



DIGITAL BUSINESS UNIVERSITY
OF APPLIED SCIENCES

CYBER- & IT-SECURITY (M.SC.)

NETWORK SECURITY

WINTERSEMESTER 2024/25

Forensische Analyse von
Netzwerkangriffen: Identifikation
typischer Angriffsmuster mit
Wireshark

Studienarbeit

Eingereicht von:

Elias HÄUSSLER

Matrikelnummer:

200094

Dozent:

Prof. Dr. Aymen GATRI

22. Februar 2025

Inhaltsverzeichnis

Abkürzungsverzeichnis	iii
1 Einleitung	1
1.1 Problemstellung	1
1.2 Inhalte der Arbeit	1
2 Netzwerke & Netzwerkforensik	2
2.1 Ziele der Netzwerkforensik	2
2.2 Anwendungsfälle in der Praxis	2
2.3 Signatur- & anomaliebasierte Angriffserkennung	2
2.4 Anomalien in Netzwerken	3
2.5 Typische Angriffsmuster im Netzwerkverkehr	3
2.5.1 Netzwerk-Scanning	3
2.5.2 Spoofing-Angriffe	4
2.5.3 Brute-Force-Angriffe	4
2.5.4 Denial of Service (DoS)-Angriffe	4
2.5.5 Malware-Angriffe	4
3 Forensische Analyse mit Wireshark	5
3.1 Wireshark als Analysewerkzeug	5
3.1.1 Betriebsmodus bei Live-Paketerfassung	6
3.1.2 Nutzung aufgezeichneter pcap-Daten	6
3.1.3 Datenverarbeitung durch Dekodierung & Rekonstruktion	7
3.2 Filterung von Netzwerkdaten	7
3.2.1 Capture-Filter	7
3.2.2 Display-Filter	7
3.3 Identifikation von Angriffsmustern anhand von pcap-Daten	8
3.3.1 Port-Scanning	8
3.3.2 Ping-Sweep	9
3.3.3 ARP-Spoofing	10
3.3.4 DDoS-Angriffe	10
3.3.5 Brute-Force-Angriffe	13
3.4 Grenzen der Angriffserkennung & alternative Werkzeuge	13
3.4.1 Limitierung auf (passive) Analyse	13
3.4.2 Begrenzte Skalierbarkeit	14
3.4.3 Einschränkungen durch verschlüsselten Datenverkehr	14
3.5 Wireshark im Vergleich mit anderen Forensik-Tools	14
3.5.1 Schwachstellenbewertung	15
3.5.2 Netzwerkscanner	15

3.5.3	Netzwerküberwachung	15
3.5.4	Intrusion Detection System (IDS)	15
4	Strategischer Nutzen der Netzwerkforensik	16
4.1	Beweissicherung & Strafverfolgung	16
4.2	Schutz kritischer Systeme & Geschäftsmodelle	16
4.3	Unterstützung bei Compliance-Anforderungen	16
5	Zusammenfassung	17
5.1	Ergebnisse	17
5.2	Ausblick	17
	Literaturverzeichnis	19

Abkürzungsverzeichnis

APT	Advanced Persistent Threat
ARP	Address Resolution Protocol
CNN	Convolutional Neural Network
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
DPI	Deep Packet Inspection
HIPAA	Health Insurance Portability and Accountability Act
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
ISO	International Organization for Standardization
KI	Künstliche Intelligenz
KNN	k-Nearest-Neighbor
MAC	Media Access Control
MITM	Man-in-the-Middle
ML	Machine Learning
NIDS	Network Intrusion Detection System
Nmap	Network Mapper
OS	Operating System
pcap	packet capture
pcapng	packet capture next generation
QoS	Quality of Service
SSH	Secure Shell
TCP	Transmission Control Protocol

1 Einleitung

Da nahezu jedes öffentliche System an ein Netzwerk angeschlossen ist, erscheint es nur logisch, dass Angreifer vermehrt Netzwerkverkehr manipulieren, um großflächige Angriffe auf Infrastrukturen und kritische Systeme durchzuführen. Bekannte Angriffsformen wie [DDoS](#)- und Malware-Angriffe nehmen immer mehr zu und stellen mittlerweile eine erhebliche Bedrohung dar. Viele dieser Angriffe folgen einem ähnlichen Muster und sind daher gut identifizierbar. Um im Angriffsfall schnell reagieren zu können und sich zugleich im Rahmen der Prävention vor zukünftigen Angriffen zu wappnen, hat sich die Netzwerkforensik im Bereich der IT-Sicherheit etabliert. In diesem Kontext sticht *Wireshark* als Paketanalyse-Tool besonders hervor, da es durch seine weite Verbreitung und den umfangreichen Funktionsumfang ein bedeutendes Instrument für die forensische Analyse von Angriffen darstellt.

1.1 Problemstellung

Wireshark ist in der Lage, eine Vielzahl an Netzwerkprotokollen zu interpretieren und sowohl inhaltlich als auch statistisch auszuwerten. In vielen Bereichen der Netzwerkforensik ist es daher unverzichtbar. Allerdings ist der Anwendungsbereich von Wireshark auch stark limitiert. In Bezug auf seine Rolle im Bereich der Netzwerkforensik werden in dieser Arbeit daher folgende Fragestellungen erörtert:

- In welchen Fällen können forensische Analysen sinnvoll eingesetzt werden und welche Vorteile bietet ein Tool wie Wireshark?
- Welche Angriffsmuster können durch eine forensische Analyse mit Wireshark zuverlässig identifiziert werden?
- Um welche zusätzlichen Maßnahmen aus dem Bereich der Netzwerkforensik können Paketanalysen effektiv ergänzt werden?

1.2 Inhalte der Arbeit

Ein grundlegender Einstieg in Netzwerke und speziell in den Bereich der Netzwerkforensik wird zunächst in [Kapitel 2](#) gegeben. Dort werden auch typische Anomalien und Angriffsmuster vorgestellt, die im nachfolgenden [Kapitel 3](#) anhand einer forensischen Analyse mit Wireshark näher untersucht werden. Ebenso werden die Grenzen von Wireshark und der direkte Vergleich mit anderen Forensik-Tools beleuchtet. Anschließend geht [Kapitel 4](#) auf den konkreten strategischen Nutzen und die Vorteile der Netzwerkforensik ein. Die Arbeit schließt in [Kapitel 5](#) mit einer Zusammenfassung der Ergebnisse und beleuchtet zukünftige Trends im Bereich der Netzwerkforensik.

2 Netzwerke & Netzwerkforensik

Mit der wachsenden Zahl vernetzter Geräte steigt auch die Gefahr von Netzwerkangriffen. Da diese zunehmend komplexer werden und immer schwerer zu erkennen sind, hat sich im Bereich der IT-Sicherheit die *Netzwerkforensik* entwickelt.

2.1 Ziele der Netzwerkforensik

Die Netzwerkforensik dient der Überwachung und Analyse des ein- und ausgehenden Netzwerkverkehrs, um Anomalien zu erkennen, die auf Sicherheitsvorfälle oder Angriffe hindeuten. Durch die systematische Erfassung, Protokollierung und Untersuchung von Netzwerkereignissen soll die Quelle eines Angriffs identifiziert werden. Darüber hinaus ermöglichen forensische Methoden die Klassifizierung des Angriffs sowie die Rückverfolgung der Angreifer, um weiterführende Sicherheitsmaßnahmen zu ergreifen (vgl. Joshi & Pilli, 2016, S. 7–8).

2.2 Anwendungsfälle in der Praxis

Netzwerkforensik findet überall dort Anwendung, wo Netzwerkverkehr analysiert werden kann. Neben der Erkennung und Untersuchung von Cyberangriffen trägt Netzwerkforensik dazu bei, Angreifer zu identifizieren, selbst wenn sie nach einem (möglicherweise erfolglosen) Angriff versuchen, ihre Spuren zu verwischen. Darüber hinaus erhöht die Möglichkeit einer forensischen Analyse den Aufwand für Angreifer erheblich, was Angriffe kostspieliger und somit unattraktiver macht (Joshi & Pilli, 2016).

Ein weiterer wichtiger Anwendungsbereich ist die Unterstützung von Unternehmen bei der Einhaltung regulatorischer Vorgaben, etwa der *International Organization for Standardization (ISO)* 27001. Durch die Anfertigung von Audit-Daten und detaillierten Protokollen können Unternehmen Nachweise über ihre Sicherheitsmaßnahmen erbringen. Zudem ermöglicht Netzwerkforensik die Dokumentation von Beweisen für weitergehende Untersuchungen, um Angriffsmethoden besser zu verstehen, eingesetzte Tools zu identifizieren und neue Angriffstechniken zu entdecken (Joshi & Pilli, 2016).

