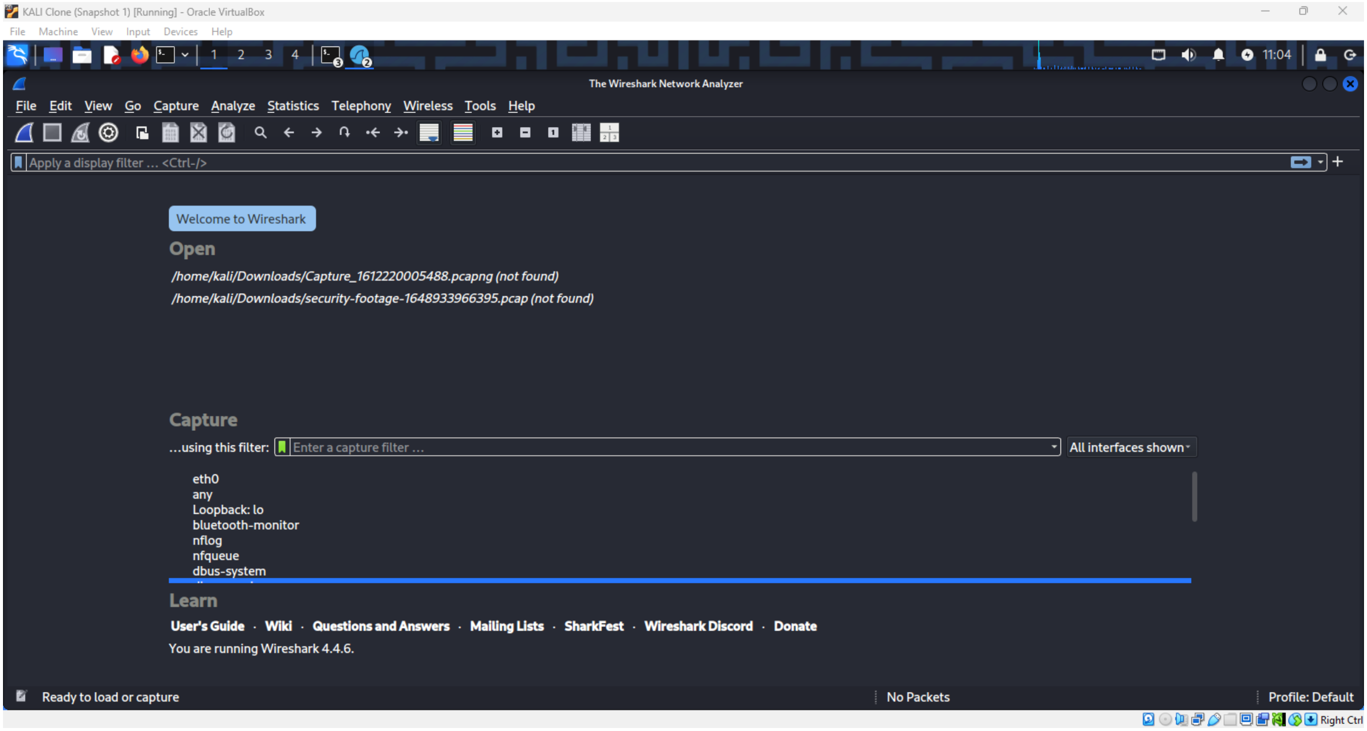
Wireshark Traffic Capture and Analysis

Cybersecurity Internship

Reported by Thumma Kiranmai

Date: June 02, 2025

Available Network Interfaces in Wireshark



KALI Clone (Snapshot 1) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
11:04
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Right Ctrl

Welcome to Wireshark

Open

/home/kali/Downloads/Capture_1612220005488.pcapng (not found)
/home/kali/Downloads/security-footage-1648933966395.pcap (not found)

