

Praktikum Rechnernetze

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

Jakob Waibel Daniel Hiller Elia Wüstner Felix 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):



Figure 1: QR-Code zum Quelltext auf GitHub

Lizenz

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



Figure 2: Badge der AGPL-3.0-Lizenz

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

SPDX-License-Identifier: AGPL-3.0

Wireshark

Einführung

An welchem Koppelement 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 capturern Sie den aktuellen Traffic. Dokumentieren Sie zunächst, was alles auf Wireshark einprasselt.



Ping

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:



DHCP

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.

No.	Time	Source	Destination	Protocol	Length	Info
47	36.2408724335	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x620e53eb
48	36.2408844427	ognsense-router.rml...	255.255.255.255	DHCP	348	DHCP Offer - Transaction ID 0x620e53eb
55	40.2502524233	0.0.0.0	255.255.255.255	DHCP	590	DHCP Request - Transaction ID 0x620e53eb
56	40.2505187228	ognsense-router.rml...	255.255.255.255	DHCP	348	DHCP ACK - Transaction ID 0x620e53eb
57	40.2509797973	linux.local	Broadcast	ARP	60	Who has 141.62.66.236? Tell 141.62.66.4
58	40.278416173	linux.local	Broadcast	ARP	60	Who has 141.62.66.250? Tell 141.62.66.4
63	45.4786694339	fog.rnrlabor.hdm-stu...	linux.local	ARP	60	Who has 141.62.66.47 Tell 141.62.66.236
65	46.4786694339	fog.rnrlabor.hdm-stu...	1.1.1.1	ARP	60	Who has 141.62.66.47 Tell 1.1.1.1
70	47.5266538893	fog.rnrlabor.hdm-stu...	linux.local	ARP	60	Who has 141.62.66.47 Tell 141.62.66.236
72	49.4071263004	0.0.0.9	255.255.255.255	DHCP	451	DHCP Discover - Transaction ID 0xc1478931
73	49.4984526725	ognsense-router.rml...	255.255.255.255	DHCP	348	DHCP Offer - Transaction ID 0xc1478931
79	50.5293533450	0.0.0.0	255.255.255.255	DHCP	463	DHCP Request - Transaction ID 0xc1478931
80	50.531124992	ognsense-router.rml...	255.255.255.255	DHCP	348	DHCP ACK - Transaction ID 0xc1478931
81	50.531125138	linux.local	Broadcast	ARP	60	ARP Announcement for 141.62.66.4
82	50.5845464928	0.0.0.0	Broadcast	ARP	60	Who has 141.62.66.236? Tell 141.62.66.4
85	54.8205154928	linux.local	Broadcast	ARP	60	Who has 141.62.66.236? Tell 141.62.66.4
92	56.3423567409	0.0.0.0	255.255.255.255	DHCP	348	DHCP Discover - Transaction ID 0xadic98d59
93	66.3423567449	0.0.0.0	255.255.255.255	DHCP	345	DHCP Offer - Transaction ID 0xadic98d59
95	66.6292416640	linux.local	Broadcast	ARP	60	Who has 141.62.66.250? Tell 141.62.66.4

Figure 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
```

dns && frame.number < 20						
No.	Time	Source	Destination	Protocol	Length	Info
11	1.357358668	rn05.rnlab0.hdm-st	opnsense-router.rn1	DNS	93	Standard query 0xa276 A google.com OPT
12	1.371692878	opnsense-router.rn1	rn05.rnlab0.hdm-st	DNS	97	Standard query response 0xa276 A google.com A 142.250.186.174 OPT

Figure 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):

Mit folgendem Command kann die DNS-Aufgabe gelöst werden:

Lösen Sie eine ARP-Anfrage aus und protokollieren Sie die Datenpakete.

Hierzu wurde ein Rechner, welcher zuvor nicht im lokalen ARP-Cache war, neu gestartet.

No.	Time	Source	Destination	Protocol	Length	Info
214	110.515578213	Linux-2.local	Broadcast	ARP	42	who has 141.62.66.6? Tell 141.62.66.5
215	110.515867298	Linux-3.local	Linux-2.local	ARP	60	141.62.66.6 is at 4c:52:62:0e:54:2b
231	115.073154795	Linux-3.local	Linux-2.local	ARP	60	who has 141.62.66.5? Tell 141.62.66.6
232	115.073186793	Linux-2.local	Linux-3.local	ARP	42	141.62.66.6 is at 4c:52:62:0e:54:2b

Figure 15: Ablauf der Anfrage

Wann wird eine ARP-Anfrage gestartet?

Sobald ein Paket an die Zieladresse (in unserem Fall 141.62.66.6) gesendet werden soll, wird eine ARP-Anfrage in Form eines Broadcasts gestartet, um das Zielgerät im Netzwerk zu ermitteln, sofern sich diese nicht bereits im ARP-Cache befindet. Dieser kann mit ip neigh show ausgelesen werden. Mit ip neigh flush all

Layer-2-Protokolle

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.

Apply a display filter: <Ctrl>/>						
No.	Time	Source	Destination	Protocol	Length	Info
173	70.088137336	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
174	71.09953778	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
175	72.088751887	Linux-2.local	224.0.0.255	MDNS	62	Standard query 0x0008 PTR _popkey-hp._tcp.local. "Qn" question
176	72.088751887	Linux-2.local	224.0.0.255	MDNS	62	Standard query 0x0008 PTR _popkey-hp._tcp.local. "Qn" question
177	75.099556889	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
178	77.099639982	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
179	79.098888085	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
180	81.098888085	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
181	83.098537792	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
182	84.098549741	Libremes-226.rnlabor.de	Broadcast	ARP	60	who has 141.02.66.226 Tell 141.02.66.226
183	84.731177879	Libremes-226.rnlabor.de	Broadcast	ARP	60	who has 141.02.66.227 Tell 141.02.66.229
184	85.098549741	Libremes-226.rnlabor.de	Broadcast	ARP	60	who has 141.02.66.227 Tell 141.02.66.229
185	85.761495159	Libremes-226.rnlabor.de	Broadcast	ARP	60	who has 141.02.66.227 Tell 141.02.66.226
186	85.91.084876527	Linux-2.local	opensemse.rnlabor.h...	DNS	66	Standard query 0x0e2a PTR 226.66.62.141.in-addr.arpa
187	85.955623699	opensemse.rnlabor.h...	Linux-2.local	DNS	137	Standard query response 0x0e2a PTR 226.66.62.141.in-addr.arpa PTR libremes-226.rnlabor.hdm-stuttgart.de
188	86.721654749	Libremes-226.rnlabor.de	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
189	86.721654749	Libremes-226.rnlabor.de	Broadcast	ARP	60	who has 141.02.66.227 Tell 141.02.66.226
190	86.785467391	Libremes-226.rnlabor.de	Broadcast	ARP	60	who has 141.02.66.227 Tell 141.02.66.226
191	87.098791211	Linux-2.local	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
192	88.029704588	Linux-3.local	224.0.0.255	MDNS	62	Standard query 0x0008 PTR _www-0193._tcp.local. "Qn" question
193	88.029704588	Linux-3.local	224.0.0.255	MDNS	62	Standard query 0x0008 PTR _www-0193._tcp.local. "Qn" question
194	91.087505484	Linux-2.local	opensemse.rnlabor.h...	ARP	42	who has 141.02.66.259? Tell 141.02.66.5
195	91.088717289	opensemse.rnlabor.h...	Linux-2.local	ARP	60	141.02.66.259 is at 00:0c:09:48:b8:14
196	91.088717289	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
197	93.088571935	HewlettP...aa:0b:be	LLDP Multicast	LLDP	312	Ma/0:4:0973:aa:bb/Lv2/128 SysName:219-WP-2920-240 I4242 SysID:HP 3972EA 2929-240 Switch, revision W8.10.0815, ROM W8.16.03 ...
198	93.088571935	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
199	93.088571935	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002
200	93.088571935	HewlettP...aa:0b:be	Spanning-tree-(For-)	STP	119	MST. Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 228020 Port = 0x0002

MAC

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 Kopplungselement 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	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
171	85.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
172	107.9998042020	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
173	70.998137336	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
174	71.99950885778	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
178	73.999729543	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
179	77.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
178	77.9998039982	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
179	79.9998088985	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
180	81.999802380	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
181	83.999531792	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
182	85.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
183	87.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
190	91.9998034042	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
190	93.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
191	97.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
192	97.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
196	97.9995050551	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
201	100.9998216873	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
203	100.999506734	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
204	100.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
206	105.9998240873	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
212	105.9998240873	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
191	87.999791212	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
192	89.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
198	91.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
199	93.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
200	97.9998037575	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
206	105.9998240873	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002
212	105.9998240873	HewlettP...an:00:00	Spanning-tree-(For... STP	119 MST_Root = 32768/0/0/0/1a:c1:5e:eb:c0	Cost = 229020	Port 1 = 0x8002

Filtern Sie auf das Protokoll BPDU/STP. Wer sendet es und welchen Sinn hat dieses Protokoll?

Das STP-Protokoll ist das Spanning Tree Protocol. Das STP-Protokoll verhindert Schleifenbildung; dies ist besonders dann von Nutzen, wenn Redundanzen vorhanden sind. Beim STP-Protokoll werden durch alle am Netz beteiligten Switches eine "Root Bridge" gewählt und redundante Links werden deaktiviert. Wie anhand der OUI der MAC-Adresse erkannt werden kann wird dieses hier von einem HP-Switch verwendet.

No.	Time	Source	Destination	Protocol	Length Info
393	182.000315869	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
394	184.001050892	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
395	186.000287784	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
397	188.000262890	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
398	190.000313040	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
400	192.000560847	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
407	194.000671189	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
408	196.000671190	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
411	198.0006053850	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
412	200.000287784	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
413	202.0002877163	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
417	204.0002877164	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
418	206.000615952	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
423	208.000637935	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
424	210.000285871	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
425	212.0002877231	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
426	214.0010408472	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
427	216.000287784	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
429	218.000280932	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002
430	220.000546853	HuaweiTP_aa:00:be	Spanning-tree-(For-	STP	119 MST Root = 32768/0/0/0:1a:c1:5e:eb:09 Cost = 220629 Port = 0x8002

SNMP

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.

Streaming and Downloads

Starten Sie einen Download einer größeren Datei aus dem Internet und stoppen Sie ihn während der Übertragung. Dokumentieren Sie, wie der Stop-Befehl innerhalb der Protokolle umgesetzt wird



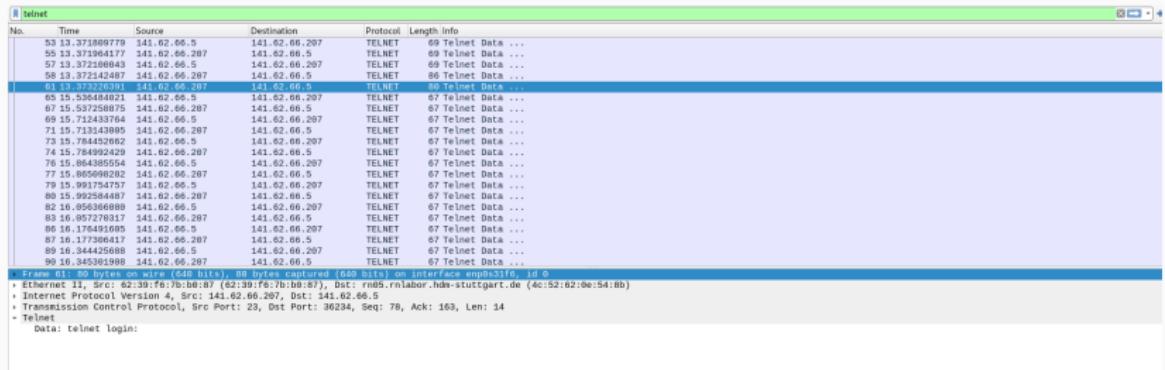
Figure 28: Capture beim Canceln des eines Downloads über HTTPS

Da der Download hier via HTTPS durchgeführt wurde, kann erkannt werden, dass die darunterliegende TCP-Verbindung unterbrochen wurde, indem die RST-Flag gesetzt wurde. Auch ein

Telnet und SSH

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.



No.	Time	Source	Destination	Protocol	Length	Info
53	13:27:38.897798	141.62.66.5	141.62.66.207	TELNET	69	Telnet Data ...
55	13:27:38.898177	141.62.66.5	141.62.66.207	TELNET	69	Telnet Data ...
57	13:37:25.000443	141.62.66.5	141.62.66.207	TELNET	69	Telnet Data ...
58	13:37:25.000487	141.62.66.207	141.62.66.5	TELNET	86	Telnet Data ...
61	13:37:25.031191	141.62.66.207	141.62.66.5	TELNET	69	Telnet Data ...
65	13:59:44.021221	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
67	13:59:44.021265	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
69	15:7124337764	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
71	15:7131438985	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
73	15:7844526862	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
74	15:7844526863	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
76	15:8643885545	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
77	15:865698282	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
79	15:991754757	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
80	15:992584487	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
82	16:05:27.790177	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
83	16:05:27.790178	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
86	16:1764916985	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
87	16:177386617	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
89	16:3444256886	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
90	16:3453098886	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...

Wireshark-Filter

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

No.	TTL	Time	Source	Destination	Protocol	Length	Info
25	255	1.1.44495569	100.64.104.254	felix-xps13.local	ICMP	68	Time-to-live exceeded (Time to live exceeded in transit)
29	255	1.14770884	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
31	255	1.1.81979337	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
89	255	1.1.49843113	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
90	255	1.1.39336909	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
112	255	1.4.85439355	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
113	255	1.4.85439375	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1527	255	1.21.51168853	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1528	255	1.21.51168854	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2031	255	1.25.44138847	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2844	255	1.25.45663749	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2845	255	1.25.45661978	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2849	255	1.25.59882226	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2869	255	1.25.59882268	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2881	255	1.25.59882234	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2882	255	1.25.59882237	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
11826	255	74.573785926	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
12018	255	75.59759660	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
12049	255	75.59759678	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
13269	255	87.12383377	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
18051	255	1.134.49847999	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18666	255	1.134.622113475	100.64.104.254	felix-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
19646	255	140.929138747	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
19852	255	145.929138748	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
23824	255	1.924237189	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
21865	255	154.345932968	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
21935	255	158.472537984	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
22148	255	158.441338164	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
22784	255	161.657466049	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.
22852	255	161.579565631	100.64.104.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QDN" question PTR_companion-link._tcp.local, "QDN" quest.

Figure 34: Capture der TTL-Werte ab 200

Der Linux-Kernel stellt standardmäßig die TTL auf 64; hier wurde