Video 13

Jochen Hollich

November 12, 2019

Abstract

Hier geht es um die Mitschrift aus Video 13 mit dem Titel "Einfach Mitschnittfilter" .

1 Mitschrift

Hier die Punkte um welche wir uns gekÃ¼mmert haben:

Grundsatz

in erster Linie stellen wir hier im GUI-Tool vomWireshark unterschiedliche Mitschnittfilter ein. diese Filter kÃ¶nnen wir unten "Aufzeichnen" ⇒ "Optionen" festlegen. Im weiteren zeige ich nur die Befehle die hier hineingeschrieben werden. Parallel werden noch auf der CLI oder der Bash Aktionen ausgefÃ¼hrt um Traffic zu triggern um diesen direkt im Wireshark aufzuzeichnen. greenmit einer Farbe hinterlegt werden.

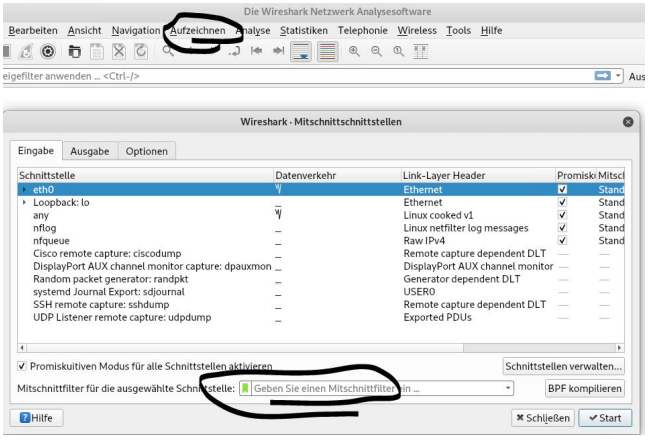

Figure 1: Bild Filter-Aufzeichnung Wire-Shark

Einfacher Mitschnittfilter auf host 8.8.8.8 mittels ping / ICMP

host 8.8.8.8

ping 8.8.8.8

hier haben wir im Wire-Shark echo request und echo reply // geht hin und her zwischen srs & destination. Somit wird nicht mehr der gesamte Traffic mitgeschnitten, sondern nur noch ein kleiner Teil, angepasst an unseren Filter.

Erweiterter Mitschnittfilter auf host 8.8.8.8 mittels ping / ICMP nur auf den SRC(source) oder auf die DST(destination)

ist er zwar nicht fettaber der Umbruch funktioniert

dst host 8.8.8.8 —— src host 8.8.8.8

ping 8.8.8.8

hier werden nur die bestimmten Hosts analysiert, und nicht mehr die beiden Kommunikationsteilnehmer

Filter auf DNS-Traffic via UDP

port 53

dig @8.8.8.8 www.cbt-24.de A

in der Konsole wir eine DNS anfrage erstellt mittels dig - dig ist sowas wie ein Frontend fÃ¼r den DNS-Service. im Wireshark sieht man die anfrage und die Response. DNS ist normalerweise immer Ã¼ber udp, da TCP erheblich aufwÃ¤ndiger ist. Das braucht ein einfache DNS Anfrage nicht. Dennoch zwinge ich im naechsten Beispiel das System einen TCP handshake bei UDP zu machen

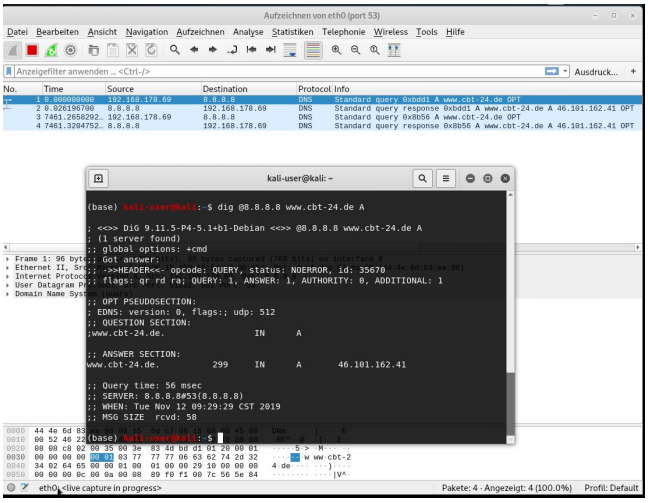

Figure 2: DNS via UDP

Filter auf DNS-Traffic via TCP

port 53

dig @8.8.8.8 www.cbt-24.de A +tcp

Hier setzten wir eine DNS-Anfrage via UDP ab, hier sind erheblich mehr tcp Handshake

1) Client = Syn
2) Server = syn, ACK
3) Client = ACK

wenn ich bei diesem Filte jetzt einen ICMP abfrage, schneidet es logischerweise nichts mit

Filter auf DNS-Traffic auf einen bestimmten port

port 53

dig @8.8.8.8 www.cbt-24.de A +tcp

heir nur Traffic via Port 53

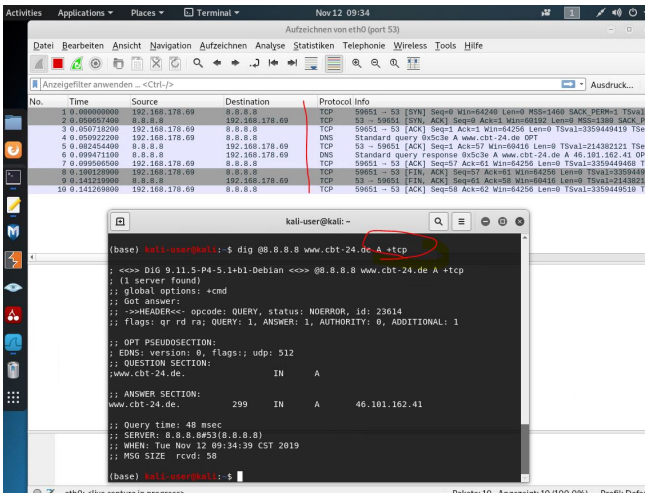

Figure 3: DNS via TCP

Filter auf DNS-Traffic auf einen bestimmten port

udp dst port 53

dig @8.8.8.8 www.cbt-24.de A +tcp

hier nur den Traffic von udp & dst

Filter auf DNS-Traffic auf einen bestimmten port mittels logischer Verknuepfungen

udp dst port 53 and dst host 8.8.8.8

dig @8.8.8.8 www.cbt-24.de A +tcp

hier die logischen Verknuepfungen, es werden nur Traffic zwischen google, via dns und udp, jetzt werden dns via udp anfragen die nicht auf 8.8.8.8 gehen nciht mitgeschnitten

Filter auf DNS-Traffic auf einen bestimmten port mittels logischer Verknuepfungen

udp dst port 53 and dst host 8.8.8.8 or dst host 192.168.178.1

dig @8.8.8.8 www.cbt-24.de A +tcp

hier werden jetzt sowohl dns traffic zwichen google und lokalem Router mitgenschnitten, aber auch alle Packete von dst host 192.168.178.1 packages ⇒ or macht einen ganz neuen Filter auf

Filter auf DNS-Traffic auf einen bestimmten port mittels logischer Verknuepfungen

udp dst port 53 and (dst host 8.8.8.8 or dst host 192.168.178.1)

dig @8.8.8.8 www.cbt-24.de A +tcp

jetzt entweder udp-Traffic von 8.8.8.8 oder on 192.168.178.1