechanismen, Benutzerverwaltung,**
  - **zentrale Datenhaltung,**
  - **zentrales Backup (first level, second level, ...)**

da diese das bevorzugte Ziel der Angreifer sind und Fehlerquellen bei proprietären Systemen nicht selbst repariert/eliminiert werden können
→ **Herstellerunabhängigkeit notwendig**!

- BSI-Grundschutz muss umgesetzt werden, v.a.
  - ✓ BSI-Standard 200-1: Managementsysteme für Informationssicherheit (ISMS)
  - ✓ BSI-Standard 200-2: IT-Grundschutz-Methodik
  - ✓ BSI-Standard 200-3: Risikomanagement
  - ✓ BSI-Standard 200-4: Business Continuity Management

Lessons [to be] learned (Part 3)

weitere “kleine”, aber effektive Maßnahmen

- **Mehrfaktor-Authentifizierung** für (möglichst) alle Dienste

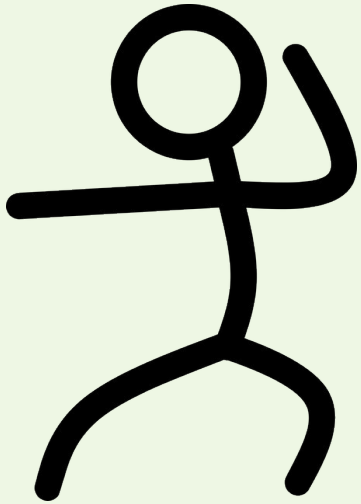
- Einführung verpflichtender **digitaler Signatur** bei
- E-Mails mit Dienstanweisungen
- Dokumenten-Archivierung

- Signatur**+Verschlüsselung** für
- E-Mails mit vertraulichem Inhalt (Notenlisten, Bewertungen, Protokolle)
- Archivierung vertraulicher Dokumente

- **Hinterlegen** im Safe (offline) von
- Zugangs-Schlüsseln für Notfälle
- komplementäre Instruktionen und Diagramme (Teil des ISMS)

Kung Fu

Open Source Werkzeuge und nützes Gedöns



- Was passiert beim Cyberangriff Phase 3 mit den Daten?
- Linux Live Systeme für “Offline”-Analysen
- Disk space - the final frontier
- Post mortem: Recovery- und Forensik-Tools für Fortgeschrittene
- Technische Schutzmaßnahmen: Wie kamen die eigentlich rein, und wie verhindern wir, dass es wieder passiert?

Angriff: Verschlüssler

(Beispiel/Analyse aus der Praxis)

Verschlüsselung der Nutzer-Dateien durch die Angreifer mit **starker Kryptographie**

- **Parallel**,
- auf **allen lokalen und über das Netzwerk erreichbaren Datenträgern**,
- mit **zufallsgeneriertem symmetrischem Schlüssel** (Geschwindigkeit),
- Blockweise **Springen in Datei** nach Zufallsprinzip, ggf. Blockindex am Ende der Datei anfügen.

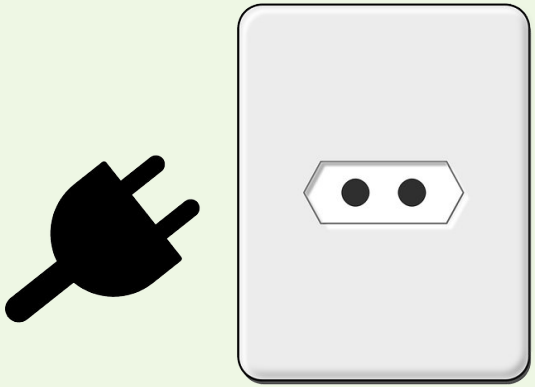
- Symmetrischer Schlüssel wird ggf., mit asymmetrischem Schlüssel verschlüsselt, an verschlüsselte Dateien angehängt (sofern Entschlüsselung überhaupt möglich sein soll)
- der komplementäre Schlüssel zum **Entschlüsseln** des symmetrischen Schlüssels befindet sich **zu keinem Zeitpunkt** auf dem betroffenen Rechner.

→ **Maximale Geschwindigkeit und Zerstörungskraft.**

Datenzerstörung stoppen!

Daher die Empfehlung, nicht lange zuzuschauen, sondern den **Stecker ziehen**, sobald der Verschlüsselungsvorgang bemerkt wird, um den **aktuellen Zustand** zur späteren Analyse/Wiederherstellung **einzufrieren**.

Alte installierte OS nicht mehr starten (**nie wieder**).



Linux Live Systeme zur „Offline“ Analyse

Z.B.

- **Kali Linux**
- **SystemRescue CD**
- **Knoppix** geht auch (ist aber auf 64bit-Systemen langsamer).

Das gewählte System darf **Datenträger auf keinen Fall schreibbar** einbinden, sondern **unbedingt read-only** und **ohne automatischen „Konsistenzcheck/Reparatur“**, sonst wird das Dateisystem (wie beim „Herunterfahren“) verändert!

Der erste Schritt der Datenrettung sollte also darin bestehen, die betroffenen Partitionen als Images auf frische Datenträger umzukopieren → `dd-rescue`

Disk Space – the final frontier

Zur **Sicherung** zwecks **Analyse** und **gerettete Daten** muss, da die Original-Dateisysteme für die Forensik **unbedingt unverändert** bleiben sollen, unbedingt **ausreichend große Storage beschafft werden**!

Kung Fu

Werkzeuge und nützes Gedöns

Linux Live-Systeme zum Booten der abgeschalteten, kompromittierten Systeme im **Read-Only-Modus (!)**, ggf. zusätzliche Tools nachzuinstallieren (USB-Stick Live-System mit Persistenz).

→ Ein „sauberer Shutdown“ oder „**Konsistenzcheck**“ der **Dateisysteme** muss zur Analyse und Recovery **unbedingt vermieden werden!**

```
mount -r /dev/sda1 /mnt
```

Dateisysteme

Das **Dateisystem** bildet die **Hardware-Ebene** der Bits und Bytes in **logische Strukturen (Dateien, Ordner, Verweise usw.)** ab.

Verschiedene Betriebssysteme verwenden ihr jeweils favorisiertes Dateisystem, z.B. FAT32/VFAT, EXFAT, NTFS, ext4, btrfs, …

Linux unterstützt **die meisten davon** (einige mangels öffentlicher Spezifikation nur teilweise durch Reverse Engineering). Das zuvor gezeigte `mount`-Kommando bindet den Datenträger bzw. eine Partition darauf als Dateisystem ein.

Manchmal sind weitere **mount-Parameter** erforderlich, z.B. `-o offset=$((512*startsektor))` oder `-o noload`.

Dateisysteme finden

Nach einem erfolgreichen Angriff sind v.a. auf virtuellen Disks (Images) oft **keine Partitionen mehr auffindbar**.

Das Open Source Tool **`testdisk`** (das wir später noch zu anderen Zwecken einsetzen) kann Dateisystem-Start und -Ende aufgrund dateisystemtypischer **Signaturen** finden.

Hier ist manchmal auch etwas kreative Suche und **manuelle Angabe von Sektoren** notwendig, um die richtigen **Partitionsgrenzen zu finden** oder wieder herzustellen.

Es sollte auch hier, sofern Änderungen geschrieben werden, immer mit einer **Kopie des Image** gearbeitet werden!

NTFS

Seit der Veröffentlichung von → **ntfs-3g** wird das Windows-Dateisystem **NTFS** sehr gut von Linux unterstützt – lesend wie schreibend.

NTFS hat einige für die Reparatur nützliche „Eigenheiten“ – wir wir später sehen werden, und einige Unix-Dateisystem-Features „nachempfunden“.

NTFS-3G

zwei empfehlenswerte Erweiterungen (Plugin-Libraries) für zwei nicht-ganz-so-praktische NTFS-Eigenarten

ntfs-3g-dedup: „Deduplikation“ – sorgt für das richtige „Zusammensetzen“ von Blöcken aus dem „ChunkStore“ unter „System Volume Information/Dedup“, wenn das Deduplikations-Feature bei NTFS eingeschaltet ist (soll „gemeinsame Inhalte“ von Dateien nur einmal speichern – also ähnlich wie bei Hardlinks unter Linux).
Fehlt das Plugin, dann werden Dateien bei eingeschalteter Deduplikation unvollständig kopiert!

