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
```

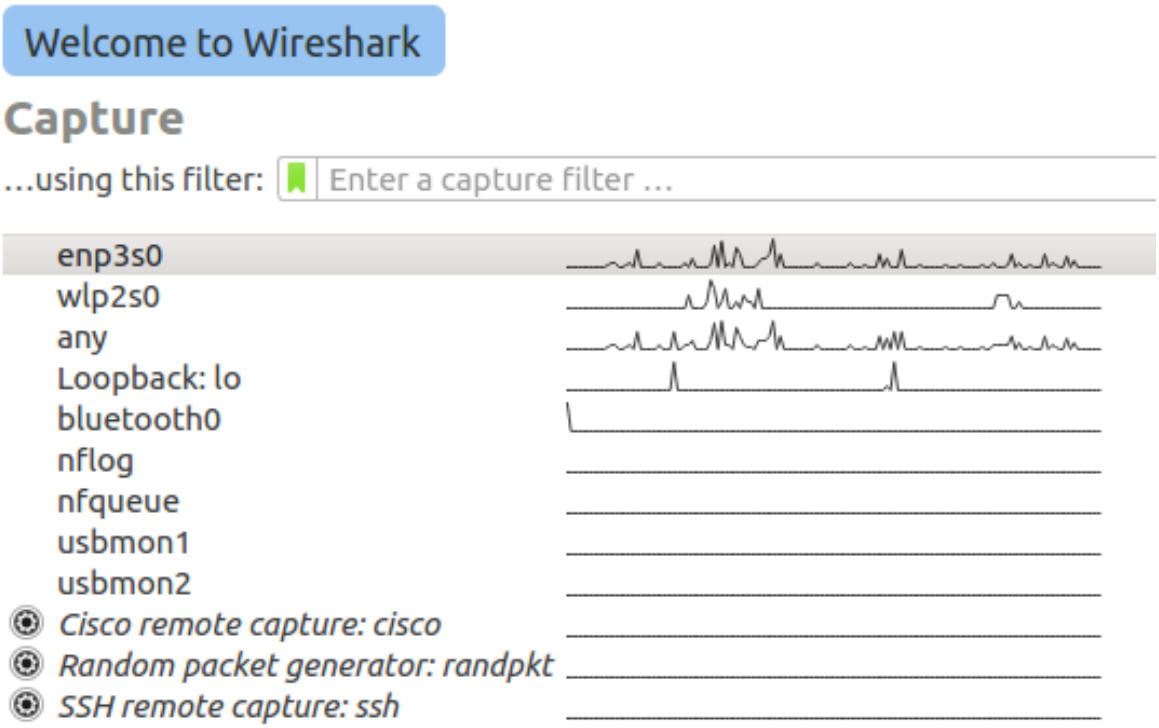
Wireshark wird anschließend über die Apllikationen oder über das Terminal ausgeführt, mithilfe des Befehls

```
wireshark
```

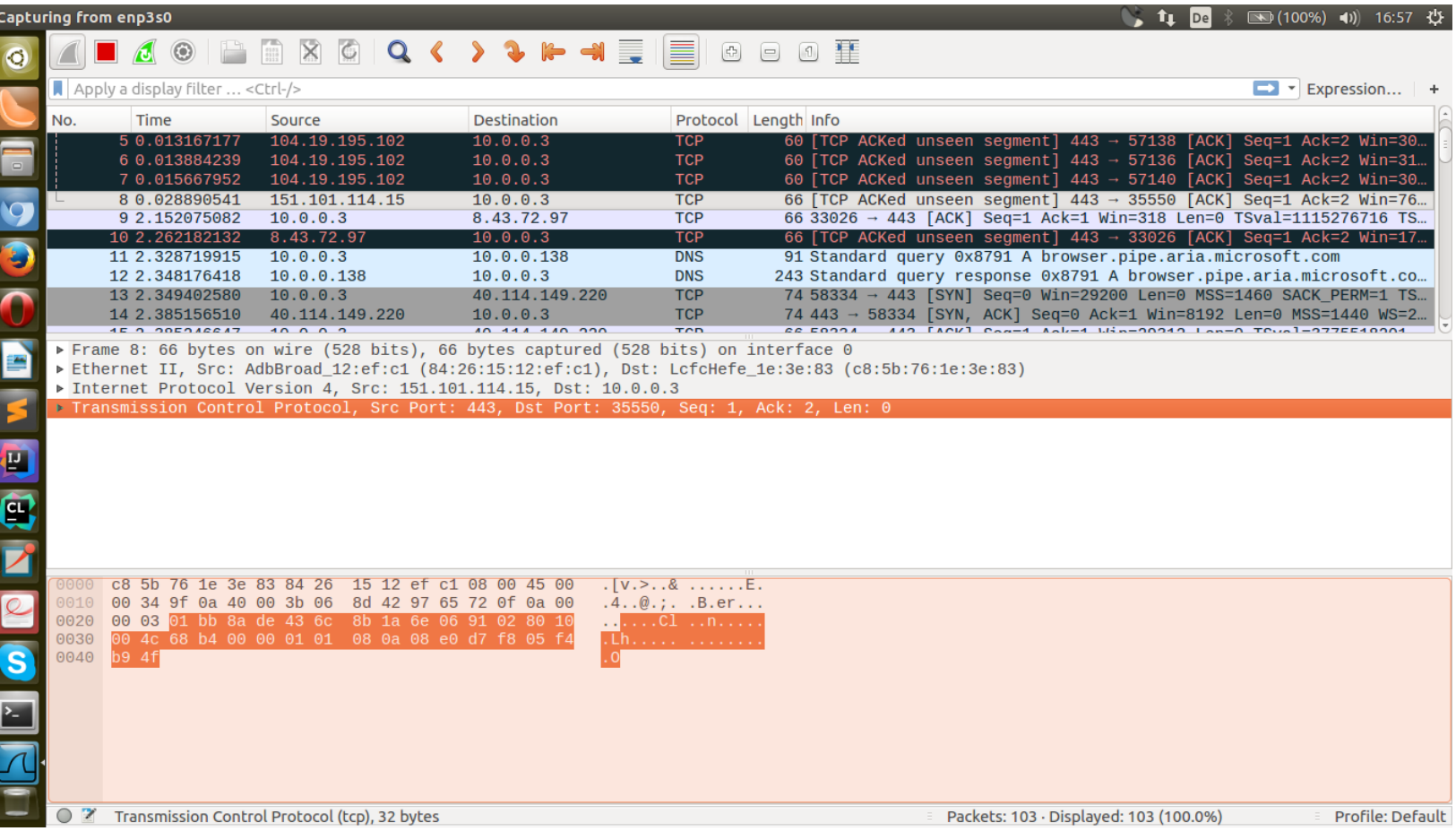
1.2 Funktionalitäten von Wireshark

1.2.1 Grundfunktionalitäten

Wireshark beobachtet stets die eingehenden Packete 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 Packete 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 Packete, mitsamt
- Packetnummer (Frame), seit Start
- Dauer der Übertragung
- Ursprungs-IP des Packetes
- Ziel des Packetes
- Das genutzte Protokol der Übertragung (DHCP, ARPA, TCP,...)
- Die Länge das 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 Packete deuten auf Packete 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

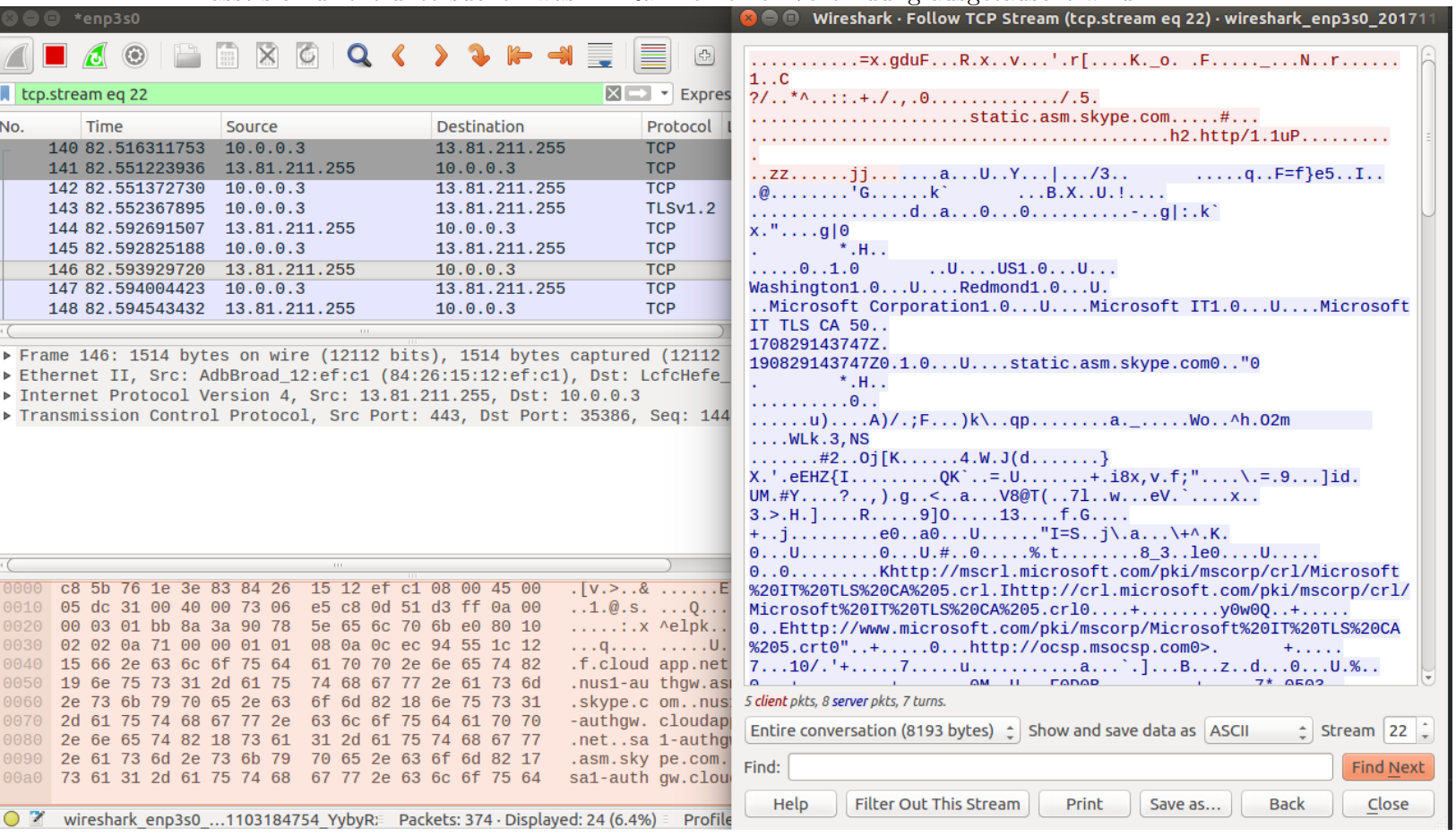
- und unterschiedlichen Informationen, je nach Protokollierungstyp.
... für TCP beispielsweise die Internet Protokol Version und das Transmission Control Protokol, inklusive deren Daten.

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.



1.2.2 Filtering

Um etwas spezifisches zu untersuchen, bietet WireShark Filteroptionen an. So kann man beispielsweise mit 'dns' explizit die DNS-Packete aus dem ganzen Satz an Packeten 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 Packete 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 Packete die an das jeweilige Netzwerk gesendet werden.

Die Aufzeichnungen im 'promiscuous mode' sind in der Regel ungefährlich für alle Beteiligten, da Packete 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 Packete durch das Gerät empfangen werden[2]. 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.

[1] *https : //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/