

Praktikum Rechnernetze

Protokoll zu Versuch 2 (Protokollanalyse mit Wireshark) von Gruppe
1

Jakob Waibel Daniel Hiller Elia Wüstner Felicitas Pojtinger

2021-10-26

Einführung

Diese Materialien basieren auf Professor Kiefers “Praktikum Rechnernetze”-Vorlesung der HdM Stuttgart.

Sie haben einen Fehler gefunden oder haben einen Verbesserungsvorschlag? Bitte eröffnen Sie ein Issue auf GitHub (github.com/pojntfx/uni-netpractice-notes):



Abbildung 1: QR-Code zum Quelltext auf GitHub

Dieses Dokument und der enthaltene Quelltext ist freie Kultur bzw. freie Software.



Abbildung 2: Badge der AGPL-3.0-Lizenz

Uni Network Practice Notes (c) 2021 Jakob Waibel, Daniel Hiller, Elia Wüstner, Felicitas Pojtinger

SPDX-License-Identifier: AGPL-3.0

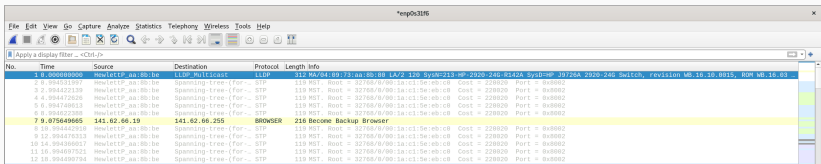
Wireshark

An welchem Koppелеlement im Systemschrank sollte der Hardware-Analysator/Netzwerk-Sniffer sinnvollerweise angeschlossen werden und warum? Welche grundsätzlichen Möglichkeiten gibt es noch?

- Switch, damit Nachrichten auf Layer 2 auch abgefangen werden können
- Grundsätzlich könnte, vor allem auch in Heimnetzwerken, der Router hierzu verwendet werden, da hier oft Router und Switch zu einem Gerät kombiniert sind.

Starten Sie Wireshark und capturen Sie den aktuellen Traffic.

Dokumentieren Sie zunächst, was alles auf Wireshark einprasselt.

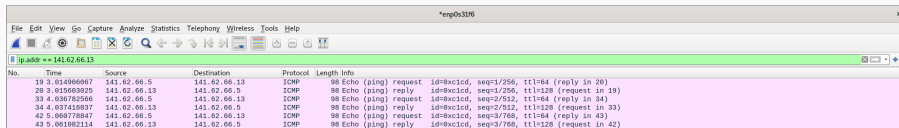


No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	hwlett@na10.be	141.62.66.255	LLDP	312	KA/01:00:13:aa:00:00 LA/2 100 SysP-213 HP/200 240 R542A SysP HP J0725A 2020 246 Switch, revision MB.10.10.0015, Rom MB.10.10.0015
2	0.00000000	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
3	0.00442139	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
4	0.00442626	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
5	0.004429633	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
6	0.004423308	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
7	0.075649605	141.62.66.19	141.62.66.255	BROWSER	216	Become Backup Browser
8	0.004426308	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
9	0.004426333	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
10	0.004426807	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
11	0.004429732	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002
12	0.004430794	hwlett@na10.be	Spawning-Tree (For...	STP	119	PST_Root = 32768/0/00-1a:c1:5e:0b:c8 Cost = 228820 Port = 0x0002

Senden Sie einen Ping zu nachfolgenden Empfängern und zeichnen Sie die entsprechenden Protokolle gezielt mit Wireshark auf. Vergleichen Sie die Protokollabläufe: wer sendet welches Protokoll warum an wen?

Pingen Sie an

Einen Rechner Ihrer Wahl im Labornetz:



The screenshot shows the Wireshark interface with a packet capture filter set to 'ip.addr == 141.62.66.13'. The packet list pane displays several ICMP Echo (ping) request and reply packets. The details pane shows the structure of an ICMP Echo (ping) request, including the type (8), code (0), and sequence number (1/256).