Capture

…using this filter: Enter a capture filter … All interfaces shown

eth0
any
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system

Learn

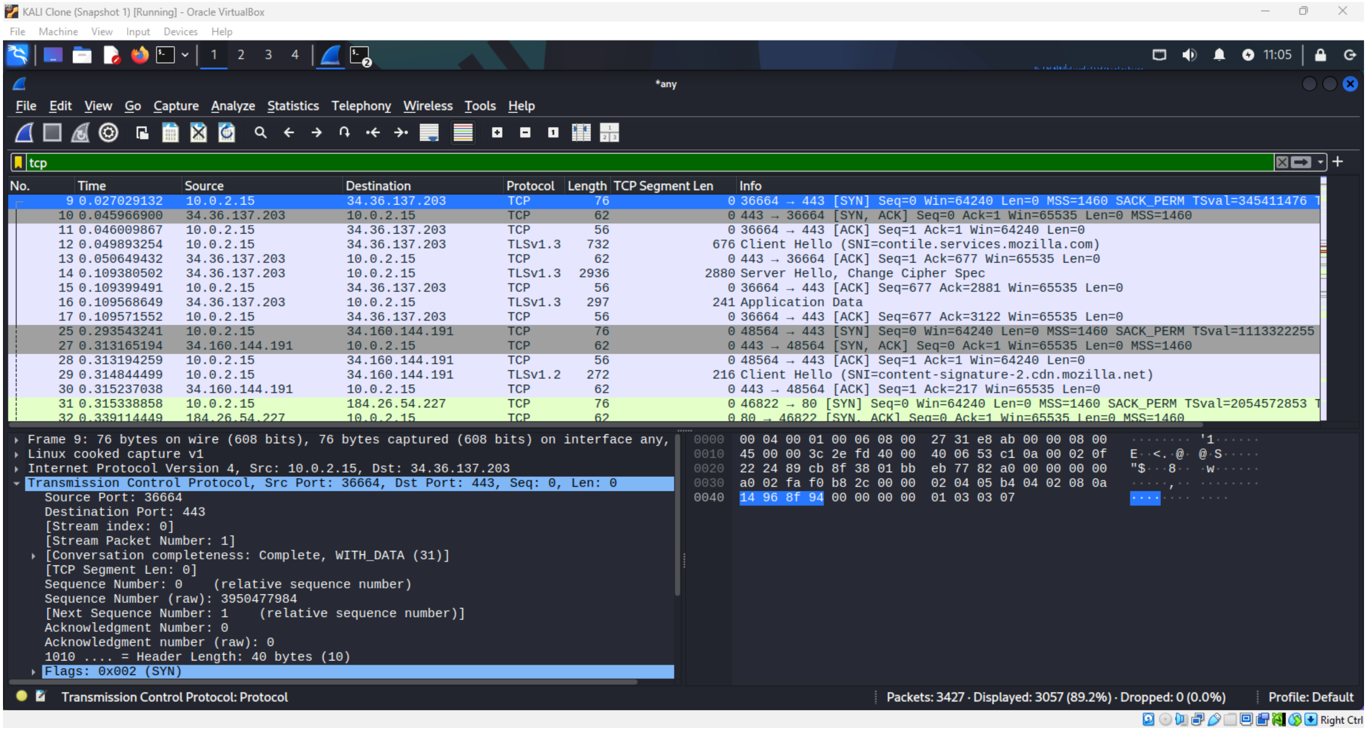
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

You are running Wireshark 4.4.6.

Ready to load or capture | No Packets | Profile: Default

This view shows the initial interface of Wireshark on a Kali Linux virtual machine. It lists all available network interfaces, such as eth0, loopback, and others. Choosing the right interface is crucial for effective packet capture, as different interfaces monitor different types of traffic. For instance, eth0 is typically used for Ethernet LAN connections. The 'any' option allows capturing from all interfaces simultaneously. The window also includes previously opened capture files, but these files were not found, indicating either deletion or relocation. The capture filter option is also available here, allowing the user to filter packets during capture. This interface is user-friendly and customizable. The user can begin a new session by double-clicking an interface or pressing the start capture button. The interface also provides access to Wireshark's online documentation and help tools. This foundational screen is where all network traffic analysis begins.

TCP Protocol Packet Analysis


KALI Clone (Snapshot 1) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
11:05
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Right Ctrl