Schließlich trägt Netzwerkforensik durch umfassende Verhaltens- und Schwachstellenanalysen zur Verbesserung bestehender Sicherheitslösungen bei. Dadurch können Netzwerkschutzmechanismen gezielt gehärtet und widerstandsfähiger gegenüber Zero-Day- und hybriden Angriffen gemacht werden (Joshi & Pilli, 2016).

2.3 Signatur- & anomaliebasierte Angriffserkennung

Diese Arbeit fokussiert sich auf die Überwachung des Netzwerkverkehrs zur frühzeitigen Erkennung von Anomalien und besserer Analyse von Sicherheitsvorfällen. Dazu werden typischerweise Netzwerksdaten genutzt, die von Firewalls, *Intrusion Detection Systems*

(*IDS*s), Routern oder Switches erfasst werden (Joshi & Pilli, 2016). Grundsätzlich unterscheidet man dabei zwischen signatur- und anomaliebasiertener Erkennung.

Bei der signaturbasierten Methode wird der Netzwerkverkehr mit bekannten Angriffsmustern abgeglichen. Dies ermöglicht eine zuverlässige Identifikation bekannter Angriffe, während neue oder unbekannte Bedrohungen schwerer zu erkennen sind. Die anomaliebasierte Erkennung hingegen analysiert Abweichungen vom normalen Netzwerkverhalten, das zuvor in einem Profil definiert wurde. Dabei kommen häufig *Machine Learning (ML)*-Technologien zum Einsatz (vgl. Kim et al., 2018, S. 5–7).

2.4 Anomalien in Netzwerken

Anomalien sind ungewöhnliche Muster, die vom normalen Verhalten abweichen. Netzwerkanomalien lassen sich grob in vier Kategorien einteilen: Betriebsanomalien, Missbrauchsanomalien, plötzliche Lastspitzen (*Flash Crowds*) und sicherheitskritische Anomalien. Letzteren kommt besondere Bedeutung zu, da ihre frühzeitige Erkennung essenziell für einen sicheren Netzwerkbetrieb ist (vgl. Hoque et al., 2014, S. 309).

2.5 Typische Angriffsmuster im Netzwerkverkehr

Die Vielfalt an Netzwerkangriffen ist enorm, und Angreifer entwickeln ständig neue Methoden, um Schwachstellen auszunutzen. Viele sicherheitskritische Anomalien lassen sich jedoch typischen Angriffsmustern zuordnen, die nachfolgend beschrieben werden.

2.5.1 Netzwerk-Scanning

Um spätere Angriffe gezielt vorzubereiten, führen Angreifer häufig Netzwerkscans durch, ohne dabei aktiv Schaden anzurichten. Dadurch erhalten sie Einblick in die Struktur des Netzwerks, darunter dessen Topologie, offene Ports und verwendete *Internet Protocol (IP)*-Adressen. Auf diese Weise lassen sich bereits vor einem eigentlichen Angriff potenzielle Schwachstellen identifizieren und die bestehende Sicherheitsinfrastruktur analysieren (Nainar & Panda, 2022).

Häufige Methoden des Netzwerk-Scannings sind beispielsweise **Ping-Sweeps**, bei denen durch *Address Resolution Protocol (ARP)*-Anfragen im lokalen Netzwerk sowie durch *Internet Control Message Protocol (ICMP)*-Echos aktive IP-Adressen ermittelt werden können. Darüber hinaus werden häufig **Port-Scans** durchgeführt, um offene Ports zu identifizieren. Ein weiteres Verfahren ist das **Operating System (OS)-Fingerprinting**, bei dem anhand charakteristischer Merkmale des Netzwerkverkehrs Rückschlüsse auf das verwendete Betriebssystem gezogen werden (Nainar & Panda, 2022).

2.5.2 Spoofing-Angriffe

Beim *Spoofing* verändert ein Angreifer die zwischen Absender und Empfänger übertragenen Pakete und kann diese Manipulation unter anderem für *Man-in-the-Middle (MITM)*-Angriffe nutzen. Dies ist insbesondere bei unverschlüsselter Kommunikation möglich, da Netzwerkpakete von Dritten mitgelesen und manipuliert werden können (Nainar & Panda, 2022).

ARP ist ein häufiges Ziel von Spoofing-Angriffen. Dabei werden *Media Access Control (MAC)*-Adressen in ARP-Tabellen manipuliert, sodass sämtlicher Datenverkehr über den Angreifer geleitet wird. Angreifer können auch das **Dynamic Host Configuration Protocol (DHCP)** ausnutzen, indem sie gezielt gefälschte ACK-Antworten auf eingehende DHCP-Anfragen senden, um den späteren Datenverkehr umzuleiten. Auch das **Domain Name System (DNS)** ist ein häufiges Ziel von Spoofing-Angriffen, bei denen durch kompromittierte DNS-Server der DNS-Cache manipuliert wird (Nainar & Panda, 2022).

2.5.3 Brute-Force-Angriffe

Bei *Brute-Force-Angriffen* wird versucht, Zugangsdaten wie Benutzernamen und Passwörter durch das Ausprobieren sämtlicher möglicher Kombinationen zu erraten, um Zugang zu einem Gerät zu erhalten. Dabei wird eine enorme Menge an Netzwerkverkehr erzeugt, was diese Angriffe in der Regel leicht erkennbar macht (Nainar & Panda, 2022).

2.5.4 Denial of Service (**DoS**)-Angriffe

Eine große Menge an Netzwerkverkehr tritt auch bei *Denial of Service (DoS)-Angriffen* auf, bei denen eine große Anzahl von Anfragen an einen Server gesendet wird, die dessen Kapazität überschreitet und er legitime Anfragen nicht mehr bearbeiten kann. Häufig werden auch koordinierte *Distributed Denial of Service (DDoS)-Angriffe* durchgeführt, bei denen zahlreiche (häufig manipulierte) Geräte beteiligt sind (Nainar & Panda, 2022). DoS-Angriffe können unter anderem auf Protokoll- und Anwendungsebene durchgeführt werden und sind daher besonders effektiv (Ndatinya et al., 2015).

2.5.5 Malware-Angriffe

Eine weitere, neben den bereits vorgestellten Angriffsmustern, sehr häufige Angriffsform ist *Malware*. Aufgrund ihrer unterschiedlichen Ausprägungen, der teils nicht hinterlassenen eindeutigen Spuren und der Tatsache, dass nicht jede Malware aktiv Netzwerkverkehr erzeugt, ist ihre Identifikation im Rahmen forensischer Maßnahmen oft schwierig. Klassische Beispiele für Malware sind Viren, Würmer, Trojaner, Spyware, Phishing und Ransomware (Nainar & Panda, 2022).

3 Forensische Analyse mit Wireshark

Wireshark ist eine quelloffene Software zur Echtzeit-Erfassung und Analyse von Netzwerkpaketen. Neben der Live-Paketerfassung ermöglicht es auch die detaillierte Untersuchung aufgezeichneter Daten. Durch leistungsstarke Filtermechanismen lassen sich relevante Pakete gezielt extrahieren, wodurch Wireshark ein unverzichtbares Werkzeug in der Netzwerkforensik darstellt (Nainar & Panda, 2022).

Grafische Benutzeroberfläche

Wireshark bietet eine grafische Benutzeroberfläche (siehe Abbildung 1) und mit TShark auch ein Kommandozeilenprogramm an. Beispiele und Ausführungen in dieser Arbeit erfolgen ausschließlich auf Basis der grafischen Benutzeroberfläche.

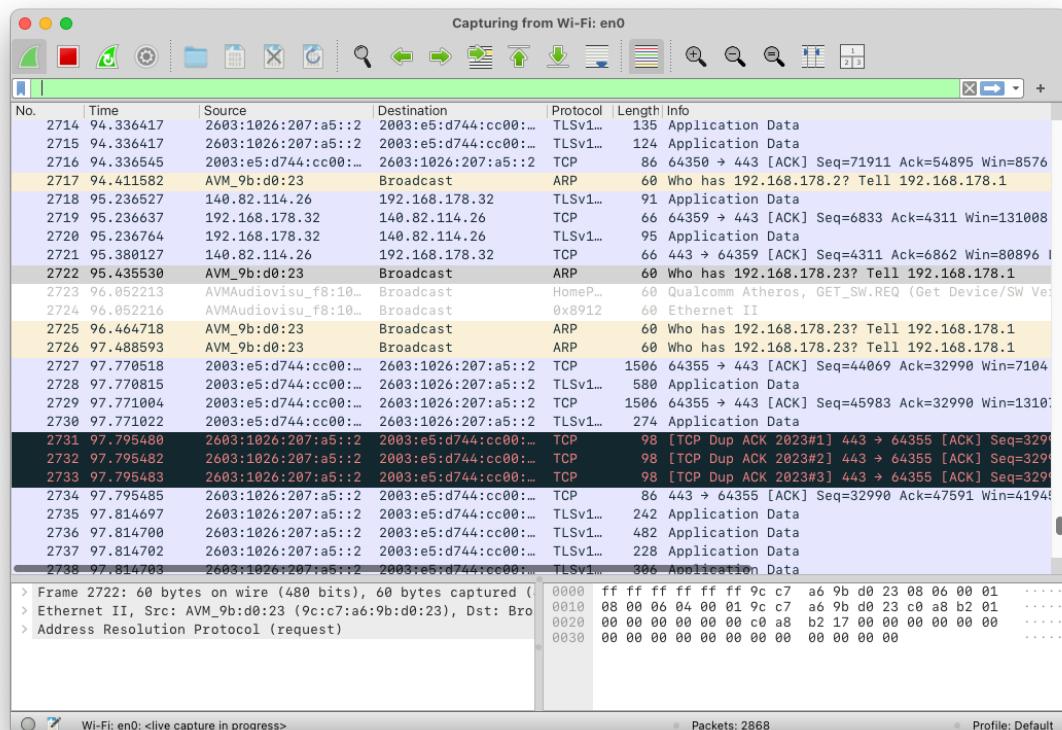


Abbildung 1: Echtzeit-Paketerfassung mit Wireshark

3.1 Wireshark als Analysewerkzeug

Um Netzwerkverkehrsdaten einer forensischen Analyse mit Wireshark zu unterziehen, müssen sie zunächst erfasst werden. Hierzu kann entweder eine passende Netzwerkschnittstelle ausgewählt werden, um Live-Paketerfassung durchzuführen, oder es wird eine *packet capture (pcap)*-Datei mit zuvor aufgezeichneten Paketdaten importiert.

3.1.1 Betriebsmodus bei Live-Paketerfassung

Bei der Echtzeit-Paketerfassung erlaubt Wireshark als Datenquelle physische oder virtuelle Netzwerkschnittstellen, darunter Netzwerkarten für kabelgebundene und kabellose Kommunikation sowie virtuelle Adapter. Netzwerkarten können für die Datenerfassung in unterschiedlichen Varianten betrieben werden (nach Cardwell, 2023):

- **Normaler Modus:** Es werden lediglich diejenigen Datenpakete verarbeitet, deren Ziel-MAC-Adresse der Netzwerkkarte oder des Broadcasts entspricht. Andere Pakete werden verworfen und nicht erfasst.
- **Promiskuitiver Modus:** Anders als beim normalen Modus werden hier alle empfangenen Datenpakete erfasst und weitergeleitet. Dies ist erforderlich, um später den gesamten Netzwerkverkehr zu analysieren.

Netzwerkarten für kabellose Kommunikation können in vier unterschiedlichen Varianten betrieben werden. Der **Monitor-Modus** ist vergleichbar mit dem promiskutiven Modus und damit für die Netzwerkanalyse am besten geeignet (Cardwell, 2023). Abbildung 1 zeigt etwa die Live-Paketerfassung über eine Wi-Fi-Schnittstelle, die im Monitor-Modus betrieben wird.

3.1.2 Nutzung aufgezeichneter pcap-Daten

Wireshark und andere Programme, wie zum Beispiel `tcpdump`, speichern vollständig erfasste Paketdaten in `pcap`-Dateien, die auf der `libpcap`-Bibliothek basieren (Nainar & Panda, 2022). Der Aufbau dieser Dateien ist immer gleich: Sie beginnen mit einem globalen Header mit verschiedenen Metadaten, gefolgt von den erfassten Paketdaten, die jeweils aus einem Paket-Header und den eigentlichen Paketdaten bestehen. Werden `pcap`-Dateien extern generiert, können sie in Wireshark importiert und zur weiteren Paketanalyse verwendet werden (Joshi & Pilli, 2016). Dabei wird neben dem älteren `pcap`- auch das neuere *packet capture next generation (pcapng)*-Dateiformat unterstützt (Nainar & Panda, 2022).

Ein Beispiel für die Paketerfassung und Speicherung in einer `pcap`-Datei mittels `tcpdump` ist in Listing 1 gegeben.

Listing 1: Netzwerk-Paketerfassung mittels tcpdump

```
1 $ tcpdump -i en0 -w capture.pcap -C 1 -c 100 -v
2   tcpdump: listening on en0, link-type EN10MB (Ethernet),
3     ↳ snapshot length 524288 bytes
4   100 packets captured
5   101 packets received by filter
6   0 packets dropped by kernel
```

3.1.3 Datenverarbeitung durch Dekodierung & Rekonstruktion

Wireshark erkennt und interpretiert die Paketdaten über eingebaute *Dissektoren*. Diese dekodieren die Binärdaten in eine lesbare Form und ermöglichen so eine einfache Darstellung der Rohdaten in einer hierarchischen Ansicht. Darüber hinaus können in Wireshark vollständige Datenkommunikationssequenzen (*Streams*) rekonstruiert werden, um etwa den gesamten Weg vom Zeitpunkt des Handshakes bis zur Beendigung der Verbindung nachzuvollziehen (Cardwell, 2023).

3.2 Filterung von Netzwerkdaten

Ein essenzieller Bestandteil der Paketerfassung und Analyse von Netzwerkdaten ist die Möglichkeit der Filterung. Angesichts des stetig wachsenden Netzwerkverkehrs durch die zunehmende Zahl vernetzter Geräte ist eine ungefilterte Analyse kaum noch praktikabel. Die große Datenmenge erfordert eine gezielte Untersuchung. Wireshark bietet hierfür zwei zentrale Filtermechanismen (Nainar & Panda, 2022).

3.2.1 Capture-Filter

Ein Capture-Filter limitiert die erfassten Paketdaten anhand vorgegebener Ausdrücke. Somit werden nur noch diejenigen Paketdaten erfasst, die je nach eingesetzten Filtern beispielsweise einem entsprechenden Protokolltyp oder anderen Protokollparametern entsprechen (Nainar & Panda, 2022). Listing 2 zeigt beispielhaft eine Filterung aller *Secure Shell (SSH)*-Verbindungen an die IP-Adresse 96.7.128.175.

Listing 2: Capture-Filter in libpcap-Syntax

```
1  tcp port 22 and dst host 96.7.128.175
```

3.2.2 Display-Filter

Ähnlich wie Capture-Filter dienen auch Display-Filter dazu, Paketdaten anhand bestimmter Kriterien einzuzgrenzen. Der entscheidende Unterschied liegt jedoch im Anwendungszeitpunkt: Während Capture-Filter bereits während der Erfassung festlegen, welche Pakete gespeichert werden, ermöglichen Display-Filter eine nachträgliche Selektion innerhalb der aufgezeichneten Daten. In Wireshark unterscheidet sich die Syntax der Display-Filter von der der Capture-Filter und bietet zudem erweiterte Filtermöglichkeiten (Nainar & Panda, 2022). Ein Beispiel zeigt Listing 3, in dem alle ARP-Pakete eines bestimmten Adressbereichs gefiltert werden.

Listing 3: Display-Filter in Wireshark-Syntax

```
1  arp and ip.addr == 192.168.1.0/24
```

3.3 Identifikation von Angriffsmustern anhand von **pcap**-Daten

Mit **pcap**-Dateien lassen sich Angriffsszenarien gut darstellen und reproduzieren, was für eine forensische Analyse unumgänglich ist. Außerdem spielen sie eine wichtige Rolle für die Beweissicherung im Rahmen forensischer Untersuchungen, bei denen häufig eine nachvollziehbare Beweismittelkette (*chain of custody*) erforderlich ist (Cardwell, 2023).

Nachfolgend wird aufgezeigt, wie sich die in [Abschnitt 2.5](#) vorgestellten Angriffsmuster anhand erfasster Paketdaten identifizieren lassen. Die Analyse erfolgt unter Anwendung verschiedener Display-Filter.

3.3.1 Port-Scanning

Port-Scans lassen sich in *vertikale* und *horizontale* Scans unterteilen. Bei einem vertikalen Scan wird ein einzelner Host auf möglichst viele offene Ports überprüft. Ein horizontaler Scan hingegen zielt darauf ab, einen bestimmten Port auf möglichst vielen Hosts zu identifizieren – häufig im Zusammenhang mit bekannten Sicherheitslücken, die auf diesem Port ausgenutzt werden können (Ndatinya et al., 2015).