No.	Time	Source	Destination	Protocol	Length	Info
19	3.014906867	141.62.66.5	141.62.66.13	ICMP	98	Echo (ping) request id=8xc1cd, seq=1/256, ttl=64 (reply in 20)
20	3.015069325	141.62.66.13	141.62.66.5	ICMP	98	Echo (ping) reply id=8xc1cd, seq=1/256, ttl=128 (request in 19)
33	4.036782566	141.62.66.5	141.62.66.13	ICMP	98	Echo (ping) request id=8xc1cd, seq=2/512, ttl=64 (reply in 34)
34	4.037418837	141.62.66.13	141.62.66.5	ICMP	98	Echo (ping) reply id=8xc1cd, seq=2/512, ttl=128 (request in 33)
42	5.060778847	141.62.66.5	141.62.66.13	ICMP	98	Echo (ping) request id=8xc1cd, seq=3/768, ttl=64 (reply in 43)
43	5.061082114	141.62.66.13	141.62.66.5	ICMP	98	Echo (ping) reply id=8xc1cd, seq=3/768, ttl=128 (request in 42)

Analysieren Sie die Abläufe bei DHCP (im Labor installiert). Ihre Teilgruppe am Nachbartisch bootet den PC am Arbeitsplatz, protokollieren Sie die DHCP-Abläufe sowie sonstigen Netzverkehr, den der PC bis zum Erhalt der IP-Adresse erzeugt.

Während des Startens werden drei DHCP-Requests für verschiedene Komponenten abgehandelt.

The screenshot shows a network traffic capture window with the following columns: No., Time, Source, Destination, Protocol, and Length. The traffic includes DHCP Discover, Offer, Request, ACK, ARP, and Broadcast messages.

No.	Time	Source	Destination	Protocol	Length	Info
47	36.248724335	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x620e53eb
48	36.249844427	ogsense-router.rn.l.	255.255.255.255	DHCP	348	DHCP Offer - Transaction ID 0x620e53eb
55	48.258252423	0.0.0.0	255.255.255.255	DHCP	590	DHCP Request - Transaction ID 0x620e53eb
56	48.259518728	ogsense-router.rn.l.	255.255.255.255	DHCP	348	DHCP ACK - Transaction ID 0x620e53eb
57	48.259797973	linux.local	Broadcast	ARP	60	Who has 141.62.66.230? Tell 141.62.66.4
58	48.278416173	linux.local	Broadcast	ARP	60	Who has 141.62.66.250? Tell 141.62.66.4
63	45.478669439	fog.rnlabor.hde-stu.	linux.local	ARP	60	Who has 141.62.66.47? Tell 141.62.66.230
65	46.562657513	fog.rnlabor.hde-stu.	linux.local	ARP	60	Who has 141.62.66.47? Tell 141.62.66.230
70	47.526653895	fog.rnlabor.hde-stu.	linux.local	ARP	60	Who has 141.62.66.47? Tell 141.62.66.230
72	48.487183804	0.0.0.0	255.255.255.255	DHCP	451	DHCP Discover - Transaction ID 0xc1478931
73	48.498452675	ogsense-router.rn.l.	255.255.255.255	DHCP	348	DHCP Offer - Transaction ID 0xc1478931
79	50.529353450	0.0.0.0	255.255.255.255	DHCP	463	DHCP Request - Transaction ID 0xc1478931
80	50.531124982	ogsense-router.rn.l.	255.255.255.255	DHCP	348	DHCP ACK - Transaction ID 0xc1478931
81	50.531251215	linux.local	Broadcast	ARP	60	ARP Announcement for 141.62.66.4
82	50.504564828	linux.local	Broadcast	ARP	60	Who has 141.62.66.230? Tell 141.62.66.4
85	54.628510780	linux.local	Broadcast	ARP	60	Who has 141.62.66.230? Tell 141.62.66.4
92	66.348215769	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xad08d050
93	66.342367149	0.0.0.0	255.255.255.255	DHCP	343	DHCP Request - Transaction ID 0xad08d050
95	66.629418649	linux.local	Broadcast	ARP	60	Who has 141.62.66.250? Tell 141.62.66.4

Abbildung 9: Gesamter Bootprozess

Dokumentieren Sie den Ablauf bei einer DNS-Abfrage

Fall 1: DNS-Server 141.62.66.250:

Mittels folgendem Command wurde eine DNS-Abfrage gemacht:

```
$ dig @141.62.66.250 google.com
google.com.      163 IN  A      142.250.186.174
```



The image shows a Wireshark packet capture window with the filter 'dns && frame.number < 20'. The packet list pane shows two packets:

No.	Time	Source	Destination	Protocol	Length	Info
--	11.1.357358688	rn05.rnlabor.hdm-st	opnsense-router.rnl	DNS	93	Standard query 0xa276 A google.com OPT
--	11.1.371692076	opnsense-router.rnl	rn05.rnlabor.hdm-st	DNS	97	Standard query response 0xa276 A google.com A 142.250.186.174 OPT

Abbildung 12: Ablauf der Anfrage

Hier nutzten wir den internen DNS Server und machen eine Anfrage auf google.com.

Fall 2: DNS-Server 1.1.1.1 (Cloudflare):

Mittels folgendem Command wurde eine DNS-Abfrage gemacht:

Gelegentlich werden vom Analyzer Broadcasts erkannt. Wer sendet sie, warum und in welchen zeitlichen Abständen?

Die Broadcasts sind ARP-Requests. Sie entstehen dadurch, da Geräte versuchen Daten an andere Geräte zu übertragen, für welche sie keinen Eintrag in ihrem ARP-Cache haben, deshalb muss eine ARP-Anfrage in Form eines Broadcasts gesendet werden, da jeder Host potenziell der gesuchte Host sein kann. Dieser besitzt gesuchte IP X und antwortet daraufhin mit seiner Mac.

No.	Time	Source	Destination	Protocol	Length	Info
173	0.89017336	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
174	0.89055778	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
175	0.89055587	Linux-3.local	224.0.0.251	NDNS	82	Standard query 0x0000 PTR _pggkey-hkp._tcp.local, "qm" question
176	0.89057954	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
177	0.89056699	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
178	0.89039982	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
179	0.89080805	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
180	0.89062308	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
181	0.89050790	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
182	0.890540741	librem5-z26.rnlabo	Broadcast	ARP	60	who has 141.62.66.227 Tell 141.62.66.226
183	0.84.731177879	librem5-z26.rnlabo	Broadcast	ARP	60	who has 141.62.66.227 Tell 141.62.66.226
184	0.85.697465721	librem5-z26.rnlabo	Broadcast	ARP	60	who has 141.62.66.227 Tell 141.62.66.226
185	0.761491938	librem5-z26.rnlabo	Broadcast	ARP	60	who has 141.62.66.227 Tell 141.62.66.226
186	0.954876527	Linux-2.local	0xnsense.rnlabor.hd	DNS	86	Standard query 0x9e26 PTR 226.06.62.141.in-addr.arpa
187	0.955623099	0xnsense.rnlabor.hd	Linux-2.local	DNS	137	Standard query response 0x9e26 PTR 226.06.62.141.in-addr.arpa PTR librem5-z26.rnlabor.hd=stuttgart.de
188	0.89057954	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
189	0.86.721454740	librem5-z26.rnlabo	Broadcast	ARP	60	who has 141.62.66.207 Tell 141.62.66.226
190	0.86.785487391	librem5-z26.rnlabo	Broadcast	ARP	60	who has 141.62.66.227 Tell 141.62.66.226
191	0.89079211	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
192	0.820704980	Linux-3.local	224.0.0.251	NDNS	81	Standard query 0x0000 PTR _www-0183._tcp.local, "qm" question
193	0.890899780	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
194	0.91.067565484	Linux-2.local	0xnsense.rnlabor.hd	ARP	42	who has 141.62.66.250? Tell 141.62.66.5
195	0.91.069737280	0xnsense.rnlabor.hd	Linux-2.local	ARP	60	141.62.66.250 is at 0d:6d:b9:4f:08:14
196	0.89050790	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
197	0.885376335	hwlettP.as:8b:be	LDP_Multicast	LLDP	312	MAC/48:89:73:8a:26:88 LA2 128 SystemC13-HP-2920-24G-R12A SysID=HP-10726A 2920-24G Switch, revision W6.16.19.0015, ROM W6.16.83
198	0.89079211	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
199	0.89039982	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002
200	0.89080805	hwlettP.as:8b:be	Spanning-tree(Tor-...	STP	119	WST_Root = 32768/0/00:1a:c1:5e:0b:c8 Cost = 228820 Port = 0x8002

HTTP und TCP

Initiieren Sie eine HTTP-Sitzung (beliebige Website) und zeichnen Sie die Protokollabläufe auf

Zuerst wird ein DNS-Request getätigt. Daraufhin folgt der 3-Way-Handshake. Dieser ist an der charakteristischen Abfolge SYN, SYN-ACK, ACK zu erkennen.

No.	Time	Source	Destination	Protocol	Length	Info
714	7.550625	100.64.84.66	141.70.124.5	DNS	80	Standard query 0x189d A news.ycombinator.com
715	7.590881	100.64.84.66	141.70.124.5	DNS	80	Standard query 0x58df AAAA news.ycombinator.com
716	7.608034	141.70.124.5	100.64.84.66	DNS	158	Standard query response 0x189d A news.ycombinator.com SOA ns-225.awsdns-20.com
717	7.613971	141.70.124.5	100.64.84.66	DNS	233	Standard query response 0x189d A news.ycombinator.com A 209.216.230.240 NS ns-1411.awsdns-08.org NS ns-1914.awsdns-07.co...
718	7.614360	100.64.84.66	209.216.230.240	TCP	78	49314 → 49314 [SYN, ECN, CWI] Seq=0 Win=0SS3S Len=0 MSS=1460 WS=4 TSval=2512581059 TSecr=0 SACK_PERM=1
719	7.765218	209.216.230.240	100.64.84.66	TCP	74	443 → 49314 [SYN, ACK, ECN] Seq=0 Ack=1 Win=0SS3S Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=2045820468 TSecr=2512581059
720	7.765334	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=2512581211 TSecr=2045829468
721	7.765826	100.64.84.66	209.216.230.240	TLSv1..	583	Client Hello
722	7.917493	209.216.230.240	100.64.84.66	TLSv1..	1514	Server Hello
723	7.917494	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=1449 Ack=518 Win=65664 Len=1448 TSval=2045828612 TSecr=2512581211 [TCP segment of a reassembled PDU]
724	7.917495	209.216.230.240	100.64.84.66	TLSv1..	1062	Certificate, Certificate Status, Server Key Exchange, Server Hello Done
725	7.917581	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=518 Ack=3893 Win=127872 Len=0 TSval=2512581363 TSecr=2045828612
726	7.917726	100.64.84.66	209.216.230.240	TCP	66	[TCP Window Update] 49314 → 443 [ACK] Seq=518 Ack=3893 Win=131712 Len=0 TSval=2512581363 TSecr=2045828612
727	7.937248	100.64.84.66	209.216.230.240	TLSv1..	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
728	7.937649	100.64.84.66	209.216.230.240	TLSv1..	786	Application Data
729	8.008785	209.216.230.240	100.64.84.66	TCP	66	443 → 49314 [ACK] Seq=3893 Ack=1364 Win=64832 Len=0 TSval=2045828783 TSecr=2512581383
730	8.093869	209.216.230.240	100.64.84.66	TLSv1..	324	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
731	8.093957	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=1364 Ack=1151 Win=138752 Len=0 TSval=2512581539 TSecr=2045828788
732	8.096295	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=4151 Ack=1364 Win=65664 Len=1448 TSval=2045828789 TSecr=2512581383 [TCP segment of a reassembled PDU]
733	8.096296	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=5099 Ack=1364 Win=65664 Len=1448 TSval=2045828789 TSecr=2512581383 [TCP segment of a reassembled PDU]
734	8.096296	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=7847 Ack=1364 Win=65664 Len=1448 TSval=2045828789 TSecr=2512581383 [TCP segment of a reassembled PDU]
735	8.096297	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=8495 Ack=1364 Win=65664 Len=1448 TSval=2045828789 TSecr=2512581383 [TCP segment of a reassembled PDU]
736	8.096298	209.216.230.240	100.64.84.66	TLSv1..	681	Application Data
737	8.096317	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=1364 Ack=10558 Win=124688 Len=0 TSval=2512581542 TSecr=2045828789
738	8.096484	100.64.84.66	209.216.230.240	TCP	66	[TCP Window Update] 49314 → 443 [ACK] Seq=1364 Ack=10558 Win=131712 Len=0 TSval=2512581542 TSecr=2045828789
739	8.223332	100.64.84.66	209.216.230.240	TLSv1..	691	Application Data
740	8.223798	100.64.84.66	209.216.230.240	TCP	78	49315 → 443 [SYN, ECN, CWI] Seq=0 Win=0SS3S Len=0 MSS=1460 WS=4 TSval=3827897587 TSecr=0 SACK_PERM=1
741	8.374585	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=18558 Ack=1989 Win=65664 Len=1448 TSval=2045829070 TSecr=2512581669 [TCP segment of a reassembled PDU]
742	8.374587	209.216.230.240	100.64.84.66	TLSv1..	823	Application Data
743	8.374653	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=1989 Ack=12763 Win=128832 Len=0 TSval=2512581820 TSecr=2045829870
744	8.376801	100.64.84.66	209.216.230.240	TLSv1..	674	Application Data
745	8.378834	209.216.230.240	100.64.84.66	TCP	78	49315 → 443 [SYN, ECN, CWI] Seq=0 Win=0SS3S Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=1335760379 TSecr=3827897587
751	8.410586	100.64.84.66	209.216.230.240	TCP	66	49315 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=3827897574 TSecr=1335760379
752	8.424337	100.64.84.66	209.216.230.240	TLSv1..	585	Client Hello
753	8.527867	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=12763 Ack=2597 Win=65664 Len=1448 TSval=2045829221 TSecr=2512581821 [TCP segment of a reassembled PDU]
754	8.527868	209.216.230.240	100.64.84.66	TLSv1..	793	Application Data
761	8.527315	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=2597 Ack=14938 Win=128896 Len=0 TSval=2512581972 TSecr=2045829221
762	8.591413	209.216.230.240	100.64.84.66	TLSv1..	222	Server Hello, Change Cipher Spec, Encrypted Handshake Message
763	8.591467	100.64.84.66	209.216.230.240	TCP	66	49315 → 443 [ACK] Seq=520 Ack=157 Win=131584 Len=0 TSval=3827897926 TSecr=1335760550
764	8.591689	100.64.84.66	209.216.230.240	TLSv1..	117	Change Cipher Spec, Encrypted Handshake Message

Wie lauten die MAC-Adressen der im Labor befindlichen Ethernet-Switches? Wie haben Sie die Switches identifizieren können. Welche Möglichkeiten der Identifizierung gibt es?

Beim Spanning-Tree-Protocol lässt sich sehen, dass die Quelle der Nachrichten immer ein HP-Gerät ist. Dieses muss ein fähiges Koppelungselement des Netzwerkes sein, welches das Spanning-Tree-Protocol unterstützt. Daher wird dies mit hoher Wahrscheinlichkeit der Ethernet-Switch sein.

MAC-Adresse: 04:09:73:aa:8b:be

No.	Time	Source	Destination	Protocol	Length	Info
170	63.999710934	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
171	63.999826275	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
172	67.999494040	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
173	79.000137336	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
174	71.000050570	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
176	73.999729543	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
177	75.999566099	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
178	77.000039002	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
179	79.999889005	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
180	81.999602308	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
181	83.999531792	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
188	85.999230004	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
191	87.999702112	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
193	89.999899705	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
196	91.999034042	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
198	93.999073926	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
199	95.999704412	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
200	97.999090051	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
201	100.000218073	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
203	101.999550734	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
204	103.999772302	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
206	105.999642753	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002
212	108.000240070	HwLettP_aa:8b:be	Spanning-tree-(for...)	STP	119	MSG, Root = 32768/0/80:1a:c1:5e:0b:c0 Cost = 220020 Port = 0x8002

Auf welchen Komponenten im Netzwerk wird das Protokoll SNMP ausgeführt?

