

Wireshark Network Traffic Capture and Analysis Report - Kali Linux

1. Introduction

This report provides a step-by-step guide on capturing and analyzing network traffic using Wireshark on Kali Linux. Wireshark is a widely-used network protocol analyzer that allows you to monitor and inspect packets in real time. This document covers installation, capture, filtering, and basic analysis techniques.

2. Installation of Wireshark on Kali Linux

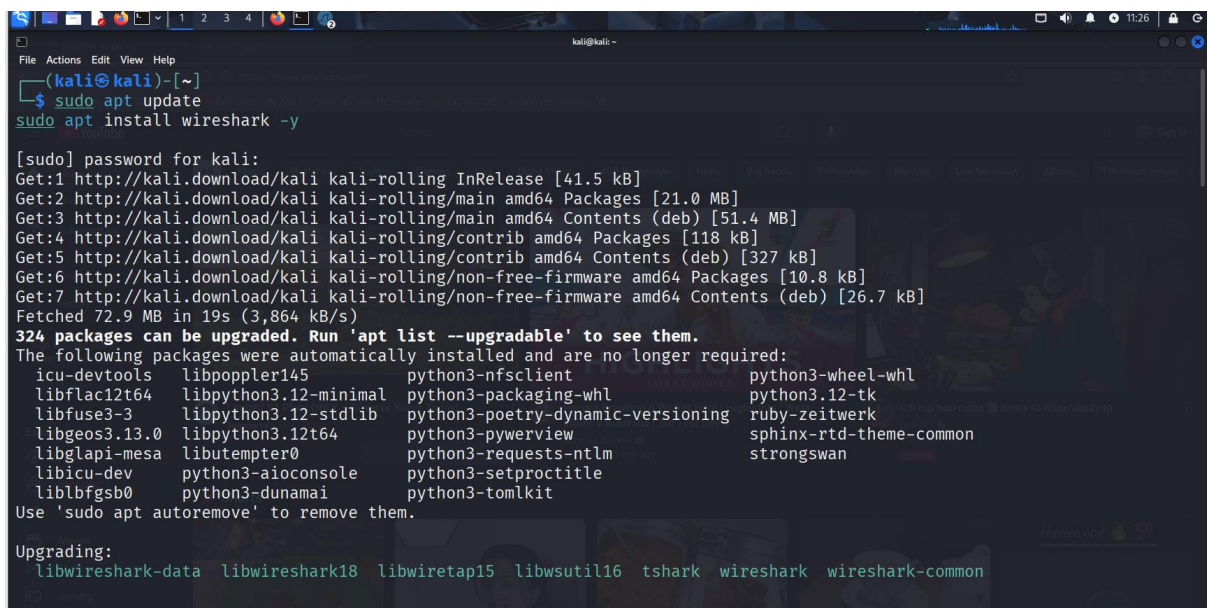
Open a terminal and run the following commands:

```
sudo apt update
```

```
sudo apt install wireshark -y
```

During installation, select **Yes** if prompted to allow non-superusers to capture packets. Then, add your user to the wireshark group

```
sudo usermod -aG wireshark $USER
```



```
kali@kali ~  
File Actions Edit View Help  
(kali@kali)~  
$ sudo apt update  
sudo apt install wireshark -y  
[sudo] password for kali:  
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]  
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]  
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.4 MB]  
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [118 kB]  
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [327 kB]  
Get:6 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.8 kB]  
Get:7 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [26.7 kB]  
Fetched 72.9 MB in 19s (3,864 kB/s)  
324 packages can be upgraded. Run 'apt list --upgradable' to see them.  
The following packages were automatically installed and are no longer required:  
icu-devtools libpoppler145 python3-nfsclient python3-wheel-whl  
libflac12t64 libpython3.12-minimal python3-packaging-whl python3.12-tk  
libfuse3-3 libpython3.12-stdlib python3-poetry-dynamic-versioning ruby-zeitwerk  
libgeos3.13.0 libpython3.12t64 python3-pywerview sphinx-rtd-theme-common  
libglapi-mesa libutempter0 python3-requests-ntlm strongswan  
libicu-dev python3-aioconsole python3-setproctitle  
liblbfgsb0 python3-dunamai python3-tomlkit  
Use 'sudo apt autoremove' to remove them.  
Upgrading:  
libwireshark-data libwireshark18 libwiretap15 libwsutil16 tshark wireshark wireshark-common
```