tcp

```
No.  Time         Source          Destination     Protocol Length TCP Segment Len Info
   9 0.027029132  10.0.2.15       34.36.137.203   TCP        76        0 36664 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=345411476 T
  10 0.045966900  34.36.137.203   10.0.2.15       TCP        62        0 443 → 36664 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
  11 0.046009867  10.0.2.15       34.36.137.203   TCP        56        0 36664 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
  12 0.049893254  10.0.2.15       34.36.137.203   TLSv1.3   732      676 Client Hello (SNI=contile.services.mozilla.com)
  13 0.050649432  34.36.137.203   10.0.2.15       TCP        62        0 443 → 36664 [ACK] Seq=1 Ack=677 Win=65535 Len=0
  14 0.109380502  34.36.137.203   10.0.2.15       TLSv1.3  2936     2880 Server Hello, Change Cipher Spec
  15 0.109399491  10.0.2.15       34.36.137.203   TCP        56        0 36664 → 443 [ACK] Seq=677 Ack=2881 Win=65535 Len=0
  16 0.109568649  34.36.137.203   10.0.2.15       TLSv1.3   297      241 Application Data
  17 0.109571552  10.0.2.15       34.36.137.203   TCP        56        0 36664 → 443 [ACK] Seq=677 Ack=3122 Win=65535 Len=0
  25 0.293543241  10.0.2.15       34.160.144.191  TCP        76        0 48564 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1113322255
  27 0.313165194  34.160.144.191  10.0.2.15       TCP        62        0 443 → 48564 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
  28 0.313194259  10.0.2.15       34.160.144.191  TCP        56        0 48564 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
  29 0.314844499  10.0.2.15       34.160.144.191  TLSv1.2   272      216 Client Hello (SNI=content-signature-2.cdn.mozilla.net)
  30 0.315237038  34.160.144.191  10.0.2.15       TCP        62        0 443 → 48564 [ACK] Seq=1 Ack=217 Win=65535 Len=0
  31 0.315338858  10.0.2.15       184.26.54.227   TCP        76        0 46822 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2054572853 T
  32 0.339114449  184.26.54.227   10.0.2.15       TCP        62        0 80 → 46822 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
```

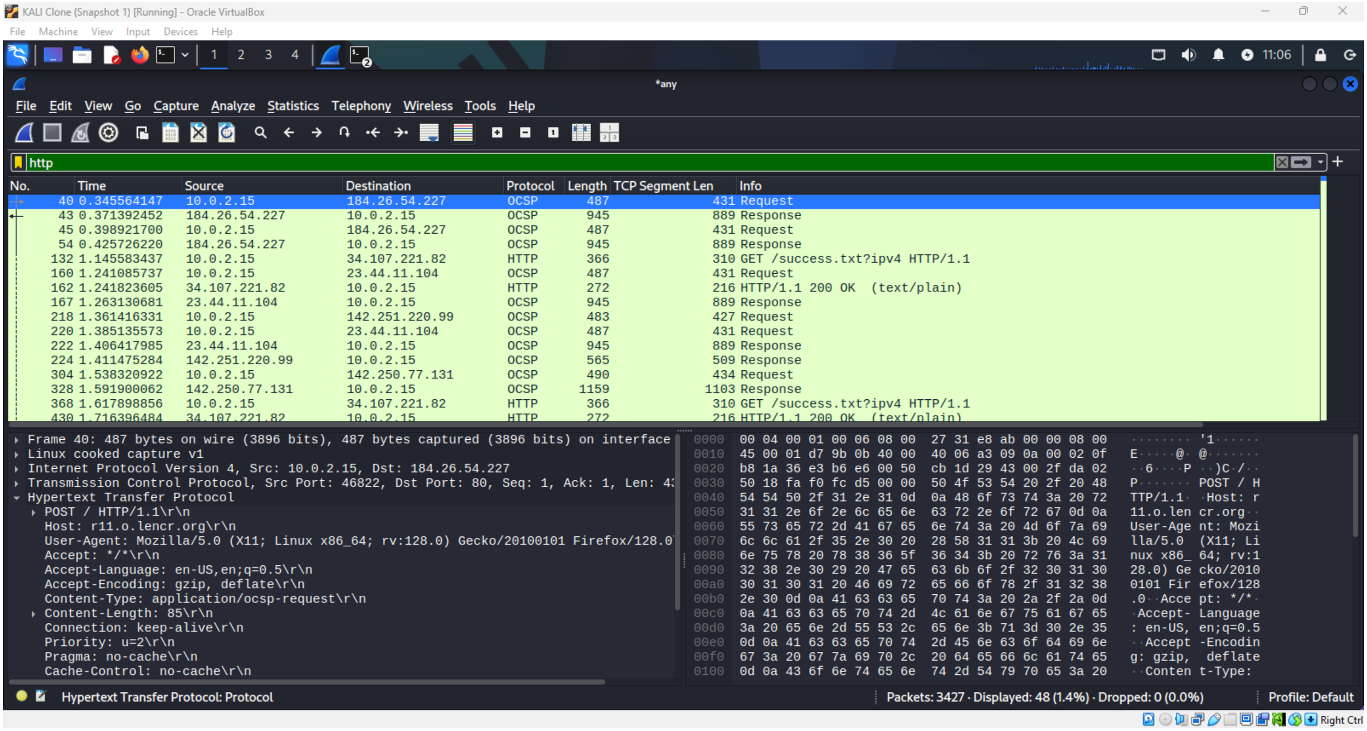
```
› Frame 9: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any,
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 10.0.2.15, Dst: 34.36.137.203
⌄ Transmission Control Protocol, Src Port: 36664, Dst Port: 443, