

1 Wireshark

1.1 Setting up Wireshark

Für Ubuntu-Users funktioniert dies recht schnell. Im Terminal (Strg+Alt+T):

```
sudo apt-get install wireshark
...
sudo dpkg-reconfigure wireshark-common
sudo adduser 'whoami' wireshark
```

Sollte weiterhin kein mitschneiden möglich sein, da der Zugriff auf /usr/bin/dumpcap nicht gestattet wird, so folgt:

```
sudo chmod +x /usr/bin/dumpcap
```

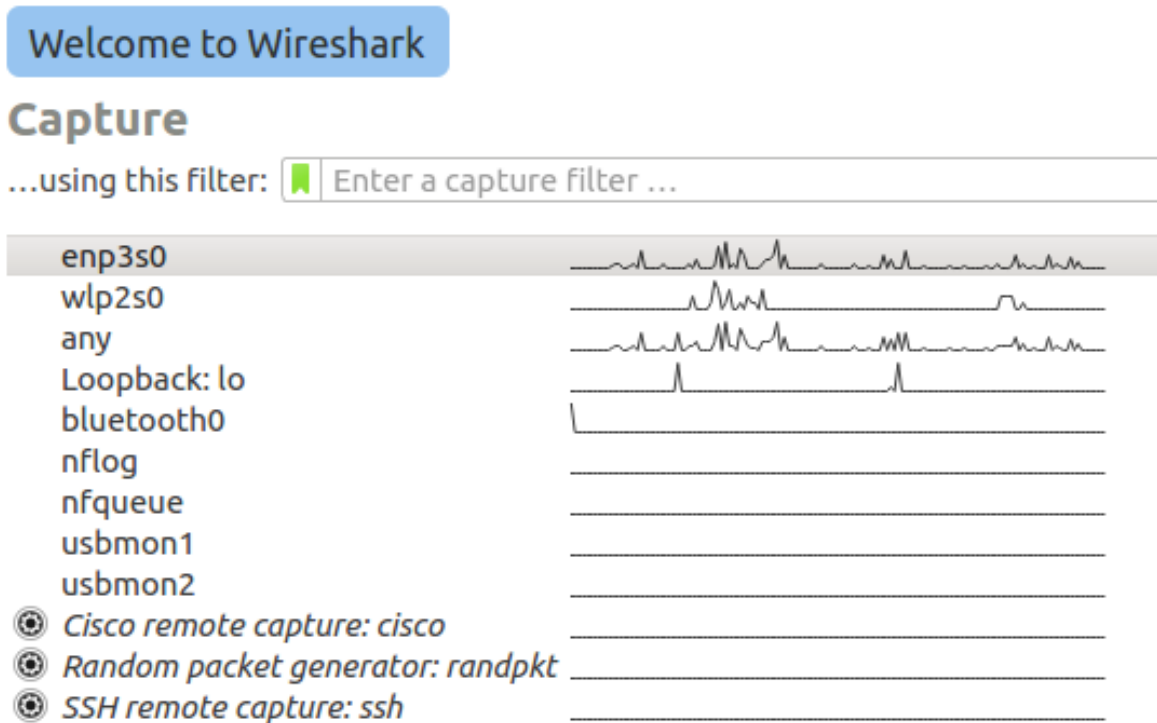
Wireshark wird anschließend über die Applikationen oder über das Terminal ausgeführt, mithilfe des Befehls

```
wireshark
```