3. Launching Wireshark

Start Wireshark via terminal:

```
wireshark
```

```

(kali@kali)-[~]
$ sudo usermod -aG wireshark $USER

(kali@kali)-[~]
$ wireshark

** (wireshark:3136) 11:10:49.593890 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::SystemPalette
** (wireshark:3136) 11:10:49.595256 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::ToolButtonPalette
** (wireshark:3136) 11:10:49.595650 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::ButtonPalette
** (wireshark:3136) 11:10:49.595667 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::CheckBoxPalette
** (wireshark:3136) 11:10:49.595679 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::RadioButtonPalette
** (wireshark:3136) 11:10:49.595691 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::HeaderPalette
** (wireshark:3136) 11:10:49.595704 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::ItemViewPalette
** (wireshark:3136) 11:10:49.595717 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::MessageBoxLabelPelette
** (wireshark:3136) 11:10:49.595729 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P
alette) const QPlatformTheme::TabBarPalette
** (wireshark:3136) 11:10:49.595741 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::palette(QPlatformTheme::P

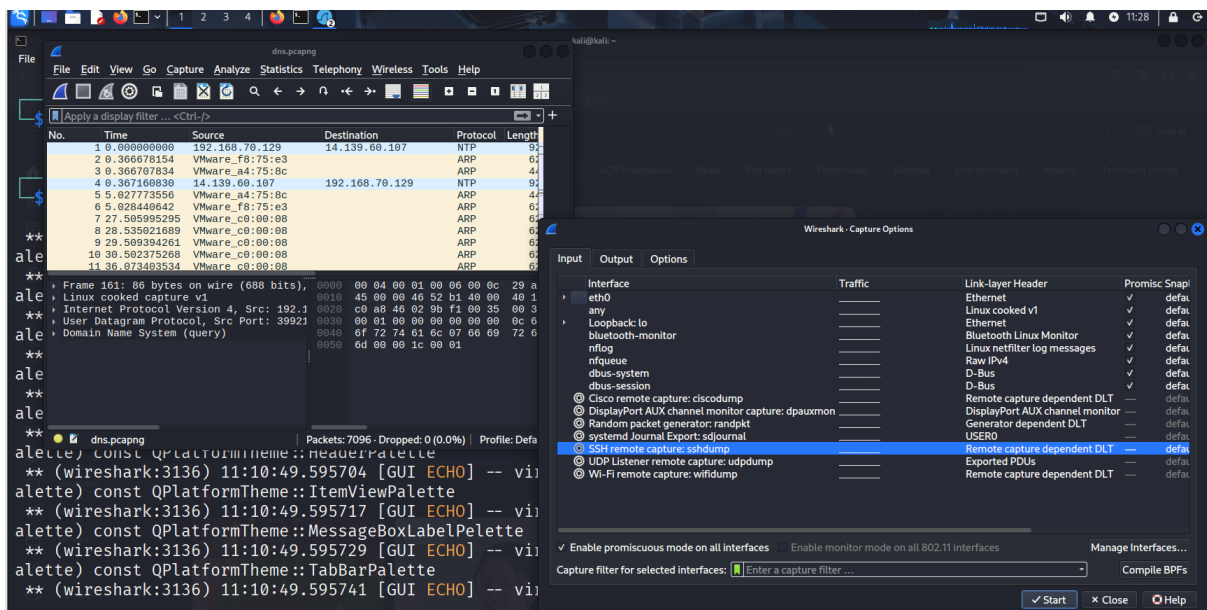
```

4. Capturing Network Traffic

- Choose the appropriate interface (e.g., eth0 for Ethernet, wlan0 for Wi-Fi).
- Click the interface to start capturing packets.

Optional Capture Filters:

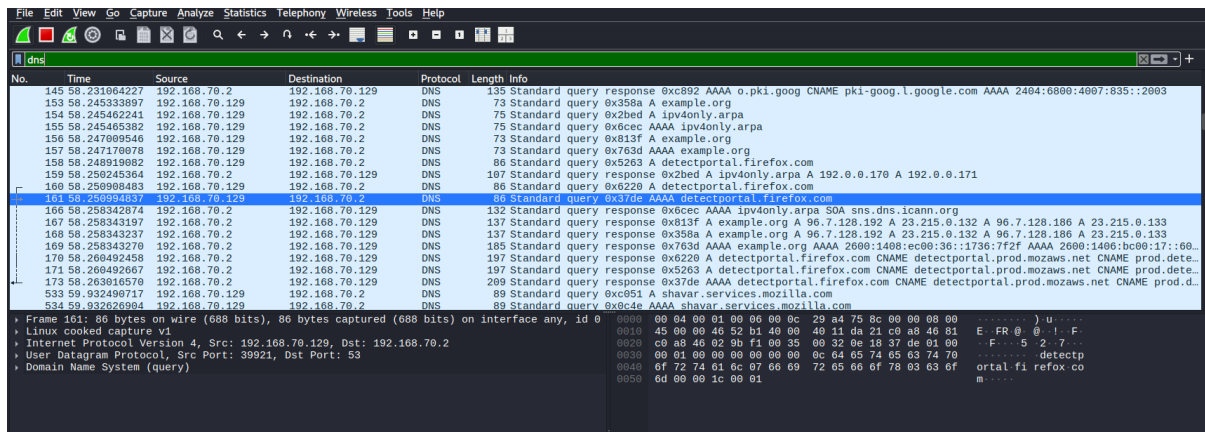
- Capture HTTP traffic: port 80
- Capture traffic from a specific IP: host 192.168.1.10



Analyzing Captured Packets

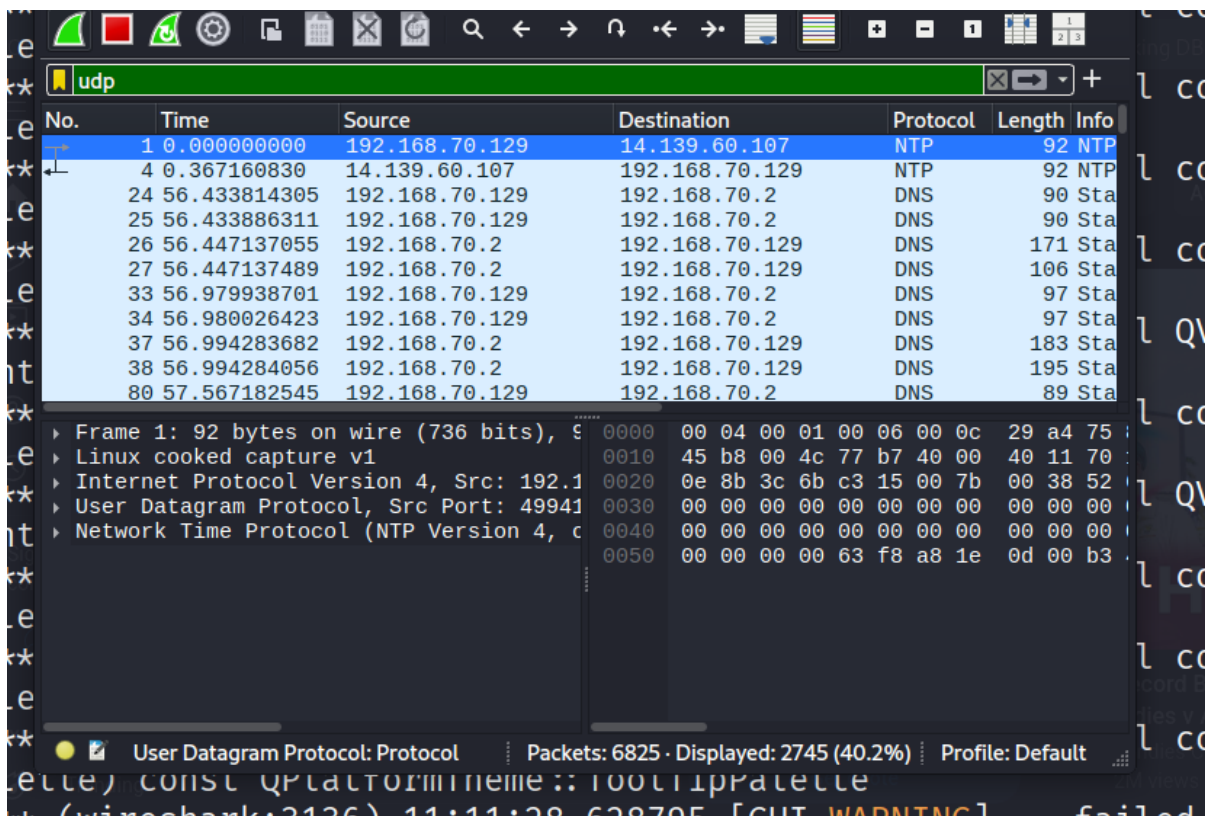
Common protocols to analyze:

- **HTTP** – Inspect requests and responses
- **DNS** – View name resolution
- **TCP/UDP** – Observe ports and data flow
- **ICMP** – Analyze ping packets



The image shows a Wireshark packet capture of DNS traffic. The packet list on the left shows several DNS queries and responses. The selected packet (No. 161) is a standard query response from 192.168.70.2 to 192.168.70.129. The packet details pane on the right shows the structure of the DNS response, including the question section, answer section, and authority section. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
145	58.231064227	192.168.70.2	192.168.70.129	DNS	135	Standard query response 0xc892 AAAA o.pki.goog CNAME pki-goog.l.google.com AAAA 2404:6800:4007:835::2003
153	58.245333897	192.168.70.129	192.168.70.2	DNS	73	Standard query 0x358a A example.org
154	58.245462241	192.168.70.129	192.168.70.2	DNS	75	Standard query 0x2bed A ipv4only.arpa
155	58.245465382	192.168.70.129	192.168.70.2	DNS	75	Standard query 0x6cec AAAA ipv4only.arpa
156	58.247099546	192.168.70.129	192.168.70.2	DNS	73	Standard query 0x813f A example.org
157	58.247179078	192.168.70.129	192.168.70.2	DNS	73	Standard query 0x763d AAAA example.org
158	58.248919882	192.168.70.129	192.168.70.2	DNS	86	Standard query 0x5263 A detectportal.firefox.com
159	58.250245364	192.168.70.2	192.168.70.129	DNS	187	Standard query response 0x2bed A ipv4only.arpa A 192.0.0.170 A 192.0.0.171
160	58.250900483	192.168.70.129	192.168.70.2	DNS	86	Standard query 0x6229 A detectportal.firefox.com
161	58.250934037	192.168.70.129	192.168.70.2	DNS	86	Standard query 0x37de AAAA detectportal.firefox.com
166	58.258342874	192.168.70.2	192.168.70.129	DNS	132	Standard query response 0x6cec AAAA ipv4only.arpa SOA sns.dns.icann.org
167	58.258343197	192.168.70.2	192.168.70.129	DNS	137	Standard query response 0x813f A example.org A 96.7.128.192 A 23.215.0.132 A 96.7.128.186 A 23.215.0.133
168	58.258343237	192.168.70.2	192.168.70.129	DNS	137	Standard query response 0x358a A example.org A 96.7.128.192 A 23.215.0.132 A 96.7.128.186 A 23.215.0.133
169	58.258343270	192.168.70.2	192.168.70.129	DNS	185	Standard query response 0x763d AAAA example.org AAAA 2600:1408:ec00:36::1736:7f2f AAAA 2600:1408:bc00:17::160
170	58.260492458	192.168.70.2	192.168.70.129	DNS	197	Standard query response 0x6229 A detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.dete
171	58.260492667	192.168.70.2	192.168.70.129	DNS	197	Standard query response 0x5263 A detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.dete
173	58.263010570	192.168.70.2	192.168.70.129	DNS	209	Standard query response 0x37de AAAA detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.dete
533	59.932490717	192.168.70.129	192.168.70.2	DNS	89	Standard query 0xc851 A shavar.services.mozilla.com
534	59.932626904	192.168.70.129	192.168.70.2	DNS	89	Standard query 0xc6de AAAA shavar.services.mozilla.com



The image shows a Wireshark packet capture of UDP traffic. The packet list on the left shows several UDP packets. The selected packet (No. 1) is an NTP packet from 192.168.70.129 to 14.139.60.107. The packet details pane on the right shows the structure of the NTP packet, including the NTP header and the NTP body. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.70.129	14.139.60.107	NTP	92	NTP
4	0.367160830	14.139.60.107	192.168.70.129	NTP	92	NTP
24	56.433814305	192.168.70.129	192.168.70.2	DNS	90	Sta
25	56.433886311	192.168.70.129	192.168.70.2	DNS	90	Sta
26	56.447137055	192.168.70.2	192.168.70.129	DNS	171	Sta
27	56.447137489	192.168.70.2	192.168.70.129	DNS	106	Sta
33	56.7979938701	192.168.70.129	192.168.70.2	DNS	97	Sta
34	56.980026423	192.168.70.129	192.168.70.2	DNS	97	Sta
37	56.994283682	192.168.70.2	192.168.70.129	DNS	183	Sta
38	56.994284056	192.168.70.2	192.168.70.129	DNS	195	Sta
80	57.567182545	192.168.70.129	192.168.70.2	DNS	89	Sta

tcp

No.	Time	Source	Destination	Protocol	Length	Info
28	56.449458975	192.168.70.129	34.36.137.203	TCP	76	583
29	56.695736547	34.36.137.203	192.168.70.129	TCP	62	443
30	56.695794676	192.168.70.129	34.36.137.203	TCP	56	583
31	56.698918772	192.168.70.129	34.36.137.203	TLSv1.3	732	Client Hello
32	56.699356309	34.36.137.203	192.168.70.129	TCP	62	443
35	56.982596681	34.36.137.203	192.168.70.129	TLSv1.3	3143	Server Hello
36	56.982648385	192.168.70.129	34.36.137.203	TCP	56	583
39	56.997620519	192.168.70.129	34.160.144.191	TCP	76	369
40	57.030899979	34.160.144.191	192.168.70.129	TCP	62	443
41	57.030932818	192.168.70.129	34.160.144.191	TCP	56	369
42	57.031917964	192.168.70.129	34.160.144.191	TLSv1.2	272	Client Hello

▶ Frame 28: 76 bytes on wire (608 bits),
 ▶ Linux cooked capture v1
 ▶ Internet Protocol Version 4, Src: 192.168.70.129, Dst: 34.36.137.203
 ▶ Transmission Control Protocol, Src Port: 583, Dst Port: 443

Transmission Cont...rotocol: Protocol ... Packets: 6685 · Displayed: 3905 (58.4%) ... Profile: Default

The image displays two screenshots of the Wireshark network traffic analysis tool. The top screenshot shows a list of captured packets, with packet 164 selected. The bottom screenshot shows the details pane for the selected packet, displaying the TCP stream data.

Top Screenshot: Packet List

No.	Time	Source	Destination	Protocol	Length	Info
164	58.258693469	192.168.70.129	142.251.222.195	HTTP	483	Request
176	58.515278398	142.251.222.195	192.168.70.129	OCSP	966	Response
179	58.515854611	192.168.70.129	34.107.221.82	HTTP	366	GET /success.txt?ip=v4 HTTP/1.1
192	58.530638193	34.107.221.82	192.168.70.129	HTTP	272	HTTP/1.1 200 OK (text/plain)
627	64.562919701	192.168.70.129	142.251.222.195	OCSP	483	Request
628	64.563545166	192.168.70.129	142.251.222.195	OCSP	483	Request
629	64.563695324	192.168.70.129	142.251.222.195	OCSP	483	Request
630	64.563842142	192.168.70.129	142.251.222.195	OCSP	483	Request
635	64.652484929	142.251.222.195	192.168.70.129	OCSP	965	Response
636	64.652485495	142.251.222.195	192.168.70.129	OCSP	965	Response
638	64.652485698	142.251.222.195	192.168.70.129	OCSP	965	Response
639	64.652485799	142.251.222.195	192.168.70.129	OCSP	965	Response
752	65.121222269	192.168.70.129	142.251.222.195	OCSP	483	Request
889	65.175674923	142.251.222.195	192.168.70.129	OCSP	965	Response
1058	65.217742929	192.168.70.129	142.251.222.195	OCSP	483	Request
1317	65.272347689	192.168.70.129	142.251.222.195	OCSP	483	Request
1353	65.279201485	142.251.222.195	192.168.70.129	OCSP	966	Response
1517	65.326173397	142.251.222.195	192.168.70.129	OCSP	1159	Response
2083	65.719759376	192.168.70.129	142.251.222.195	OCSP	483	Request

Bottom Screenshot: Packet Details

Frame 164: 483 bytes on wire (3864 bits), 483 bytes captured (3864 bits) on interface any, id 0

Ethernet II, Src: Linux cooked capture v1, Dst: 142.251.222.195

Internet Protocol Version 4, Src: 192.168.70.129, Dst: 142.251.222.195

Transmission Control Protocol, Src Port: 60876, Dst Port: 80, Seq: 1, Ack: 1, Len: 427

Hypertext Transfer Protocol

Online Certificate Status Protocol

Packet Bytes: 0000 00 04 00 01 00 06 00 0c 29 a4 75 8c 00 00 00 00 ... u ...

Top Screenshot: Filter

tcp.port == 80 || udp.port == 80

Bottom Screenshot: Filter

tcp.stream

Packet List (Bottom Screenshot)

No.	Time	Source	Destination	Protocol	Length	Info
28	56.449458975	192.168.70.129	34.36.137.203	TCP	76	583
29	56.695736547	34.36.137.203	192.168.70.129	TCP	62	443
30	56.695794676	192.168.70.129	34.36.137.203	TCP	56	583
31	56.698918772	192.168.70.129	34.36.137.203	TLSv1.3	732	Client Hello
32	56.699356309	34.36.137.203	192.168.70.129	TCP	62	443
35	56.982596681	34.36.137.203	192.168.70.129	TLSv1.3	3143	Server Hello
36	56.982648385	192.168.70.129	34.36.137.203	TCP	56	583
39	56.997620519	192.168.70.129	34.160.144.191	TCP	76	369
40	57.030899979	34.160.144.191	192.168.70.129	TCP	62	443
41	57.030932818	192.168.70.129	34.160.144.191	TCP	56	369
42	57.031917964	192.168.70.129	34.160.144.191	TLSv1.2	272	Client Hello

Packet Details (Bottom Screenshot)

Frame 28: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0

Ethernet II, Src: Linux cooked capture v1, Dst: 34.36.137.203

Internet Protocol Version 4, Src: 192.168.70.129, Dst: 34.36.137.203

Transmission Control Protocol, Src Port: 60876, Dst Port: 80, Seq: 1, Len: 427

Hypertext Transfer Protocol

Online Certificate Status Protocol

Packet Bytes (Bottom Screenshot): 0000 00 04 00 01 00 06 00 0c 29 a4 75 8c 00 00 00 00 ... u ...

5. Stopping the Capture

Click the red square **Stop** button in the toolbar to end the capture.

Conclusion

Wireshark is a powerful tool for network traffic capture and analysis. Mastering its filters, stream tracking, and protocol analysis enables in-depth understanding of network behavior and threat detection.