

Zero to Zeek



© Black Hills Information Security
@BHInfoSecurity

Troy Wojewoda
Security Analyst @BHIS

> quser



Troy Wojewoda

Security Analyst | Consultant | Threat Hunter | DFIR @BHIS

Previously...

HOST
FORENSICS

MALWARE ANALYST (H|N)IDS
INCIDENT RESPONDER

THREAT HUNTER
INTELLIGENCE

NETWORK

SOC MANAGER
SECURITY
ENGINEER



© Black Hills Information Security
@BHInfoSecurity

Education/Certifications

- BS Computer Engineering & Computer Science (CNU)
- GSE, GRID, GNFA, GCFA, GCIH, GCIA, GREM, GAWN, GSEC (GOLD), CISSP

Why NSM?



- Essential for identifying and measuring risk in a computer network
- Provides a "high-ground" advantage for Defensive Operators
- Complementary to observed endpoint activity
- Provides coverage where endpoint telemetry is lacking
- **Opportunity to detect threat actor before endpoint compromise**
 - Learn from missed **activity** opportunities
 - Push further up the TA Kill Chain



What is Zeek?



- About Zeek IDS
 - Developed by Vern Paxson
 - 30+ years old
 - IDS but more...
- Meta-data all the net
- Built on frameworks
- Rule logic constructs

“Bro is not strictly an intrusion detection system that generates alerts, like Snort. Rather, Bro generates a range of NSM data, including session data, transaction data, extracted content data, statistical data, and even alerts -- if you want them.”