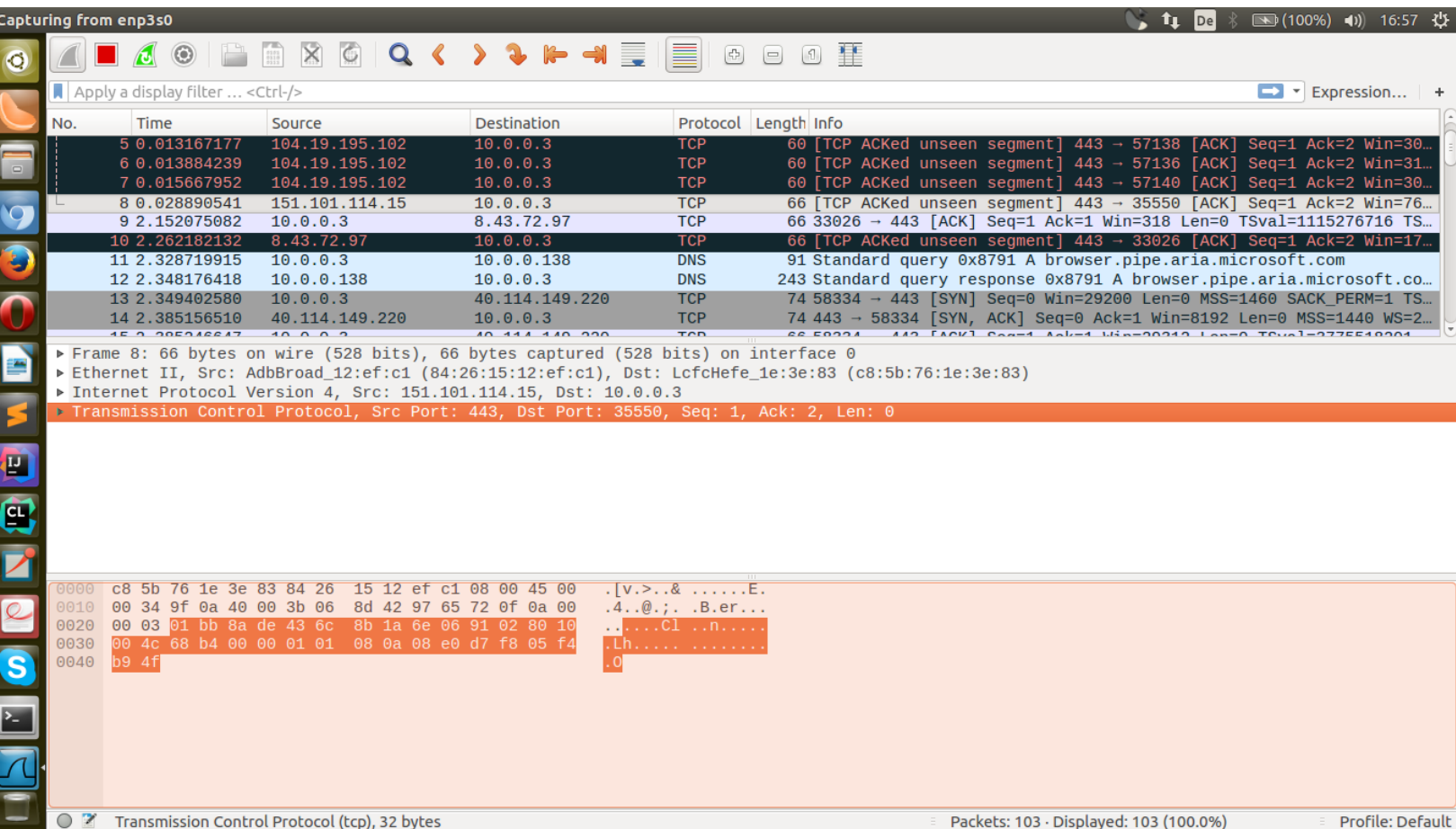
1.2 Funktionalitäten von Wireshark

1.2.1 Grundfunktionalitäten

Wireshark beobachtet stets die eingehenden Pakete des jeweiligen Interfaces. Auf dessen Hauptseite kann man bereits die Internetschnittstellen des jeweiligen Gerätes erfassen und beobachten.



Möchte man ein gewisses Interface 'sniffen', so selektiert man dieses und startet die Beobachtung.



Hier beobachtet man eine explizite Schnittstelle zum Internet, mit all dessen Datatransfer. Ist der 'promiscuous mode' aktiviert, so erfasst man alle Pakete die sich im derzeitigen Netzwerk bewegen, nicht nur des eigenen Gerätes.

Capture > Options > verify "Enable promiscuous mode on all interfaces" checkbox

Im oberen Feld sieht man die eingehenden Pakete, mitsamt

- Paketnummer (Frame), seit Start
- Dauer der Übertragung
- Ursprungs-IP des Packetes
- Ziel des Packetes
- Das genutzte Protokoll der Übertragung (DHCP, ARPA, TCP,...)
- Die Länge des Packetes in Hexadezimalziffern
- Informationen zum Packet, etwa der Request ein

Die Färbungen deuten auf die Packetformate hin. Beispielsweise ist per default TCP pink oder UDP blau markiert. Schwarz gekennzeichnete Pakete deuten auf Pakete mit Fehlern hin. Für mehr, siehe

View > Coloring Rules

Im mittleren Feld stehen konkrete Daten zum betrachteten Packet. Für gewöhnlich gliedert sich dieser Teil in

- Frame-Nummer und abgefangenen Bits
... mitsamt Daten über dieses Frame: Protocols in Frame, Marked, Ignored, ...
- Verbindungstyp der Übertragung
... inklusive Sender-IP und Empfänger-IP, oftmals in IPv6-Format sowie Übertragungsprotokoll
- und unterschiedlichen Informationen, je nach Protokollierungstyp.
... für TCP beispielsweise die Internet Protokoll Version und das Transmission Control Protokoll, inklusive deren Daten.