Eine Variante des Port-Scans, der sog. *Stealth-Scan*, ist deutlich in [Abbildung 2](#) sichtbar. Dabei werden massenhaft SYN-Pakete über das *Transmission Control Protocol (TCP)* versendet, ohne den vollständigen *Three-Way-Handshake* abzuschließen (Cardwell, 2023). Mögliche Display-Filter sind in [Listing 4](#) und [Listing 5](#) gegeben.

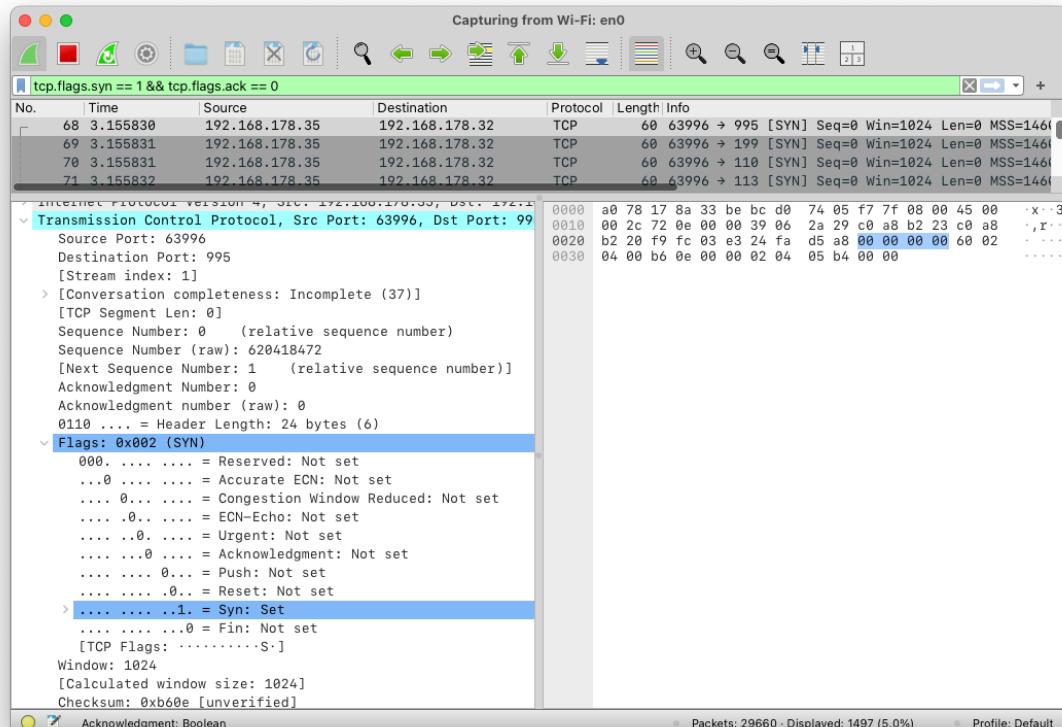


Abbildung 2: Anzeige übermittelter Pakete im Rahmen eines Stealth-Scans

Listing 4: Display-Filter zur Erkennung eines vertikalen Stealth-Scans

```
1  tcp.flags.syn == 1 && tcp.flags.ack == 0 && ip.dst ==
  ↳ 192.168.178.32
```

Listing 5: Display-Filter zur Erkennung eines horizontalen Stealth-Scans

```
1  tcp.flags.syn == 1 && tcp.flags.ack == 0 && tcp.port == 443
```

3.3.2 Ping-Sweep

Ein Ping-Sweep dient dazu, festzustellen, welche Hosts in einem Netzwerk erreichbar sind. Eine spezielle Variante davon sind sog. *Echo-Requests*, die auf den ersten Blick wie gewöhnliche ICMP-Pings wirken, sich jedoch in einem Detail unterscheiden: Sie enthalten ausschließlich 0 Byte an Nutzdaten. Diese Eigenschaft macht sie in Wireshark besonders leicht identifizierbar. [Listing 6](#) zeigt einen passenden Display-Filter, während [Abbildung 3](#) die entsprechende Ansicht in Wireshark darstellt (Nainar & Panda, 2022).

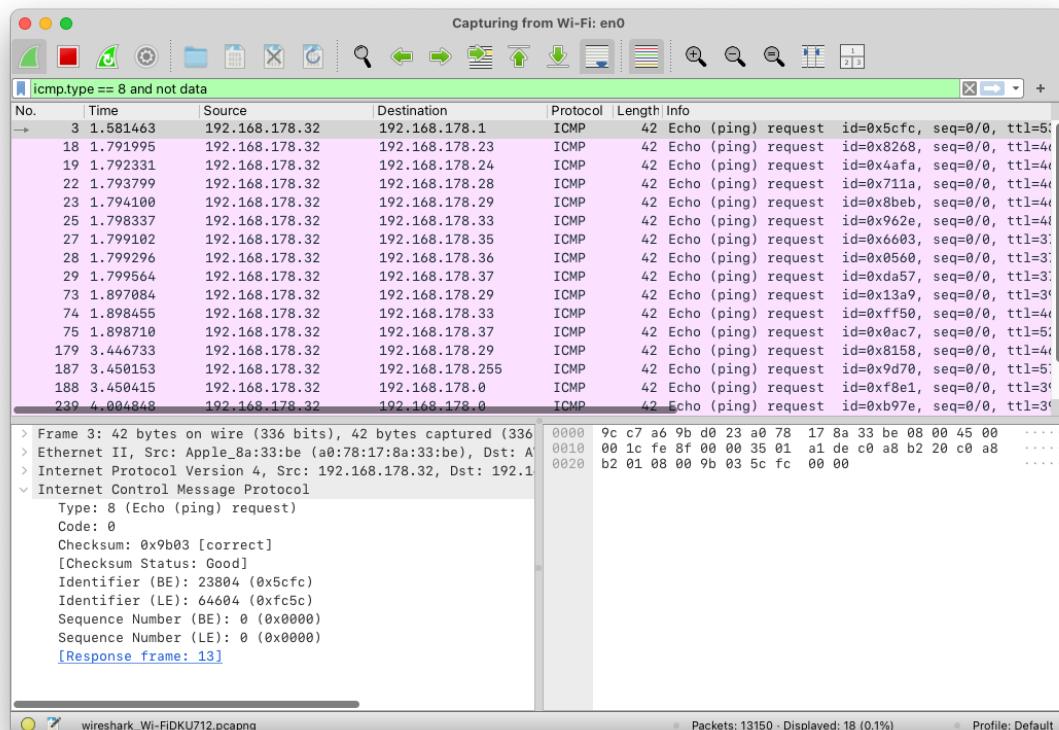


Abbildung 3: Anzeige übermittelter ICMP-Pakete im Rahmen eines Ping-Sweeps

Listing 6: Display-Filter zur Erkennung eines Ping-Sweeps

```
1  icmp.type == 8 and not data
```

3.3.3 ARP-Spoofing

ARP-Spoofing wird häufig genutzt, um [MITM](#)-Angriffe durchzuführen. Dabei sendet ein Angreifer manipulierte ARP-Antworten, um seine eigene MAC-Adresse als Ziel für eine bestimmte IP-Adresse auszugeben. Dadurch wird der gesamte Datenverkehr über ihn umgeleitet. Infolgedessen existieren im Netzwerk mehrere unterschiedliche MAC-Adressen für dieselbe IP-Adresse – ein Anzeichen, das sich in Wireshark gut sichtbar machen lässt (siehe [Abbildung 4](#) und [Listing 7](#)) (Cardwell, 2023).