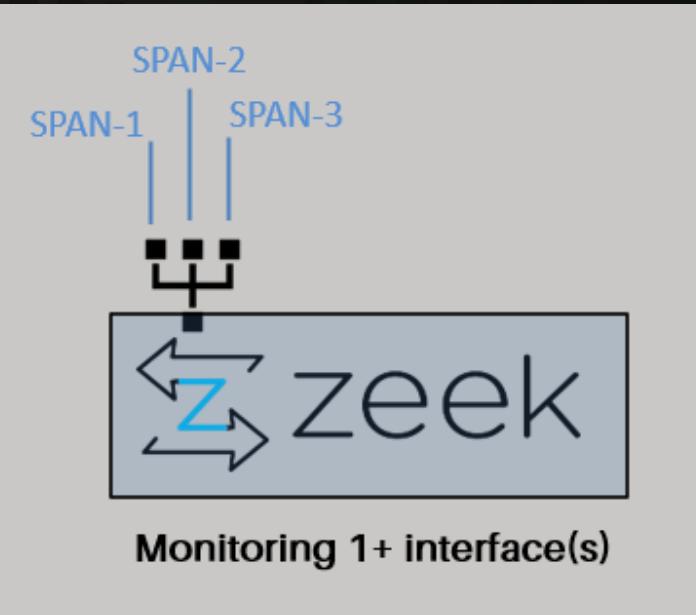
- Richard Bejtlich, TaoSecurity



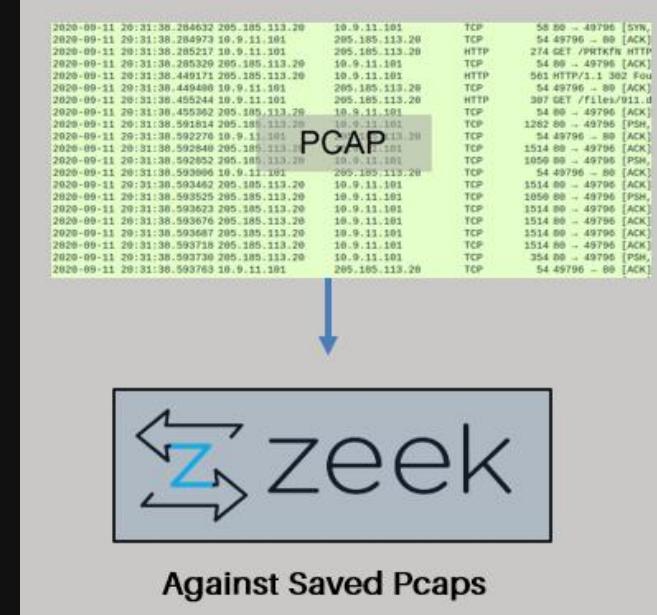
Modes of Operation



Continuous Monitoring



Processing PCAPs



Step 0

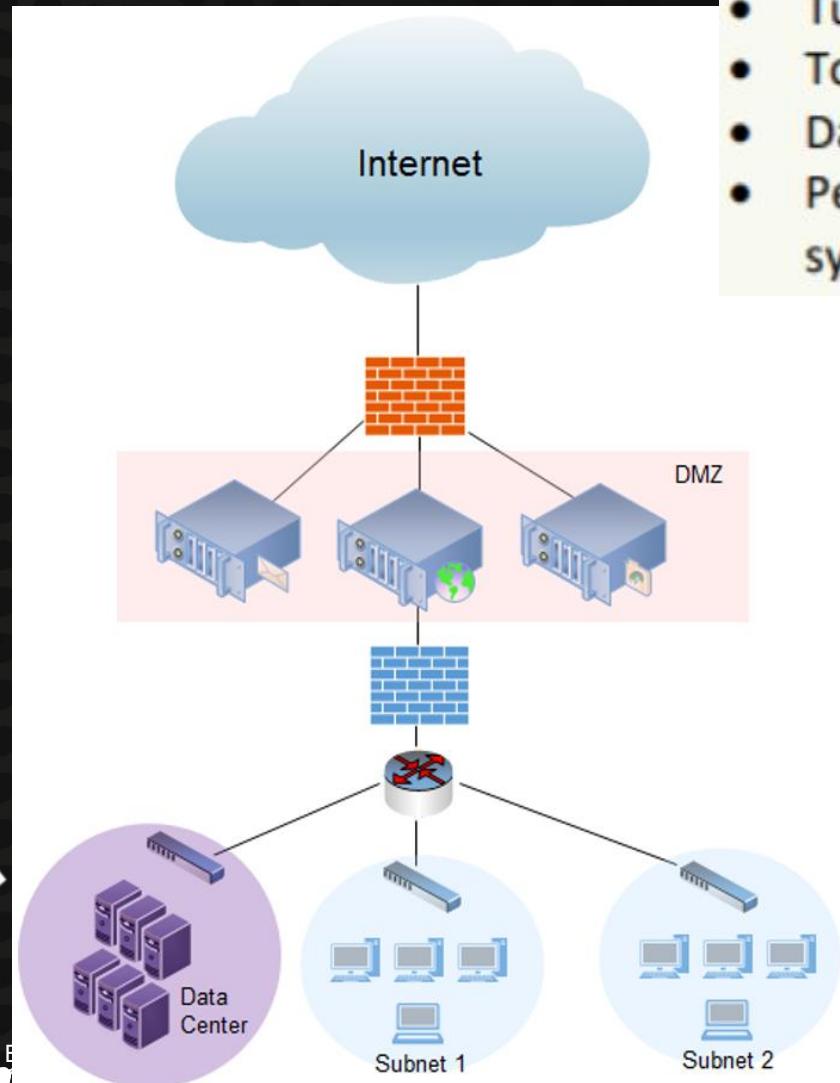


Prep Time: $-\backslash(\ツ)_/-$

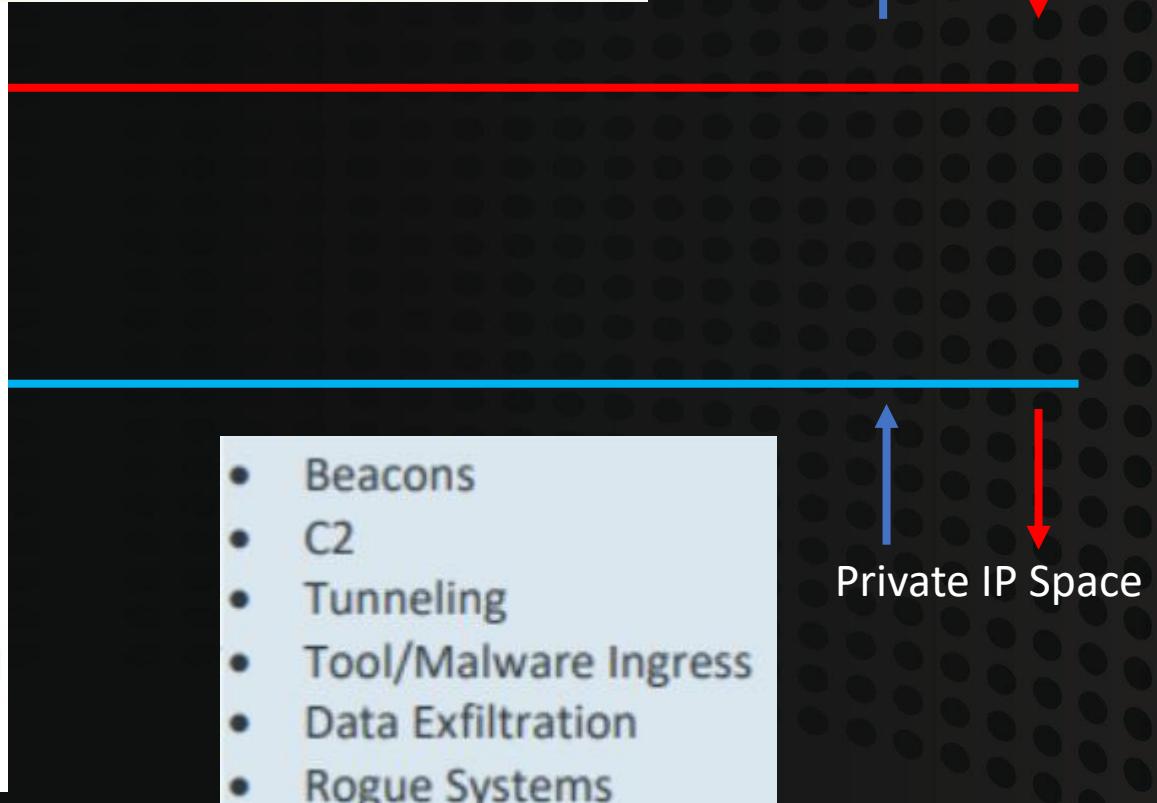
Install Time: $\frac{1}{-\backslash(\ツ)_/-}$



NSM Placement



- Perimeter Device Compromise
- C2
- Tunneling
- Tool/Malware Ingress
- Data Exfiltration
- Perimeter Scanning for exposed systems and services



Step 0



- Acquire Hardware*
- Install OS
- Identify Monitoring Ports (Network Interfaces)
 - Helpful commands
- Span/Tap Traffic (Placement)



ip a
ip -h -s link show [int_name]
ethtool -S [int_name]



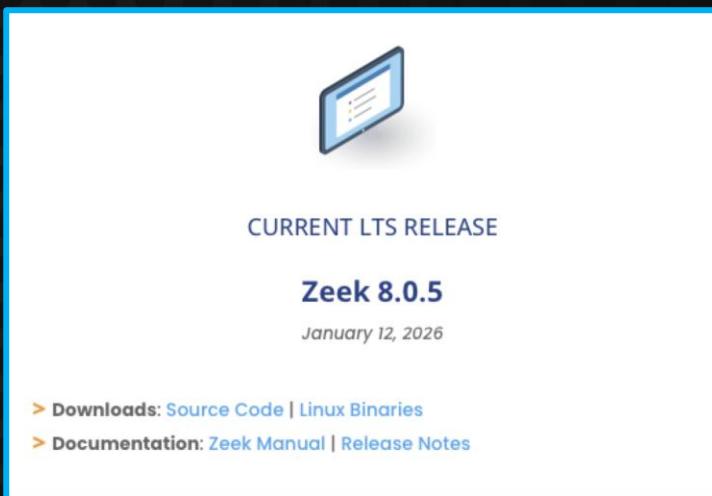
Step 1



Step 1

Decide the path you want to take:

1. Docker
 - o <https://github.com/activecm/docker-zeek>
2. Source Code | Linux Binaries
 - o <https://zeek.org/get-zeek/>



Quickstart

You'll first need Docker. If you don't already have it here is a quick and dirty way to install it on Linux:

```
curl -fsSL https://get.docker.com | sh -
```

Otherwise, follow the [install instructions](#) for your operating system.

You can then use the `zeek` script in this repo to quickly get Zeek running. We recommend putting this `zeek` script in your system `PATH`. The rest of this readme will assume this repo's `zeek` script is in the system `PATH`.

```
sudo wget -O /usr/local/bin/zeek https://raw.githubusercontent.com/activecm/docker-zeek/main/zeek
sudo chmod +x /usr/local/bin/zeek
```

Then use the script to start Zeek.

```
zeek start
```



Step 1.a



<https://github.com/activecm/docker-zeek>

```
$ curl -fsSL https://get.docker.com | sh -  
  
$ sudo wget -O /usr/local/bin/zeek  
https://raw.githubusercontent.com/activecm/docker-zeek/master/zeek  
  
$ sudo chmod +x /usr/local/bin/zeek  
  
$ zeek start
```

Quickstart

You'll first need Docker. If you don't already have it here is a quick and dirty way to install it on Linux:

```
curl -fsSL https://get.docker.com | sh -
```

Otherwise, follow the [install instructions](#) for your operating system.