Im unteren Feld steht dann der Inhalt der jeweiligen Frames in non-human readable Format.

Eine weitere Interessante Funktion die WireShark bietet ist das direkte Verfolgen einer *-verbindung. Via

> Rightclick on Frame > Follow > *-Stream

Lässt sich direkt untersuchen was im Rahmen einer Verbindung ausgetauscht wird.

The screenshot displays the Wireshark network protocol analyzer interface. The top pane shows a list of captured packets. The middle pane displays the details of the selected packet (Frame 146), showing it is an Ethernet II frame from AdbBroad_12:ef:c1 to LcfcHefe_10:0:0:3, encapsulating an Internet Protocol Version 4 packet from 13.81.211.255 to 10.0.0.3, which in turn encapsulates a Transmission Control Protocol packet from port 443 to port 35386.

The bottom pane shows the raw packet data in hexadecimal and ASCII. The right pane, titled 'Wireshark · Follow TCP Stream (tcp.stream eq 22) · wireshark_enp3s0_201711...', displays the decoded data of the selected TCP stream. The data is shown in ASCII format and includes a series of HTTP requests and responses, including a 200 OK response from a Microsoft server.

No.	Time	Source	Destination	Protocol
140	82.516311753	10.0.0.3	13.81.211.255	TCP
141	82.551223936	13.81.211.255	10.0.0.3	TCP
142	82.551372730	10.0.0.3	13.81.211.255	TCP
143	82.552367895	10.0.0.3	13.81.211.255	TLSv1.2
144	82.592691507	13.81.211.255	10.0.0.3	TCP
145	82.592825188	10.0.0.3	13.81.211.255	TCP
146	82.593929720	13.81.211.255	10.0.0.3	TCP
147	82.594004423	10.0.0.3	13.81.211.255	TCP
148	82.594543432	13.81.211.255	10.0.0.3	TCP

Frame 146: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112) on interface eth0, Src: AdbBroad_12:ef:c1 (84:26:15:12:ef:c1), Dst: LcfcHefe_10:0:0:3, Src: 13.81.211.255, Dst: 10.0.0.3, Seq: 144

Entire conversation (8193 bytes) Show and save data as ASCII Stream 22

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

1.2.2 Filtering

Um etwas spezifisches zu untersuchen, bietet Wireshark Filteroptionen an. So kann man beispielsweise mit 'dns' explizit die DNS-Pakete aus dem ganzen Satz an Paketen herausfiltern. Für weitere Optionen, siehe

Analyze > Display Filters

Man kann auch eigene Filter erstellen und einbauen.¹

Übliche Befehlskonfigurationen können wie folgt lauten:

- ◊ tcp.port==80
 - Suche nach allen TCP Verbindungen die an Port 80 docken
- ◊ udp contains 64
 - Suche durch alle Pakete welche eine x64 Zahl im Frame aufweisen
- ◊ http.connection matches "Keep-Alive"
 - Suche nach allen TCP-Verbindungen mit Keep-Alive Kondition
- ◊ tcp.flags & 0x02
 - Suche durch alle TCPs mit gesetzter SYNchronize-Flag
- ◊ tcp.port in 80 443 8080
 - Suche alle TCPs welche an Port 80 oder 443 oder 8080 docken

1.2.3 Aufzeichnen von Datenverkehr

Das Aufzeichnen des Datenverkehrs erfolgt gewöhnlicherweise direkt mit dem Starten des Sniffings. Der 'promiscuous mode', wie oben bereits erwähnt, betrachtet auch Pakete die an das jeweilige Netzwerk gesendet werden.

Die Aufzeichnungen im 'promiscuous mode' sind in der Regel ungefährlich für alle Beteiligten, da Pakete nur empfangen werden wenn sie tatsächlich der Empfänger-IP entsprechen. Jedoch kann man dies mit einem SPAN port, einem Switch Port Analyzer, umgehen, wodurch jegliche Pakete durch das Gerät empfangen werden². Aufgrund der Human-Readable-Umschreibung von betrachteten Streams, kann man somit sensible Daten wie Passwörter und IDs abfangen und auslesen. Eine Gegenmaßnahmen dazu sind Enkryptionen durch SSL oder TLS oder eingeschränkte Zutrittsrechte zu Serverräumen.³ Um einen SPAN aufzusetzen benötigt man physikalischen Zugang zum jeweiligen Router.

¹https://www.wireshark.org/docs/wsug_html_chunked/ChWorkBuildDisplayFilterSection.html

²<https://blog.packet-foo.com/2016/07/how-to-use-wireshark-to-steal-passwords/>

³<https://blog.packet-foo.com/2016/11/the-network-capture-playbook-part-4-span-port-in-depth/>