Es konnte kein SNMP-Traffic im Netzwerk gefunden werden. SNMP, das Simple Network Management Protocol, wird jedoch meist zur Wartung von verbundenen Geräte im Network verwendet, woraus sich schließen lässt, dass es auf Komponenten wie Switches, Routern oder Servern zum Einsatz kommen würde.

Protokollieren Sie den Ablauf einer TELNET-Verbindung zur IP-Adresse 141.62.66.207 (login: praktikum; passwd: versuch). Können Sie Passwörter im Wireshark-Trace identifizieren? Wie verhält sich im Vergleich dazu eine SSH-Verbindung zum gleichen Server?

Wie zu erkennen ist, wird für eine Telnet-Verbindung eine TCP-Verbindung aufgebaut. Die Passwörter sind zu erkennen.

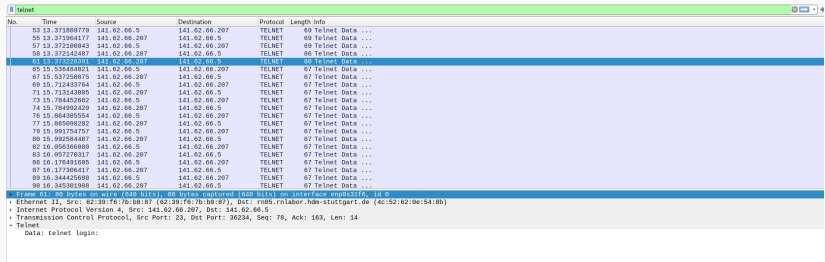
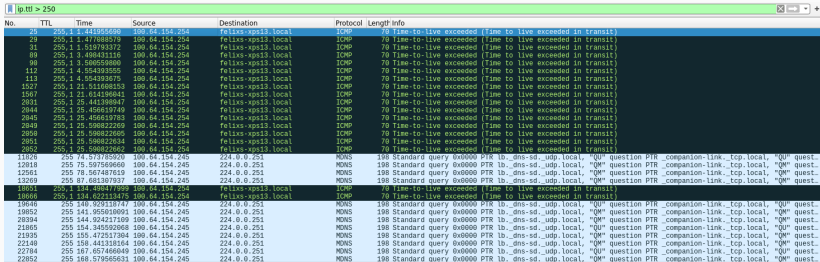


Abbildung 30: Capture des Telnet-Logins

Entwickeln, testen und dokumentieren Sie Wireshark-Filter zur Lösung folgender Aufgaben:

Nur IP-Pakete, deren TTL größer ist als ein von Ihnen sinnvoll gewählter Referenzwert



The screenshot shows the Wireshark interface with a filter 'ip.ttl > 250' applied. The packet list pane displays the following data:

No.	TTL	Time	Source	Destination	Protocol	Length	Info
25	255	1.1441955690	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
49	255	1.3171088119	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
91	255	1.1519793972	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
90	255	1.3408431116	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
99	255	1.3408659900	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
112	255	1.4354939555	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
113	255	1.4354939675	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1527	255	1.21511669153	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1507	255	1.21613196941	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2031	255	1.25441398947	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2044	255	1.25458619749	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2045	255	1.25458619783	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2049	255	1.25508822269	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2050	255	1.25508822695	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2051	255	1.25508823034	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2052	255	1.25508822662	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
11828	255	74.573785920	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
12018	255	75.597589660	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
12561	255	78.967487619	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
13269	255	87.681307937	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
18601	255	1.134490471999	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18606	255	1.134622313475	100.64.154.254	felixs-eps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
19646	255	148.929118747	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
19852	255	141.955010091	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
20304	255	144.924217109	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
21805	255	154.345592068	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
21935	255	155.472517304	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
22149	255	156.441383181	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
22784	255	167.657466049	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-
22852	255	168.579565631	100.64.154.245	224.0.0.251	NDNS	198	Standard query 0x0000 PTR 1b.dns.sd._udp.local. "00" question PTR_companion-link_tcp.local. "00" quest-

Abbildung 34: Capture der TTL-Werte ab 200

Der Linux-Kerne steuert standardmäßig die TTL auf 64; hier wurde ab 200 gefiltert, damit ausschließlich "ungewöhnliche" Pakete wie z.B.