You can then use the `zeek` script in this repo to quickly get Zeek running. We recommend putting this `zeek` script in your system `PATH`. The rest of this readme will assume this repo's `zeek` script is in the system `PATH`.

```
sudo wget -O /usr/local/bin/zeek https://raw.githubusercontent.com/activecm/docker-zeek/master/zeek  
sudo chmod +x /usr/local/bin/zeek
```

Then use the script to start Zeek.

```
zeek start
```



Step 1.b



A screenshot of the Zeek 8.0.5 LTS release page. The page features a blue header with the text "CURRENT LTS RELEASE" and a small icon of a computer monitor. Below the header, the text "Zeek 8.0.5" is prominently displayed, followed by the date "January 12, 2026". At the bottom of the page, there are two links: "Downloads: Source Code | Linux Binaries" and "Documentation: Zeek Manual | Release Notes".



<https://github.com/zeek/zeek/wiki/Binary-Packages>

- Debian
- Fedora
- OpenSUSE
- Raspbian
- Ubuntu



Step 2



Quickstart

You'll first need Docker. If you don't already have it here is a quick and dirty way to install it on Linux:

```
curl -fsSL https://get.docker.com | sh -
```

```
$ curl -fsSL https://get.docker.com | sh -
```



```
$ sudo wget -O /usr/local/bin/zeek https://raw.githubusercontent.com/activecm/docker-zeek/master/zeek
```



```
$ sudo chmod +x /usr/local/bin/zeek
```



```
$ zeek start
```

g. We recommend putting this zeek
's zeek script is in the system PATH .

https://raw.githubusercontent.com/activecm/docker-zeek/master/zeek



Step 3 Zeek (install)



```
zuser@zuser-NUC11TNHi7:~$ zeek start
Could not find /opt/zeek/etc/node.cfg. Generating one now.
? Choose your capture interface(s): [Use arrows to move, space to select, type to filter,
? for more help]
> [ ] docker0          UP      172.17.0.1
  [x] enp88s0          UP      -
  [ ] enp89s0          UP      192.168.0.111  fe80::4a21:bf:fe5f:df4b
  [ ] lo              UP      127.0.0.1  ::1
```



Reviewing the Script



```
root@localhost:/opt/z  
/usr/local/bin/zeek  
root@localhost:/opt/z  
/usr/local/bin/zeek:
```

It's a script!

very long lines (534)



Reviewing the Script

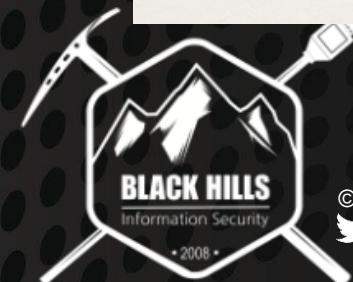
```
35 HOST_ZEEK=${zeek_top_dir:-/opt/zeek}
36 IMAGE_NAME="activecm/zeek:${zeek_release:-latest}"
37
38 # initializes Zeek directories and config files on the host
39 ▼ init_zeek_cfg() {
40     # create a temporary container to run commands
41     local container="zeek-init-$RANDOM"
42     ▼ $SUDO docker run \
43         --ulimit nofile=1048576:1048576 \
44         --detach \
45         --name $container \
46         -v "$HOST_ZEEK":/zeek \
47         --network host \
48         "$IMAGE_NAME" \
49         sh -c 'while sleep 1; do :; done' >/dev/null 2>&1
50     # ensure the temporary container is removed
51     trap "$SUDO docker rm --force $container >/dev/null 2>&1" EXIT
52
53     # run commands using $SUDO docker to avoid unnecessary sudo calls
54     # create directories required for running Zeek
55     ▼ $SUDO docker exec $container mkdir -p \
56         "/zeek/manual-logs" \
57         "/zeek/logs" \
58         "/zeek/spool" \
59         "/zeek/etc" \
60         "/zeek/share/zeek/site/autoload" 2>/dev/null \
61         || true # suppress error code if symlink exists
```



Customizing...



```
root@localhost:/opt/zeek/share/zeek/site/autoload# ls -1
001-unload-scripts.zeek
100-default.zeek
200-inactivity_timeout.zeek
900-zkg.zeek
```



Customizing...moar



Install a Plugin

You can install Zeek packages from <https://packages.zeek.org/> using the Zeek Package Manager, `zkg`. For example, to install the `hassh` plugin:

```
# Run `zeek start` if you haven't already
docker exec -it zeek zkg install hassh
# Restart Zeek to activate plugin
zeek restart
```



Note: Currently only plugins that don't require compiling can be installed.

```
root@localhost:/opt/zeek/spool/manager# docker volume ls
DRIVER      VOLUME NAME
local        zeek-zkg-plugin
local        zeek-zkg-script
local        zeek-zkg-state
```



Customizing...moarrr



```
@load policy/tuning/json-logs.zeek
```

```
event zeek_init() {
    Log::disable_stream(Syslog::LOG);
}
```



Cleanup

```
$ sudo curl -o /usr/local/bin/zeek_log_clean.sh \
  https://raw.githubusercontent.com/activecm/zeek-log-
clean/main/zeek_log_clean.sh

$ sudo chmod +x /usr/local/bin/zeek_log_clean.sh

$ echo "* * * * * root flock -n /tmp/zeek-log-clean
/usr/local/bin/zeek_log_clean.sh" \
  | sudo tee /etc/cron.d/zeek-log-clean
```



Docker-Zeek Recap

- **/usr/local/bin/zeek** – Docker-Zeek handler script (just type **zeek** as a shortcut)
 - **zeek restart** – Restarts the docker container
- **/opt/zeek/** – Zeek's top level directory
- **/opt/zeek/etc/** - Zeek sensor configuration
 - /opt/zeek/etc/node.cfg contains the entry for which interface Zeek will attach its network processing from (interface=af_packet::eth1)
- **/opt/zeek/logs/** - Contains Zeek's archived logs, where logs older than one hour are rotated to.
 - For example: /opt/zeek/logs/2025-10-31/ will contain all the logs for October 31, 2025.
- **/opt/zeek/spool/manager/** - Contains the current hour logs (log roll to /opt/zeek/logs/ at the top of each hour)
- **/opt/zeek/share/zeek/site/autoload/** - Contains additional Zeek configurations that support persistent customizations (i.e., will reapply when the container restarts).
 - For example, this is where we configured the logs to be in JSON format and disabled syslog.log logging by zeek



Docker-Zeek Recap



- **/usr/local/bin/zeek_log_clean.sh** – Runs every 60s to ensure no more than 90% disk utilization
- **# docker exec -it zeek /bin/bash** – Enter docker-zeek container in an interactive bash shell
- **# zeek restart** – Restart docker-zeek container
- **# zeek start** – Start docker-zeek container
- **# zeek stop** – Stop docker-zeek container
- **# zeek update** – Update to the latest docker-zeek container instance