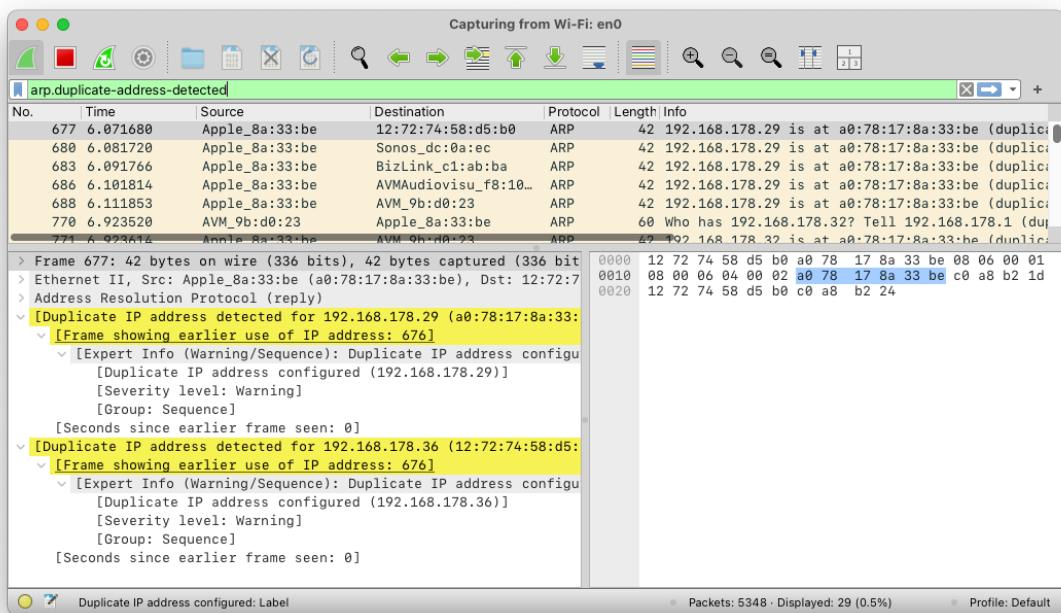


Abbildung 4: Anzeige von IP-Adressduplicaten während eines ARP-Spoofing-Angriffs

Listing 7: Display-Filter zur Erkennung von IP-Adressduplicaten

```
1 arp.duplicate-address-detected
```

3.3.4 DDoS-Angriffe

DDoS-Angriffe lassen sich in Wireshark anhand charakteristischer Verhaltensmuster identifizieren. Ein häufiges Beispiel ist ein *SYN-Flooding*-Angriff, bei dem eine große Anzahl an SYN-Paketen von zahlreichen beteiligten Hosts (oft Teil eines Botnets) an das Ziel gesendet wird, ohne dass die entsprechenden SYN-ACK-Antworten beantwortet werden. Dadurch bleibt – ähnlich wie beim Stealth-Scan in [Abschnitt 3.3.1](#) – der Three-Way-Handshake unvollständig, was eine Vielzahl halboffener Verbindungen erzeugt und das Zielsystem schließlich überlastet (Cardwell, 2023). Unbeantwortete SYN-Pakete lassen sich durch die in [Listing 4](#) und [Listing 5](#) vorgestellten Display-Filter leicht erkennen. Zum Vergleich ist in [Abbildung 5](#) ein vollständiger Three-Way-Handshake abgebildet.

Source	Destination	Protocol	Length	Info
192.168.178.27	192.168.178.1	TCP	78	61269 → 80 [SYN] Seq=0 Win=65535 L
192.168.178.1	192.168.178.27	TCP	74	80 → 61269 [SYN, ACK] Seq=0 Ack=1
192.168.178.27	192.168.178.1	TCP	66	61269 → 80 [ACK] Seq=1 Ack=1 Win=1

Abbildung 5: Vollständiger Three-Way-Handshake in Wireshark

Nutzung realer Paketerfassungsdaten in Beispielen

Zur besseren Veranschaulichung werden in den nachfolgenden Beispielen Paketerfassungsdaten realer DDoS-Angriffe verwendet. Die Daten sind dem öffentlichen GitHub-Repository *StopDDoS/packet-captures*¹ entnommen.

Bei dem nachfolgenden SYN-Flooding-Angriff wurden massenhaft Pakete an Port 25565 des Zielsystems versendet. Dies lässt der unter [Listing 8](#) abgebildete Display-Filter erkennen, der auch in [Abbildung 6](#) angewendet wurde.

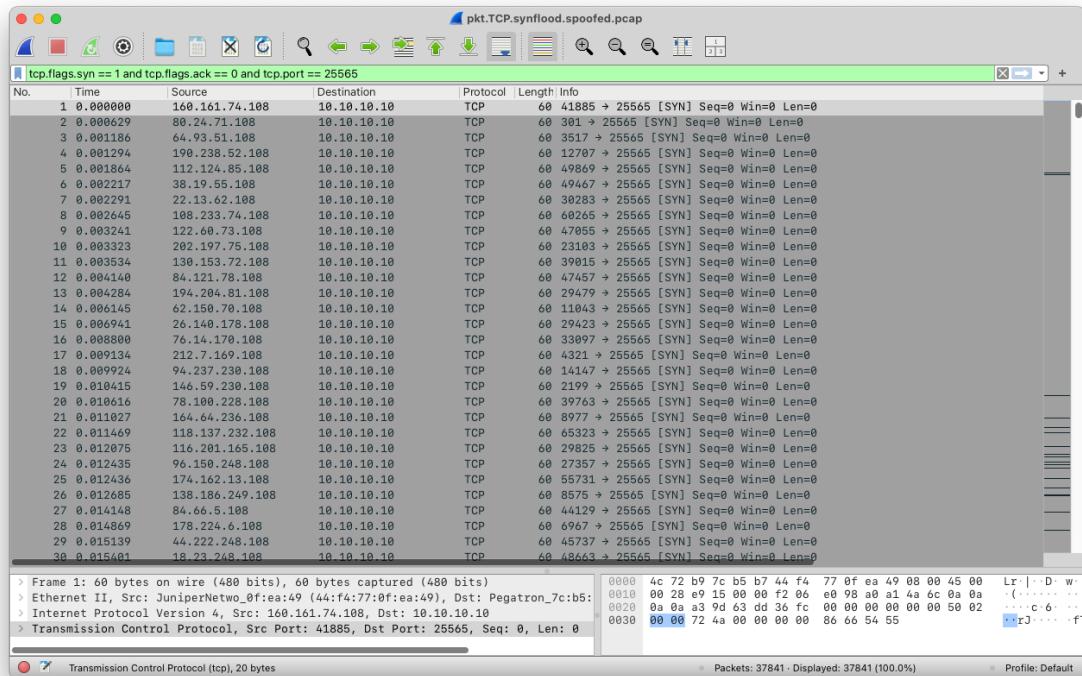


Abbildung 6: SYN-Flooding auf Port 25565 des Zielsystems

Listing 8: Display-Filter zur Erkennung eines SYN-Floodings

```
1  tcp.flags.syn == 1 and tcp.flags.ack == 0 and tcp.port ==
   ↪ 25565
```

¹Das GitHub-Repository ist unter <https://github.com/StopDDoS/packet-captures> verfügbar. Der verwendete Datensatz `pkt.TCP.synflood.spoofed.pcap` basiert auf dem Commit `f763b84` (abgerufen am 20. Februar 2025).

Darüber hinaus stellt Wireshark nützliche Werkzeuge zur statistischen Analyse von DDoS-Angriffen bereit. So kann beispielsweise anhand der geografischen Verteilung der beteiligten Hosts klar erkannt werden, ob es sich um einen globalen Angriff handelt. Unter *Statistics* → *Endpoints* listet Wireshark die beteiligten Hosts übersichtlich auf (siehe Abbildung 7) und ermöglicht die Erstellung einer geografischen Übersicht (siehe Abbildung 8) (Nainar & Panda, 2022).

