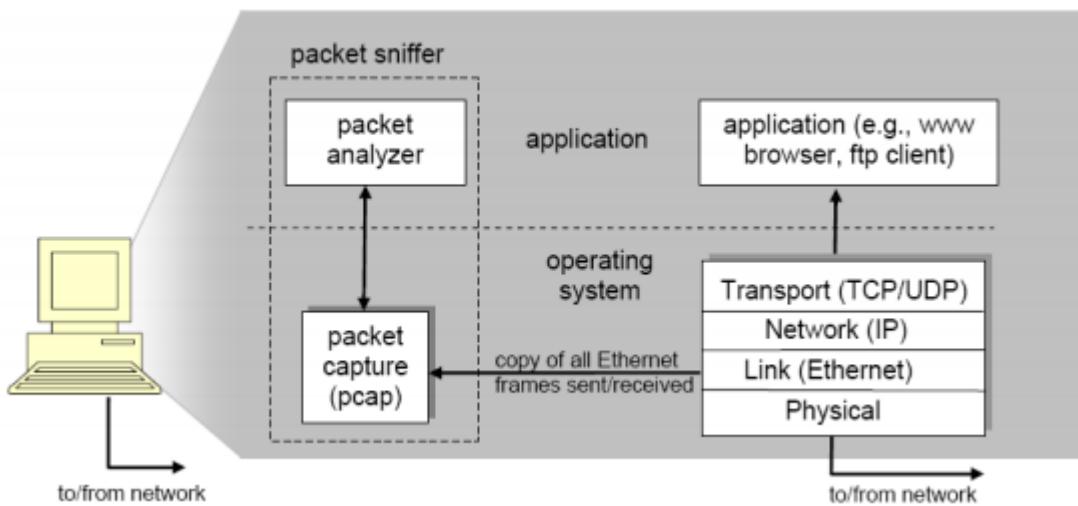


Netzwerkanalyse mit Wireshark

Das Monitoring inklusive der notwendigen Detailanalyse des Datenverkehrs im Netzwerk ist ohne ein leistungsfähiges Analysesystem unmöglich. Eines der **wichtigsten Netzwerktools** für jeden **Administrator** ist **Wireshark** – ein Open Source-Netzwerkanalysator, mit dem Sie alle **Pakete im Netzwerk aufzeichnen** und die Paketinhalte detailliert analysieren.

Packet Sniffer

Das grundlegende Werkzeug für die Beobachtung von Daten zwischen Rechnern wird als **packet sniffer** bezeichnet. Wie der Name schon sagt, **fängt** dieses Werkzeug **empfangene/gesendete Daten** Ihres Rechners **ab**. Ein Sniffer ist immer **passiv**, das heißt er **verschickt keine Daten**, sondern **speichert Kopien der Daten** der Kommunikation auf Ihrem Rechner.



Das Bild 1 zeigt die Struktur eines „Sniffers“. Rechts unten im Bild sind die beteiligten Protokolle abgebildet, es werden also Protokolle der Layer 1 bis 4:

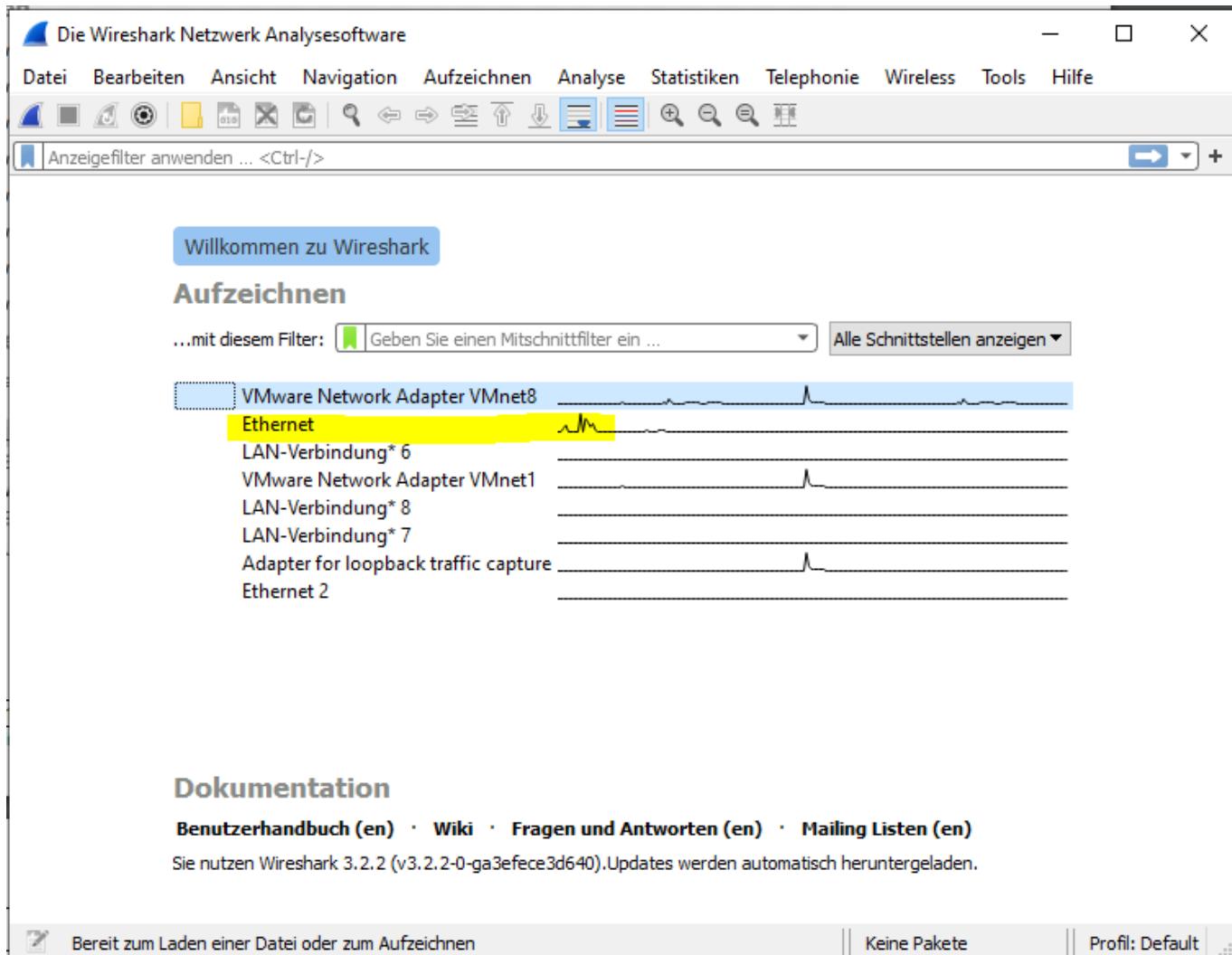
- oben ist die Applikation zu sehen, in unserem Fall ist dies der Webbrowser (Firefox, IE).
- die Blöcke die in den gestrichelten Linien eingerahmt sind gehören zu unserem „Sniffer“.

Am einfachsten lässt sich ein Netzwerk sniffen wenn Hubs als Verbindung der Netzwerkssegmente zwischengeschaltet sind. Bei anderen Verbindungen wie z.B. bei einem Switch bekommt man im Normalfall Probleme den Netzwerkverkehr abzuhören, da der Switch die Daten vom Sender nur an den tatsächlichen Empfänger weiterleitet. Somit würde man nur die eigenen Daten, sowie unwichtigeren Netzwerkverkehr wie z.B. Broadcasts aufzeichnen. Für diesen Fall haben viele Hersteller dieser Komponenten **Switches** bzw. **Router** mit einer **Monitorfunktion** in Ihrem Portfolio. Damit ist es möglich Netzwerddaten eines gewünschten Ports auf einen anderen zu spiegeln. Auf diesen gespiegelten Port kann man nun direkt zugreifen.

Einführung in Wireshark

Startet man das Programm, muss man zuerst eine Schnittstelle (z.B.: Ethernet) wählen, die man

abhören möchte:



Der Bildschirm, in dem die aufgezeichneten Daten bearbeitet und analysiert werden ist in 3 Bereiche aufgeteilt:

1) Paketliste

In der Paketliste, sieht man alle aufgezeichneten Frames. Folgende Spalten werden standardmäßig angezeigt:

1. No. = ist eine fortlaufende Nummerierung der Frames 2. Time = zeigt den Zeitabschnitt der Aufzeichnung an 3. Source = zeigt den Absender eines Frames an (meist die IP) 4. Destination = zeigt den Empfänger des Frames an (meist die IP) 5. Protocol = zeigt das verwendete Protokoll des Frame an 6. Info = gibt zusätzliche Informationen zum Frame bekannt

No.	Time	Source	Destination	Protocol	Length	Info
1158	214.841234	192.168.1.29	192.168.1.10	TCP	54	63336 → 443 [ACK] Seq=580
1159	215.590861	192.168.1.29	192.168.1.10	TLSv1.3	995	Application Data
1160	215.590942	192.168.1.29	192.168.1.10	TLSv1.3	155	Application Data
1161	215.591186	192.168.1.10	192.168.1.29	TCP	60	443 → 63336 [ACK] Seq=861
1162	215.984084	74.125.133.189	192.168.1.29	UDP	82	443 → 65264 Len=40
1163	216.010090	192.168.1.29	74.125.133.189	UDP	70	65264 → 443 Len=28
1164	216.202940	52.109.88.122	192.168.1.29	TLSv1.2	99	Application Data
1165	216.243318	192.168.1.29	52.109.88.122	TCP	54	63005 → 443 [ACK] Seq=71
1166	217.636990	192.168.1.29	185.199.109.153	TCP	55	[TCP Keep-Alive] 63228 →
1167	217.646662	185.199.109.153	192.168.1.29	TCP	66	[TCP Keep-Alive ACK] 443
1168	217.879712	108.177.15.189	192.168.1.29	UDP	82	443 → 59210 Len=40
1169	217.906999	192.168.1.29	108.177.15.189	UDP	70	59210 → 443 Len=28
1170	219.255630	192.168.1.20	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
1171	219.782986	192.168.1.29	52.113.194.132	TCP	55	[TCP Keep-Alive] 63339 →
1172	219.794214	52.113.194.132	192.168.1.29	TCP	66	[TCP Keep-Alive ACK] 443
1173	220.790910	192.168.1.29	173.194.76.189	UDP	65	59222 → 443 Len=23
1174	220.830983	173.194.76.189	192.168.1.29	UDP	63	443 → 59222 Len=21

2) Paketdetails

In den Paketdetails werden die OSI-Layer (Schichten) des Datenframes angezeigt. Durch anklicken des Pfeil-Symbols kann der gewählte Layer erweitert werden.

```
> Frame 1384: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{428B0
> Ethernet II, Src: Giga-Byt_4f:3a:d2 (00:1a:4d:4f:3a:d2), Dst: HewlettP_0b:2f:71 (24:be:05:0b:2f:71)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.29
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 63336, Seq: 10545, Ack: 8934, Len: 0
  Source Port: 443
  Destination Port: 63336
  [Stream index: 26]
  [TCP Segment Len: 0]
  Sequence number: 10545 (relative sequence number)
  Sequence number (raw): 1749144258
  [Next sequence number: 10545 (relative sequence number)]
  Acknowledgment number: 8934 (relative ack number)
  Acknowledgment number (raw): 2511940191
  0101 .... = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
  Window size value: 434
  [Calculated window size: 55552]
  [Window size scaling factor: 128]
  Checksum: 0x366a [unverified]
```

Natürlich variieren die Protokolldetails von Protokoll zu Protokoll. Nur die ersten 2 Schichten sind immer vorhanden.

3) Paketdaten (Hexadezimal)

Hier sind die Daten einmal im Hexadezimalsystem (links) und nebendran im Klartext (rechts) bzw. in entschlüsselter Form angezeigt.

0000	24 be 05 0b 2f 71 00 1a 4d 4f 3a d2 08 00 45 00	\$.../q... MO:... E...
0010	00 28 8b 6c 40 00 40 06 2b ec c0 a8 01 0a c0 a8	.(1@:@+.....
0020	01 1d 01 bb f7 68 68 41 d2 c2 95 b9 2a 5f 50 10hhA*_P...
0030	01 b2 36 6a 00 00 00 00 00 00 00 00 00 00 00 00	..6j.....

Filter

Nachdem praktisch ständig Netzwerkpakete gesendet und empfangen werden, ist es wichtig, dass es Möglichkeit gibt, um den Netzwerktraffic zu filtern. In dem in der Abbildung gezeigten Textfeld können beliebige Filter eingestellt werden.

The screenshot shows the Wireshark interface with a filter applied: `http && ip.addr==194.232.104.150`. Two packets are listed:

No.	Time	Source	Destination	Protocol	Length	Info
9256	704.600432	192.168.1.29	194.232.104.150	HTTP	904	GET / HTTP/1.1
9262	704.621447	194.232.104.150	192.168.1.29	HTTP	550	HTTP/1.1 301 Moved Permanent

The details pane shows the selected packet (9256) in expanded view:

```

> Frame 9256: 904 bytes on wire (7232 bits), 904 bytes captured (7232 bits) on interface \Device\NPF_{4
> Ethernet II, Src: HewlettP_0b:2f:71 (24:be:05:0b:2f:71), Dst: Netgear_3e:59:a9 (b0:7f:b9:3e:59:a9)
> Internet Protocol Version 4, Src: 192.168.1.29, Dst: 194.232.104.150
> Transmission Control Protocol, Src Port: 63465, Dst Port: 80, Seq: 1, Ack: 1, Len: 850
< Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.orf.at\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,appli
    Accept-Encoding: gzip, deflate\r\n
  
```

Im obigen Beispiel wurden nur http-Pakete mit der IP-Adresse 194.232.104.150 angezeigt. Wie man in der Abbildung erkennen kann, ist/war dies die IP-Adresse der Internetseite www.orf.at. Da http natürlich das TCP-Protokoll, IP-Protokoll & Ethernet-Protokoll nutzt, werden auch diese Daten angezeigt.

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_10

Last update: **2021/12/14 16:24**