Story Time

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

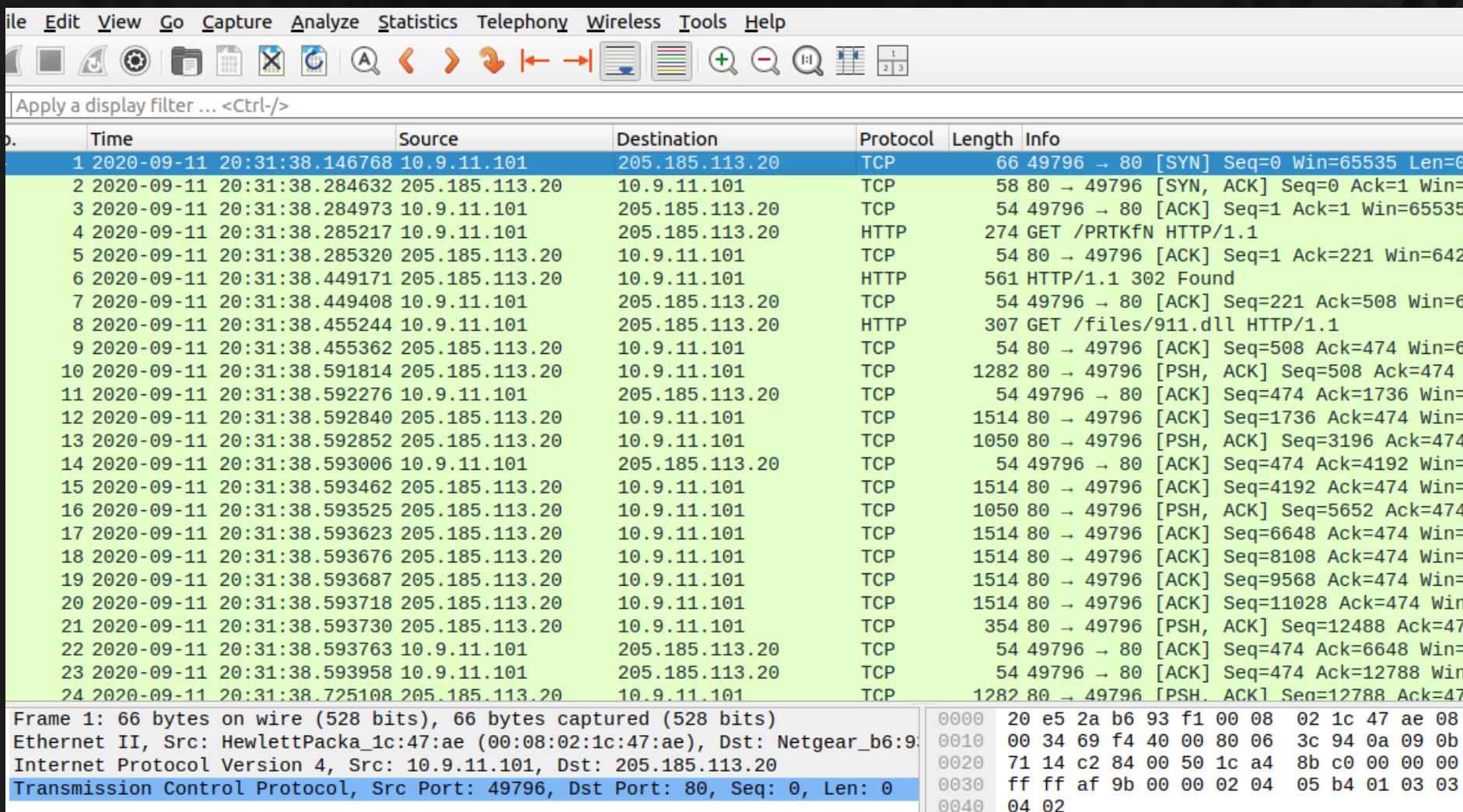
Time	Source	Destination	Protocol	Length	Info
1 2020-09-11 20:31:38.146768	10.9.11.101	205.185.113.20	TCP	66	49796 → 80 [SYN] Seq=0 Win=65535
2 2020-09-11 20:31:38.284973	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=1 Ack=1 Win=65535
3 2020-09-11 20:31:38.285217	10.9.11.101	205.185.113.20	HTTP	274	GET /PRTKfn HTTP/1.1
4 2020-09-11 20:31:38.449408	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=221 Ack=508
5 2020-09-11 20:31:38.455244	10.9.11.101	205.185.113.20	HTTP	307	GET /files/911.dll HTTP/1.1
6 2020-09-11 20:31:38.592276	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=173
7 2020-09-11 20:31:38.593006	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=419
8 2020-09-11 20:31:38.593763	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=664
9 2020-09-11 20:31:38.593958	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=127
10 2020-09-11 20:31:38.725644	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=140
11 2020-09-11 20:31:38.725918	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=189
12 2020-09-11 20:31:38.726101	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=226
13 2020-09-11 20:31:38.726608	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=312
14 2020-09-11 20:31:38.726707	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=324
15 2020-09-11 20:31:38.727390	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=3489
16 2020-09-11 20:31:38.739511	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=361
17 2020-09-11 20:31:38.739728	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=373
18 2020-09-11 20:31:38.861342	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=422
19 2020-09-11 20:31:38.878129	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=447

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0
Ethernet II, Src: HewlettPacka_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:93:11 (00:0c:00:b6:93:11)
Internet Protocol Version 4, Src: 10.9.11.101, Dst: 205.185.113.20
Transmission Control Protocol, Src Port: 49796, Dst Port: 80, Seq: 0, Len: 0

Hex	Dec
0000	20 e5 2a b6 93 f1 00 08 02 1c 47
0010	00 34 69 f4 40 00 80 06 3c 94 0a
0020	71 14 c2 84 00 50 1c a4 8b c0 00
0030	ff ff af 9b 00 00 02 04 05 b4 01
0040	04 02



Story Time



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2020-09-11 20:31:38.146768	10.9.11.101	205.185.113.20	TCP	66	49796 → 80 [SYN] Seq=0 Win=65535 Len=0
2	2020-09-11 20:31:38.284632	205.185.113.20	10.9.11.101	TCP	58	80 → 49796 [SYN, ACK] Seq=0 Ack=1 Win=65535
3	2020-09-11 20:31:38.284973	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=1 Ack=1 Win=65535
4	2020-09-11 20:31:38.285217	10.9.11.101	205.185.113.20	HTTP	274	GET /PRTKfn HTTP/1.1
5	2020-09-11 20:31:38.285320	205.185.113.20	10.9.11.101	TCP	54	80 → 49796 [ACK] Seq=1 Ack=221 Win=6424
6	2020-09-11 20:31:38.449171	205.185.113.20	10.9.11.101	HTTP	561	HTTP/1.1 302 Found
7	2020-09-11 20:31:38.449408	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=221 Ack=508 Win=65535
8	2020-09-11 20:31:38.455244	10.9.11.101	205.185.113.20	HTTP	307	GET /files/911.dll HTTP/1.1
9	2020-09-11 20:31:38.455362	205.185.113.20	10.9.11.101	TCP	54	80 → 49796 [ACK] Seq=508 Ack=474 Win=6424
10	2020-09-11 20:31:38.591814	205.185.113.20	10.9.11.101	TCP	1282	80 → 49796 [PSH, ACK] Seq=508 Ack=474 Win=6424
11	2020-09-11 20:31:38.592276	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=1736 Win=6424
12	2020-09-11 20:31:38.592840	205.185.113.20	10.9.11.101	TCP	1514	80 → 49796 [ACK] Seq=1736 Ack=474 Win=6424
13	2020-09-11 20:31:38.592852	205.185.113.20	10.9.11.101	TCP	1050	80 → 49796 [PSH, ACK] Seq=3196 Ack=474 Win=6424
14	2020-09-11 20:31:38.593006	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=4192 Win=6424
15	2020-09-11 20:31:38.593462	205.185.113.20	10.9.11.101	TCP	1514	80 → 49796 [ACK] Seq=4192 Ack=474 Win=6424
16	2020-09-11 20:31:38.593525	205.185.113.20	10.9.11.101	TCP	1050	80 → 49796 [PSH, ACK] Seq=5652 Ack=474 Win=6424
17	2020-09-11 20:31:38.593623	205.185.113.20	10.9.11.101	TCP	1514	80 → 49796 [ACK] Seq=6648 Ack=474 Win=6424
18	2020-09-11 20:31:38.593676	205.185.113.20	10.9.11.101	TCP	1514	80 → 49796 [ACK] Seq=8108 Ack=474 Win=6424
19	2020-09-11 20:31:38.593687	205.185.113.20	10.9.11.101	TCP	1514	80 → 49796 [ACK] Seq=9568 Ack=474 Win=6424
20	2020-09-11 20:31:38.593718	205.185.113.20	10.9.11.101	TCP	1514	80 → 49796 [ACK] Seq=11028 Ack=474 Win=6424
21	2020-09-11 20:31:38.593730	205.185.113.20	10.9.11.101	TCP	354	80 → 49796 [PSH, ACK] Seq=12488 Ack=474 Win=6424
22	2020-09-11 20:31:38.593763	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=6648 Win=6424
23	2020-09-11 20:31:38.593958	10.9.11.101	205.185.113.20	TCP	54	49796 → 80 [ACK] Seq=474 Ack=12788 Win=6424
24	2020-09-11 20:31:38.725108	205.185.113.20	10.9.11.101	TCP	1282	80 → 49796 [PSH, ACK] Seq=12788 Ack=474 Win=6424

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: HewlettPacka_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:9f (00:09:00:00:b6:9f)
Internet Protocol Version 4, Src: 10.9.11.101, Dst: 205.185.113.20
Transmission Control Protocol, Src Port: 49796, Dst Port: 80, Seq: 0, Len: 0

0000	20 e5 2a b6 93 f1 00 08	02 1c 47 ae 08
0010	00 34 69 f4 40 00 80 06	3c 94 0a 09 0b
0020	71 14 c2 84 00 50 1c a4	8b c0 00 00 00
0030	ff ff af 9b 00 00 02 04	05 b4 01 03 03
0040	04 02	