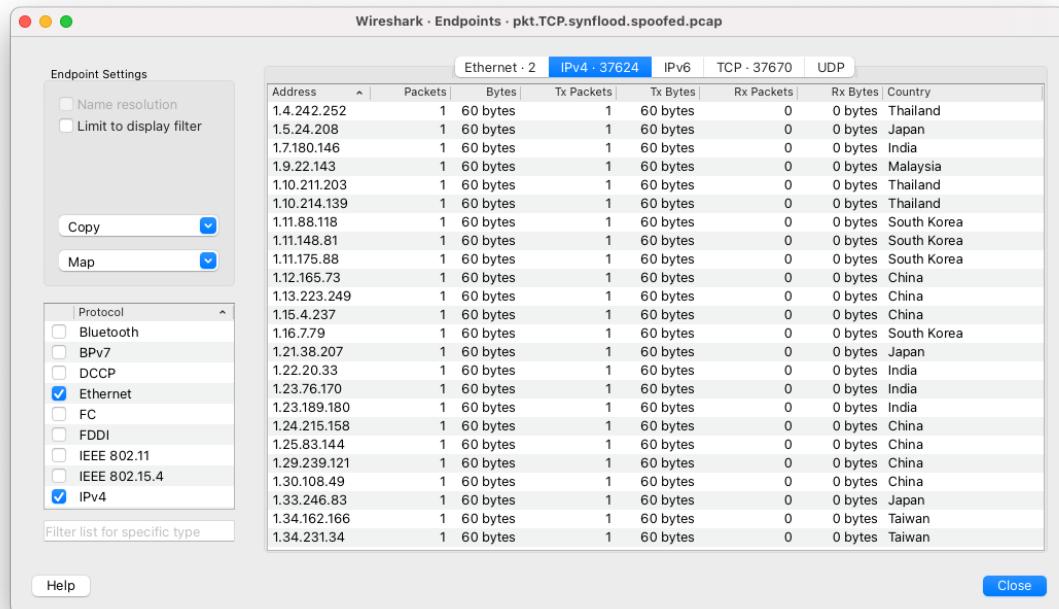


Abbildung 7: Übersicht erfasster Endpunkte während eines SYN-Floodings

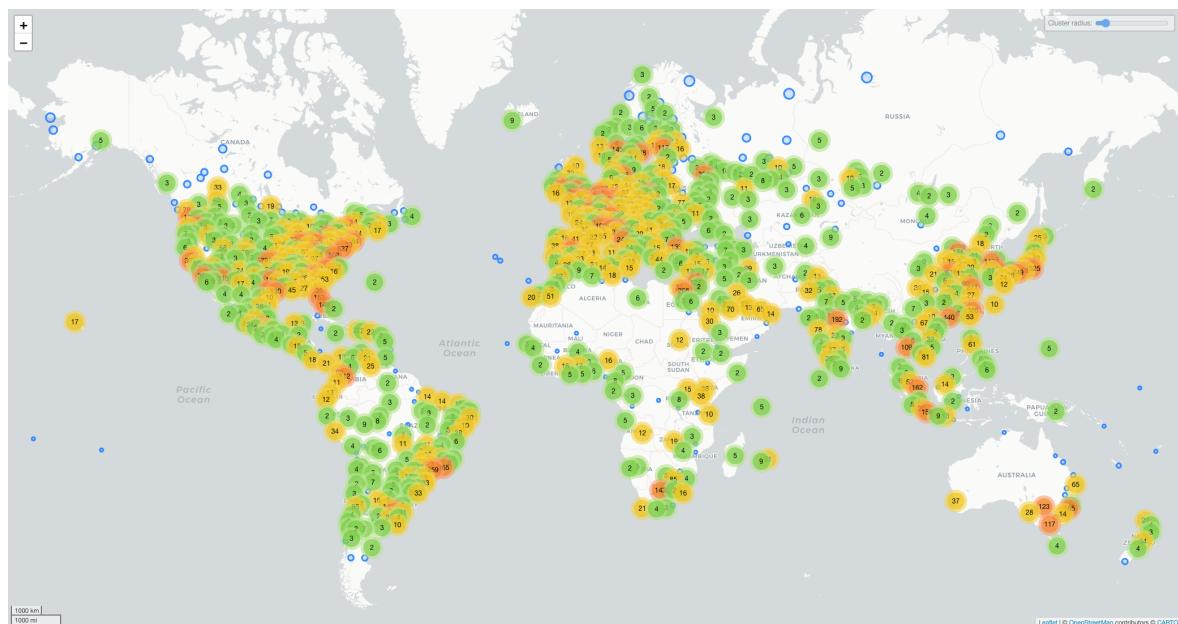


Abbildung 8: Geografische Darstellung der an einem SYN-Flooding beteiligten Hosts

3.3.5 Brute-Force-Angriffe

Brute-Force-Angriffe, beispielsweise auf das **SSH**-Protokoll, lassen sich in Wireshark gut identifizieren, da sie eine große Menge nahezu identischer Netzwerkpakete erzeugen. Diese Anomalie zeigt sich nicht nur in der hohen Anzahl an Anfragen an einen bestimmten Port, sondern auch darin, dass sie meist von einem einzelnen Host ausgehen. Mithilfe eines Display-Filters kann der Angriff effektiv sichtbar gemacht werden, da das **SSH**-Protokoll typischerweise über Port 22 läuft (siehe Abbildung 9 und Listing 9) (Cardwell, 2023).

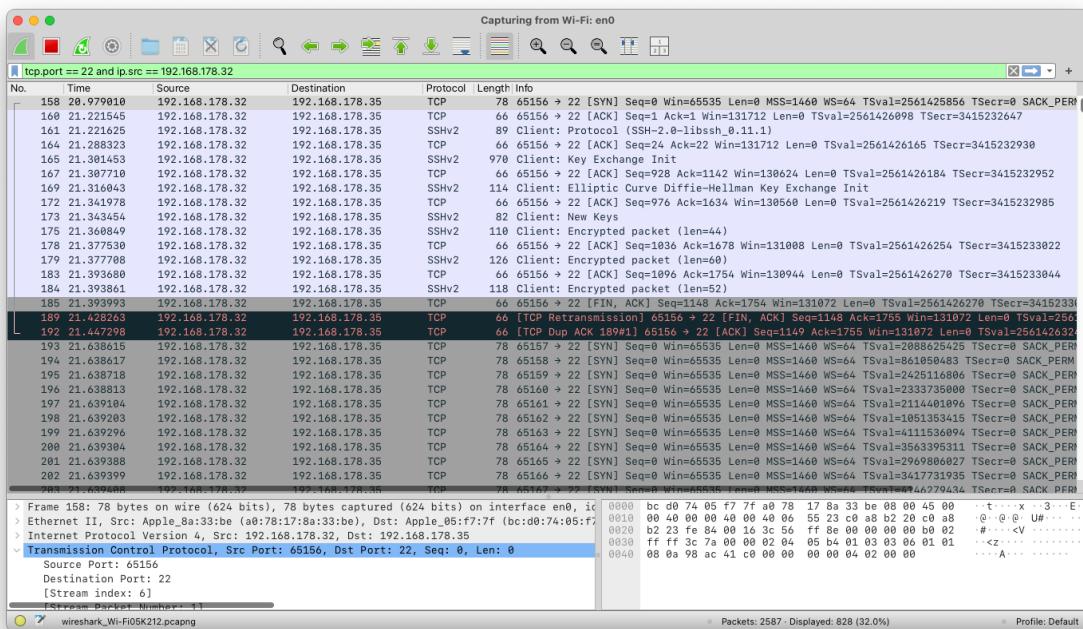


Abbildung 9: Brute-Force-Angriff auf das **SSH**-Protokoll

Listing 9: Display-Filter zur Erkennung eines Brute-Force-Angriffs

```
tcp.port == 22 and ip.src == 192.168.178.32
```

3.4 Grenzen der Angriffserkennung & alternative Werkzeuge

Wireshark spielt eine zentrale Rolle in der forensischen Analyse des Netzwerkverkehrs, ist aber nicht für jeden Anwendungsfall uneingeschränkt nutzbar. Prävention, Verteidigung und Analyse von Angriffen erfordern weitaus mehr Kompetenzen, als ein Paketanalysetool wie Wireshark liefern kann.

3.4.1 Limitierung auf (passive) Analyse

Die umfassenden Analysemöglichkeiten von Wireshark stellen zugleich seine größte Einschränkung bei der Angriffsprävention und -abwehr dar. Da es in der Regel als passives

Analysetool eingesetzt wird, können Angriffe meist erst nach der Aufzeichnung erkannt werden – zu einem Zeitpunkt, an dem Gegenmaßnahmen oft nicht mehr wirksam sind. Zwar kann Wireshark zur Prävention zukünftiger Angriffe beitragen, bietet jedoch nur begrenzte Unterstützung bei der Abwehr laufender Attacken (Banerjee et al., 2010).

Alternativen

Mit *Intrusion Detection System (IDS)* und *Intrusion Prevention System (IPS)* existieren zwei speziell auf die Prävention und Abwehr von Angriffen ausgelegte Systeme. Sie sind direkt in das Netzwerk integriert und können Anomalien in Echtzeit identifizieren und unmittelbar darauf reagieren (Banerjee et al., 2010).

3.4.2 Begrenzte Skalierbarkeit

In Hochgeschwindigkeitsnetzwerken werden enorme Datenmengen erzeugt, die ein Analysetool wie Wireshark vollständig verarbeiten muss, um eine lückenlose Erfassung und Speicherung zur Durchführung forensischer Analysen zu gewährleisten. Oft können hierbei nicht alle Pakete aufgezeichnet werden, was zu unvollständigen Protokollen und erschwerter Angriffsanalyse führt (Khan et al., 2016).

Alternativen

Spezialisierte Erfassungssysteme, verteilte Packet-Capturing-Methoden und optimierte Speicherlösungen wie *Time Machine Packet Capturing* sind für den skalierbaren Einsatz konzipiert und eignen sich insbesondere für Szenarien, in denen klassische Tools wie Wireshark an ihre Grenzen stoßen (Khan et al., 2016).

3.4.3 Einschränkungen durch verschlüsselten Datenverkehr

Wireshark unterstützt die Erkennung und Verarbeitung einer Vielzahl von Netzwerkprotokollen, von denen viele mittlerweile auch in verschlüsselter Form eingesetzt werden können, um die Sicherheit für Anwender*innen zu erhöhen. Ein großer Teil des Netzwerkverkehrs erfolgt heute bereits über das *Hypertext Transfer Protocol Secure (HTTPS)*, wodurch Paketdaten nicht mehr im Klartext sichtbar sind. Einzelne Informationen, etwa im Rahmen des Three-Way-Handshakes, lassen sich zwar weiterhin ermitteln, jedoch erschwert der Einsatz verschlüsselter Protokolle insgesamt die Analyse bestimmter Angriffsformen erheblich und stellt eine Herausforderung für die forensische Untersuchung dar (Cardwell, 2023; Nainar & Panda, 2022).

3.5 Wireshark im Vergleich mit anderen Forensik-Tools

Paketanalyse ist nur ein kleiner Bestandteil der Netzwerkforensik. Darüber hinaus existieren viele weitere Anwendungsgebiete.

3.5.1 Schwachstellenbewertung

Ziel dieser Tools ist die Identifikation und Bewertung von Schwachstellen in einem System, um dessen Anfälligkeit für böswillige Angriffe oder unbefugte Zugriffe zu minimieren. Häufig werden hierzu auch gefälschte Angriffe durchgeführt, um Schwachstellen besser zu erkennen. Verglichen mit Wireshark haben Tools zur Schwachstellenbewertung den Vorteil, dass sie eine direkte Identifizierung von Angriffen ermöglichen. In Wireshark ist dies hingegen nur durch eine überwiegend manuelle Analyse möglich, weshalb auch Programme wie *Metasploit* einen wichtigen Bestandteil der Netzwerkforensik darstellen (Joshi & Pilli, 2016).

3.5.2 Netzwerkscanner

Das bereits in [Abschnitt 2.5.1](#) vorgestellte Verfahren ist nicht nur für Angreifer von Bedeutung. Es stellt eine wichtige Sicherheitsstrategie zum Auffinden aktiver Hosts und der von ihnen angebotenen Dienste in einem Netzwerk dar und dient damit vor allem dem Aufspüren potenzieller Angreifer. Wireshark, als vorrangig passives Analysewerkzeug, eignet sich im Gegensatz zu Netzwerkscannern nicht für die präventive Analyse. Tools wie *Network Mapper* (*Nmap*) hingegen ermöglichen in der Regel keine tiefgehenden Untersuchungen nach einem Sicherheitsvorfall, sondern konzentrieren sich primär auf die frühzeitige Identifikation potenzieller Angriffsvektoren (Joshi & Pilli, 2016).

3.5.3 Netzwerküberwachung

Die Verfügbarkeit eines Netzwerks oder Netzwerksegments kann von entscheidender Bedeutung sein, weshalb eine effiziente Überwachung von Leistung, *Quality of Service* (*QoS*), Verzögerung und Bandbreite erforderlich ist. Überwachungstools wie `tcpstat` oder `ntop` erfassen kontinuierlich den Netzwerkverkehr, um wichtige Einblicke in den Zustand eines Netzwerks zu ermöglichen. Während Wireshark ebenfalls Pakete in Echtzeit aufzeichnen kann, bietet es keine detaillierte Performance-Analyse und ist nicht für eine dauerhafte Überwachung ausgelegt (Joshi & Pilli, 2016).

3.5.4 Intrusion Detection System (IDS)

Diese Software- oder Hardware-basierten Systeme dienen der Überwachung eines Netzwerks auf potenziell bösartige Aktivitäten. Im Fokus steht die Erkennung verdächtigen Datenverkehrs, wobei einige **IDS** auch in der Lage sind, unautorisierte Zugriffe zu unterbinden. Ein bekanntes *Network Intrusion Detection System* (*NIDS*) ist *Snort*. Wie bereits in [Abschnitt 3.4.1](#) dargestellt, ist ein **IDS** eine wertvolle Ergänzung im Bereich der Netzwerkforensik, da Tools wie Wireshark durch ihren Einsatz als passives Analysetool bei der Echtzeit-Erkennung von Angriffen nicht geeignet sind (Joshi & Pilli, 2016).

4 Strategischer Nutzen der Netzwerkforensik

Die strategische Bedeutung der Netzwerkforensik ergibt sich aus ihren praktischen Anwendungsfeldern (siehe [Abschnitt 2.2](#)). Da die Implementierung forensischer Maßnahmen und die Durchführung entsprechender Analysen mit erheblichem Aufwand verbunden sein kann, müssen Unternehmen sorgfältig abwägen, inwieweit eine Einführung sinnvoll ist. Unabhängig vom konkreten Einzelfall lassen sich mehrere grundlegende Argumente identifizieren, die für den Einsatz von Netzwerkforensik sprechen.

4.1 Beweissicherung & Strafverfolgung

Die stetige Zunahme des Netzwerkverkehrs sowie immer komplexere und raffiniertere Angriffsszenarien führen zu einer wachsenden globalen Bedrohungslage. Prominente Angriffe auf Unternehmen wie Google, Facebook oder Twitter verdeutlichen diese Entwicklung. Auch Phishing-Angriffe auf Bankkonten rücken zunehmend in den Fokus, da sie erhebliche finanzielle und rechtliche Konsequenzen nach sich ziehen können. Durch den Einsatz von Netzwerkforensik können Angreifer nicht mehr unbemerkt agieren, da digitale Spuren gesichert werden, die eine Identifikation und rechtliche Verfolgung ermöglichen. Diese gerichtsfesten Beweise tragen nicht nur zur Strafverfolgung bei, sondern wirken zugleich abschreckend auf potenzielle Täter (Khan et al., [2016](#)).

4.2 Schutz kritischer Systeme & Geschäftsmodelle

Viele Unternehmen sind heute auf sichere Online-Transaktionen angewiesen, etwa im Bereich *E-Business*. Damit verbunden ist das Vertrauen der Anwender*innen, dass sie beim Nutzen dieser Systeme auf ein hohes Maß an Sicherheit zählen können. Gerade diese Systeme sind daher häufig Ziel groß angelegter Angriffe, was für Unternehmen sowohl Rufschädigung als auch im schlimmsten Fall den wirtschaftlichen Ruin zur Folge haben kann. Netzwerkforensik trägt in diesem Kontext dazu bei, indem sie schädlichen Netzwerkverkehr aktiv filtert. Durch forensische Analysen können manipulierte Netzwerkpakete identifiziert und Malware aufgedeckt werden (Khan et al., [2016](#)).

4.3 Unterstützung bei Compliance-Anforderungen

Die Einhaltung strenger Gesetze wie dem *Health Insurance Portability and Accountability Act (HIPAA)* in den USA stellt Unternehmen zunehmend vor Herausforderungen. Diese Gesetze verlangen oft die Implementierung umfassender Sicherheitsprogramme, um sensible Daten vor böswilligen Angriffen zu schützen. Durch die Überwachung und Analyse ihres Netzwerkverkehrs können Unternehmen nicht nur sicherstellen, dass sie den rechtlichen Anforderungen entsprechen, sondern auch das Vertrauen der Benutzer*innen gewinnen und so ihr Marktportfolio erweitern (Khan et al., [2016](#)).

5 Zusammenfassung

5.1 Ergebnisse

Wireshark ist ein vielseitiges Tool für die forensische Analyse und die Identifikation typischer Angriffsmuster. Sniffing, Spoofing, DDoS- und Brute-Force-Angriffe lassen sich mithilfe von Display-Filtern und statistischen Analysetools in Wireshark effektiv erkennen. Für Unternehmen bietet Wireshark aufgrund seiner breiten Anwendbarkeit in der Praxis zahlreiche Vorteile. Da nahezu überall Netzwerkverkehr stattfindet, kann Wireshark auf vielfältige Weise genutzt werden, um Angreifer zu identifizieren, regulatorische Vorgaben wie ISO 27001 zu erfüllen und bestehende Sicherheitslösungen zu optimieren. Der strategische Nutzen von Wireshark wird insbesondere durch den Schutz kritischer Systeme und Geschäftsmodelle sowie die Möglichkeit der gerichtsfesten Beweissicherung deutlich, wodurch Behörden aktiv Strafverfolgungsmaßnahmen einleiten können.