→ **ntfs-3g-system-compression**: (De-)komprimiert Dateien, wenn das „kompressions-Feature“ eingeschaltet ist.

Angriff: Spuren verwischt, Dateien gelöscht...

Die Angreifer **löschen** im letzten Schritt auch ihre **eigenen Tools** - bis auf „kleine Geschenke“ – Baukasten-**Trojaner** verankert im kompromittierten Betriebssystem, die viele verschiedene Gruppen nutzen und die keinen Aufschluss über die individuelle Gruppe erlauben - auch in noch funktionsfähiger Systemsoftware für einen evtl. erneuten Zugriff von außen im Falle einer System-Reparatur. (Daher auch die Empfehlung, das kompromittierte OS nicht mehr zu starten).

Außerdem werden **Systemprotokolle** (Windows-Eventlogs: `Windows\System32\winevt\Logs\*.evtx , Unix: /var/log/*)` verkürzt, auf **0 Byte Dateigröße** gesetzt oder mit **0-Bytes überschrieben**, um Hinweise auf den Infektionsweg zu vernichten.

QUIZ

Was bewirkt das Kommando
`rm -f datei.pdf` *)
?

A. Die Datei “datei.pdf” und ihr Inhalt wird **gelöscht.**

B. Die Datei “datei.pdf” und ihr Inhalt ist **nicht mehr auffindbar.**

C. Die in “datei.pdf” enthaltenen **Bits gehen verloren.**

D. **Nichts davon.**

*) Windows: **`del datei.pdf`**

Kung Fu

Werkzeuge und nützes Gedöns

Beim „**Löschen**“ von Dateien entfernen die meisten Dateisysteme den **Dateieintrag zunächst aus den Metadaten / Tree / FAT** und markieren den ursprünglich belegten Platz als „freigegeben“. **Transaktionsbasierte** Dateisysteme bieten zudem die Möglichkeit, eine **Anzahl der letzten Aktionen rückgängig** zu machen.

testdisk - Findet Partitionen und Dateisysteme, kann bei FAT32 und NTFS gelöschte Dateien, die noch in den Metadaten vermerkt sind, kopieren.

ntfsundelete - Findet die letzten „Lösch“-Transaktionen und kann diese rückgängig machen bzs. die gelöschten Dateien kopiere.

hexedit - Universal on-disk-editor mit Suchfunktion

QUIZ

Was bewirkt das Kommando
`echo "Kaputt." > datei.jpg`
?

A. Der Inhalt der Datei “datei.jpg” wird **überschrieben.**

B. Die Bilddaten der Datei “datei.jpg” werden **unbrauchbar** gemacht.

C. Die in “datei.jpg” enthaltenen **Bits gehen verloren.**

D. **Nichts davon.**

Kung Fu

Werkzeuge und nützes Gedöns

Beim „Überschreiben“ von Dateien positionieren die meisten Dateisysteme (z.B. NTFS!) aus Effizienz- und Konsistenzgründen die „neuen“ Daten an einer anderen Stelle als die alten, und markieren den ursprünglich belegten Platz als „freigegeben“.

→ Der **alte Dateiinhalt verbleibt auf dem Datenträger**, bis er (zufällig, oder wenn der Rest der **Disk voll** ist) doch **überschrieben wird**.

`photorec` `- Finden und Kopieren von Dateiinhalten über Kennungs-Bits und Metadaten.`

Kung Fu

Werkzeuge und nützes Gedöns

Mit dem (Linux-)Kommando

```
find /mnt/partition -type f -printf '%T+ %p\n' | \
 sort | \
 tee timestamps.txt
```

lässt sich eine nach Modifikationsdatum sortierte Dateiliste erzeugen.

```
...
2023-06-08+06:22:59.4428442000 ./Users/guest/NTUSER.DAT
...
2023-06-08+12:04:44.0000000000 ./$Recycle.Bin/S-1-5-21-1413309913-2581364771-2488145185-47081/$R9HEUXM/armageddon.exe
...
```

Die **Zeitstempel** können Hinweis auf den **Zeitpunkt von System-Modifikationen**, Logins (modifizierte User-Directories) und Verschlüsselungs-Start und -Ende geben, oder auch verschobene Dateien finden.