!(src/dst) IP



!(Source/Destination) - instead: Originator/Responder

		id.orig_h		id.resp_h						
1599856316.795735	C4hwc83kYBVlkBkb1	10.9.11.101	55080	10.9.11.1	53	udp	dns	0.329710		
1599856316.864111	CGje492TWZQlwJ1aQk	10.9.11.101	56241	10.9.11.1	53	udp	dns	0.152311		
1599856321.780000	CYRrsD3lSZ1fAansVb	10.9.11.101	49810	52.114.128.69	443	tcp	ssl	0.602856		
1599856317.457680	CeRAwk3sI8BlbXZQJ7	10.9.11.101	52454	10.9.11.1	53	udp	dns	0.094578		
1599856318.919963	CnQvb02n4DqFRuS8Ie	10.9.11.101	49809	65.52.108.90	443	tcp	ssl	4.079217		
1599856318.446434	CLz0R43VB8DKCoDcx4	10.9.11.101	56743	10.9.11.1	53	udp	dns	0.059654		
1599856318.850775	CkMGwq2stJPrYaf401	10.9.11.101	55832	10.9.11.1	53	udp	dns	0.067695		
1599856324.270197	Crybsd31Hc2CkZNph	10.9.11.101	49811	52.114.128.69	443	tcp	ssl	1.003278		
1599856317.033366	CVxxoUMaa34zATyA1	10.9.11.101	49802	23.219.226.88	443	tcp	ssl	10.014538		
1599856317.033425	CdoZ3J1d05T3Epfl4d	10.9.11.101	49801	23.219.226.88	443	tcp	ssl	10.014605		
1599856326.615444	CNkyNJdxym9006jd3	10.9.11.101	49812	52.114.128.69	443	tcp	ssl	0.645213		
1599856324.135401	CaJce23eCmrRd5FHq1	10.9.11.101	59241	10.9.11.1	53	udp	dns	0.132686		
1599856347.864813	CJaGJ03bJ1hqGWIrNg	10.9.11.101	49813	52.109.4.5	443	tcp	ssl	0.774984		
1599856347.668890	CRq74L2liayk0ad0gd	10.9.11.101	52325	10.9.11.1	53	udp	dns	0.195472		
1599856366.144543	CqgYkUEvNeaSM07mj	10.9.11.101	62705	10.9.11.1	53	udp	dns	3.015339		
1599856371.706935	CkV00egcpHJMHLCo	10.9.11.101	63404	10.9.11.1	53	udp	dns	0.119682		
1599856373.024140	CTywoc3eJSR1Ba8alf	10.9.11.101	54510	10.9.11.1	53	udp	dns	0.075406		
1599856373.548449	CZUpta2tmB4gt02q3e	10.9.11.101	60917	10.9.11.1	53	udp	dns	0.067894		
1599856324.713508	C3SYmFzwuEr6zNl18	10.9.11.101	138	10.9.11.255	138	udp	-	-		
1599856383.607662	CjrmZjyaFKiWQDXF3	10.9.11.101	49820	31.184.253.244	80	tcp	http	0.523879		
1599856384.139041	CSZ0mUG7iYY7AMQgi	10.9.11.101	49821	31.184.253.244	80	tcp	http	1.566584		
1599856386.025612	CPLDdKHWgij9LhKf3	10.9.11.101	49826	31.184.253.244	80	tcp	http	1.009726		

Resources



- <https://zeek.org/>
- <https://github.com/activecm/docker-zeek>
- <https://github.com/activecm/zeek-log-clean>
- <https://corelight.com/hubfs/resources/zeek-cheatsheets/corelight-cheatsheet-poster.pdf>
- <https://www.blackhillsinfosec.com/introduction-to-zeek-log-analysis-wrap/>

<https://www.youtube.com/watch?v=a2Cp6VYQuvU>



Upcoming Training!



Foundations of Network Forensics and Analysis (4h – PFWYC)

- Workshop: January 30, 2026 (Remote)

<https://www.antisyphontraining.com/product/workshop-foundations-of-network-forensics-and-analysis-with-troy-wojewoda/>

Network Forensics & Incident Response (2Day Course)

- WWHF @Mile High: February 10 & 11 2026 (Denver, CO/Remote)

<https://wildwesthackinfest.com/wild-west-hackin-fest-mile-high-2026-pre-con/>

- SOC Summit: March 30 & 31 (Remote)

<https://www.antisyphontraining.com/product/network-forensics-and-incident-response-with-troy-wojewoda/>



© Black Hills Information Security
@BHInfoSecurity



- Black Hills Information Security
 - <http://www.blackhillsinfosec.com>
 - @BHInfoSecurity
- Troy Wojewoda
 - @wojeb blaze
 - <https://www.linkedin.com/in/troy-wojewoda-92387183>