Jedoch ist die Paketanalyse nicht das einzige relevante Mittel in der Netzwerkforensik. Sie stellt vielmehr einen wichtigen Baustein innerhalb eines umfassenderen Sicherheitsansatzes dar. Die Begrenzung auf eine primär passive Analyse und die eingeschränkte Skalierbarkeit sind nur zwei Gründe, warum eine forensische Untersuchung mit Wireshark an die Grenzen der Angriffserkennung stoßen kann. Hier bieten sich ergänzende oder alternative Tools wie IDS, IPS und spezialisierte Speicherlösungen wie *Time Machine Packet Capturing* an, die die Analyse erweitern. Ein weiterer limitierender Faktor ist die zunehmende Verschlüsselung des Netzwerkverkehrs, die den Einsatz von Analysetools wie Wireshark, insbesondere im Bereich der *Deep Packet Inspection (DPI)*, einschränkt. Dennoch bleibt Wireshark eines der zentralen Werkzeuge in der Netzwerkforensik. In Kombination mit Netzwerkscannern, Netzwerküberwachung und Tools zur Schwachstellenbewertung entsteht ein umfangreicher Werkzeugkasten zur Erkennung, Verteidigung, Prävention und Analyse von Netzwerkangriffen.

5.2 Ausblick

Wireshark bietet neben den in dieser Arbeit vorgestellten Technologien wie Capture- und Display-Filter noch deutlich umfangreichere Features. Es erlaubt etwa die Interaktion mit *Pandas* und das Schreiben eigener Skripte mithilfe der Skriptsprache *Lua*, wodurch es anwendungsspezifisch erweitert werden kann (Cardwell, 2023). Allerdings existieren darüber hinaus noch einige neuartige Technologien, die weit über die Fähigkeiten von Wireshark hinausgehen. Einige davon setzen auf Wireshark als Einzelbestandteil, andere wiederum stellen komplett neuartige Entwicklungen dar. Nachfolgend eine kleine Auswahl aktueller Entwicklungen im Bereich der Netzwerkforensik:

- Mit Wireshark als Tool für die Paketerfassung haben Manjula und Mangla (2023) einen Ansatz zur Echtzeit-Erkennung von DDoS-Angriffen unter Verwendung

von Klassifizierungsalgorithmen wie *Naive Bayes*, *k-Nearest-Neighbor (KNN)* und *Random Forest* vorgestellt. Dabei kommt auch die von *Apache Spark* entwickelte **ML**-Bibliothek *MLib* zum Einsatz, was viele neue Möglichkeiten im Bereich *Künstliche Intelligenz (KI)* eröffnet und dieses spannende Themenfeld mit dem Bereich der Netzwerkforensik verknüpft.

- Auch Bhardwaj und Dave (2023) haben sich der Untersuchung von Angriffen in der Netzwerkforensik gewidmet und ein Framework präsentiert, das auf neuronalen Netzen basiert. Die Untersuchung erfolgt dabei auf Basis einer paketbasierten Analyse (ähnlich wie in Wireshark), einer grafischen Analyse, durch die der gesamte Angriffsfluss identifiziert werden kann, und einer forensischen explorativen Datenanalyse. Durch den Einsatz eines eindimensionalen *Convolutional Neural Network (CNN)* zeigt sich, dass das vorgeschlagene Framework effektiver ist als frühere Ansätze in der Angriffserkennung.
- Mei et al. (2024) haben ein weiteres, auf *Deep Learning* basierendes Netzwerkforensik-Framework vorgestellt, das speziell auf die Erkennung von *Advanced Persistent Threats (APTs)* abzielt. Dabei werden Feature-Filtering-Techniken eingesetzt, um wichtige Informationen zur Rückverfolgung eines Angriffs zu gewinnen. Das entwickelte Modell ist in der Lage, Anomalien im Netzwerk zu erkennen, und zeigt dabei im Vergleich zu anderen **KI**-basierten Methoden eine hohe Leistung.

Insgesamt lässt sich feststellen, dass Netzwerkforensik ein sehr wichtiger Bestandteil der IT-Sicherheit ist und es immer wieder neuartige Entwicklungen gibt, die diesen Zustand bestätigen. Tools wie Wireshark sind immer noch weit verbreitet und für die forensische Analyse unverzichtbar. Jedoch werden auch immer wieder neuartige Konzepte und Frameworks vorgestellt, die noch effizientere Analysen, Präventions- und Verteidigungsmaßnahmen bieten.

Literaturverzeichnis

- Banerjee, U., Vashishtha, A., & Saxena, M. (2010). Evaluation of the Capabilities of Wireshark as a tool for Intrusion Detection. *International Journal of computer applications*, 6(7), 1–5.
- Bhardwaj, S., & Dave, M. (2023). Enhanced neural network-based attack investigation framework for network forensics: Identification, detection, and analysis of the attack. *Computers & security*, 135, 103521–.
- Cardwell, K. (2023). *Tactical Wireshark: A Deep Dive into Intrusion Analysis, Malware Incidents, and Extraction of Forensic Evidence* (1. Aufl.). Apress.
- Hoque, N., Bhuyan, M. H., Baishya, R. C., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, 40, 307–324.
- Joshi, R., & Pilli, E. S. (2016). *Fundamentals of Network Forensics: A Research Perspective* (1. Aufl.). Springer Nature.
- Khan, S., Gani, A., Wahab, A. W. A., Shiraz, M., & Ahmad, I. (2016). Network forensics: Review, taxonomy, and open challenges. *Journal of Network and Computer Applications*, 66, 214–235.
- Kim, K., Aminanto, M. E., & Tanuwidjaja, H. C. (2018). *Network Intrusion Detection using Deep Learning*. Springer Singapore.
- Manjula, H., & Mangla, N. (2023). An approach to on-stream DDoS blitz detection using machine learning algorithms. *Materials today : proceedings*, 80, 3492–3499.
- Mei, Y., Han, W., Li, S., Lin, K., Tian, Z., & Li, S. (2024). A Novel Network Forensic Framework for Advanced Persistent Threat Attack Attribution Through Deep Learning. *IEEE transactions on intelligent transportation systems*, 25(9), 12131–12140.
- Nainar, N. K., & Panda, A. (2022). *Wireshark for Network Forensics: An Essential Guide for IT and Cloud Professionals* (1. Aufl.). Apress.
- Ndatinya, V., Xiao, Z., Manepalli, V. R., Meng, K., & Xiao, Y. (2015). Network forensics analysis using Wireshark. *International Journal of Security and Networks*, 10(2), 91–106.

Eigenständigkeitserklärung

Ich trage die Verantwortung für die Qualität des Textes sowie die Auswahl aller Inhalte und habe sichergestellt, dass Informationen und Argumente mit geeigneten wissenschaftlichen Quellen belegt bzw. gestützt werden. Die aus fremden Quellen direkt oder indirekt übernommenen Texte, Gedankengänge, Konzepte, Grafiken usw. in meinen Ausführungen habe ich als solche eindeutig gekennzeichnet und mit vollständigen Verweisen auf die jeweilige Quelle versehen. Alle weiteren Inhalte dieser Arbeit (Textteile, Abbildungen, Tabellen etc.) ohne entsprechende Verweise stammen im urheberrechtlichen Sinn von mir.

Hiermit erkläre ich, dass ich die vorliegende Studienarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle sinngemäß und wörtlich übernommenen Textstellen aus fremden Quellen wurden kenntlich gemacht.

Die vorliegende Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Erklärung zu (gen)KI-Tools

Verwendung von (gen)KI-Tools

Ich versichere, dass ich mich (gen)KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Ich verantworte die Übernahme jeglicher von mir verwendeter Textpassagen vollumfänglich selbst. In der [Übersicht verwendeter \(gen\)KI-Tools](#) habe ich sämtliche eingesetzte (gen)KI-Tools, deren Einsatzform sowie die jeweils betroffenen Teile der Arbeit einzeln aufgeführt. Ich versichere, dass ich keine (gen)KI-Tools verwendet habe, deren Nutzung der Prüfer bzw. die Prüferin explizit schriftlich ausgeschlossen hat.

Hinweis: Sofern die zuständigen Prüfenden bis zum Zeitpunkt der Ausgabe der Aufgabenstellung konkrete (gen)KI-Tools ausdrücklich als nicht anzeigen-/kennzeichnungspflichtig benannt haben, müssen diese nicht aufgeführt werden.

Ich erkläre weiterhin, dass ich mich aktiv über die Leistungsfähigkeit und Beschränkungen der unten genannten (gen)KI-Tools informiert habe und überprüft habe, dass die mithilfe der genannten (gen)KI-Tools generierten und von mir übernommenen Inhalte faktisch richtig sind.

Übersicht verwendeter (gen)KI-Tools

Die (gen)KI-Tools habe ich, wie im Folgenden dargestellt, eingesetzt.

(gen)KI-Tool	Einsatzform	Betroffene Teile der Arbeit
ChatGPT	Generierung von ersten Ideen für eine geeignete Gliederung	Gesamte Arbeit
	Generierung einer Auswahl möglicher Literaturweke	Gesamte Arbeit
	Generierung von Zusammenfassungen verschiedener Literaturwerke	Gesamte Arbeit

Münster, 22. Februar 2025



Elias Häußler