Angriff: Einmal erlangte Kontrolle über Server und Clients verstetigt durch gut versteckt installierte Backdoors

- Ein von den **Angreifern gut versteckt installiertes Programm** (bei Windows meist ein Powershell-Skript) baut periodisch immer wieder eine Verbindung zu einem Command & Control Server auf, von innen nach außen, zur Aktivierung der nächsten Stufe, und wartet auf Instruktionen.
- Ggf. wird durch **Eventsteuerung** Schadsoftware **automatisch reaktiviert** oder **neu installiert**, sobald ein neuer Datenträger, lokal oder übers Netz, angeschlossen wird (v.a. bei Windows).
- Die **Angreifer deaktivieren Protokollierungs-Mechanismen** und Malware-Erkennung,
- und **hinterlassen**, auch nach Phase 3, „**kleine (trojanische) Geschenke**“ in Mailboxen, Dokumenten, ausführbaren Dateien.

Wie findet und eliminiert man alle Schadsoftware auf dem kompromittierten System?

- ...kaum möglich, jedenfalls nicht mehr mit / auf dem kompromittierten System, wenn proprietäre Betriebssysteme im Spiel!

- Eher: **Malware-Scanner** von **Live-System** aus auf Datenträgern **suchen lassen**, und nur sicher „saubere“ Daten auf neue Storage archivieren, danach alles löschen (zumindest, wenn die alten Datenträger wiederverwendet werden sollen). (`apt install clamav`, `clamdscan` ...)

- Schwierig: „mehrfach verpackte“ Daten oder **proprietäre Container** (wie PST-Mailarchive) sind nicht einfach zu scannen (`apt install pst-utils`, `readpst` ...).

Angriff: Daten-Ausleitung

- **Geringe Datenrate** - fällt kaum auf im Vergleich mit den täglich anfallenden Daten.
- **Verbindung wird von Malware von innen nach außen aufgebaut** (umgeht Firewall-Einschränkungen) über zufälligen Port
- **Datenstrom** i.d.R. verschlüsselt und **unauffällig**.

Angriff: Zugang und Schwachstellen-Scan

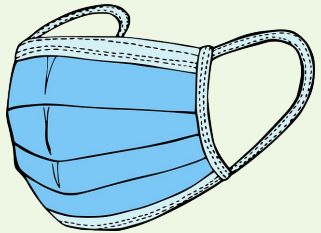
Zugriff auf das Server-Netzwerk zunächst mit unprivilegiertem Account (z.B. durch → Phishing-Listen in Foren erworben, VPN-Zugänge, Malware auf User-Hardware, die Zugang von außen erlaubt), dann Suche nach Angriffspunkten, Default-Admin-Zugängen, bekannten Backdoors in Routern/Gateways etc., z.B.

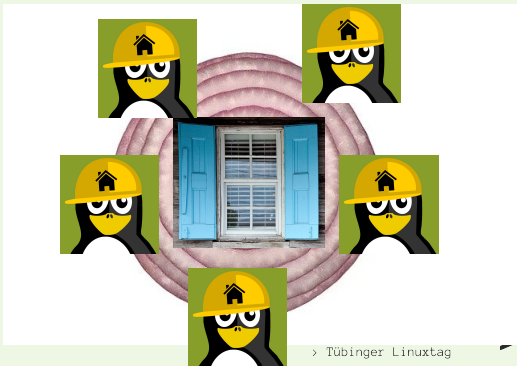
```
sudo nmap -sV --script=vuln ip-range
```

Man sollte dies regelmäßig selbst durchführen und Schwachstellen schließen, bevor Angreifer es tun.
Mit → openvas/greenbone ist der Output besser lesbar.

Technische (Open Source) Vorsorge-Maßnahmen

- Regelmäßige **Log-Analyse** (ggf. Push-Nachrichten)
- Intrusion Detection System (**IDS**) und/oder Intrusion Prevention System (**IPS**) v.a. auf Servern
- Firewall/Monitoring nicht nur **von außen nach innen**, sondern auch **von innen nach außen** (hoher Konfigurationsaufwand!)
- **Offline-Backup** bzw. **nicht löschbares Backup** (und Verschlüselungs-Schlüssel extern aufbewahren)
- **Mehrfaktor-Authentifizierung** Prio1 **Admins**, Prio2 **User**
- „Single Point of Failure“ zentrale Administration überdenken (Worst Case?)
- Mehr **Open Source** und **lokale Cloudlösungen** in kritischen Bereichen (+gut **dokumentierte Datenformate**/Container)
- Regelmäßige Lektüre **Security-Newsticker**
- **Proprietäre Altsysteme**, die sich (noch) nicht migrieren lassen,durch **Zwiebelsystem schützen** (**Linux-Proxies vorschalten**)
- →**ISMS** **mit Notfallplan** etablieren





(N)IDS

→ SNORT:

- `iptables`-basiert (unter Linux), daher sehr resourcenschonend, gibt's aber auch für Windows (**plattformunabhängig**).
- Basisregelwerk erkennt **außergewöhnliche Netzwerk-Aktivitäten / Scans**
- **Automatischen Mailversand** an Admins bei **Erkennung von Bedrohungen konfigurieren!**

→ Tiger:

- Basisregelwerk erkennt **außergewöhnliche Dateisystem-Aktivitäten** / neu installierte oder modifizierte Programme mit s-Flag, merkwürdige Dateinamen oder Verzeichnisse
- Kombinierbar mit anderen Filesystem Monitoring-Systemen wie →Tripwire

Ein einfaches IPS für SSH mit iptables

```
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent \
         --update --seconds 20 --hitcount 6 --name SSH --rsource -j DROP
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent \
         --set --name SSH --rsource -j ACCEPT

ip6tables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent \
          --update --seconds 20 --hitcount 6 --name SSH --rsource -j DROP
ip6tables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent \
          --set --name SSH --rsource -j ACCEPT



# Limitiert auf 6 SSH-Verbindungsversuche, alle 20 Sekunden,
# von der gleichen IP-Adresse → Bremst Brite Force Passwort-Rate-Attacken
# wirksam aus.
```

Fertiges IPS-Produkt für Linux: → fail2ban

Bedeutet nicht „Fehler beim Aussperren“ sondern
„**Aussperren bei Fehlern (Einbruchsversuchen)**“ ;-)

```
# Installation
sudo apt install fail2ban

# Konfiguration (eigene Regeln)
vim -o /etc/fail2ban/.../*.local
/etc/init.d/fail2ban reload
```

Standard-Regeln zur Absicherung von `apache2`, `lighttpd`, `sshd`, `vsftpd`, `qmail`, `postfix`.

Kann man sich drauf verlassen, dass Cyberkriminelle nur Windows™ angreifen?

Auch wenn es in der Praxis so aussieht, dass **bevorzugt proprietäre Systeme** angegriffen werden, so können natürlich auch Linux-Rechner Konfigurationsfehler oder Schwachstellen aufweisen, die regelmäßig durch **Sicherheitsüberprüfungen** gefunden und durch **Security-Updates** behoben werden müssen.
→ **Empfehlung Kurse und LPI-Zertifikat „Security Essentials“**

Bei **Open Source Systemen** wie Linux sind aber die **Möglichkeiten, Fehler selbst zu reparieren** und den **Aufbau des Systems zu verstehen**, **Funktionen** auf den **notwendigen Zweck zu reduzieren** und **Sicherheitsmechanismen** zu **etablieren**, am besten/einfachsten **umzusetzen**.

Mehr **Heterogenität der Betriebssysteme** im Netzwerk statt „Monokultur“ kann zudem die **Infektionskette unterbrechen**.

Prof. Dipl.-Ing. Klaus Knopper

<klaus.knopper@hs-kl.de>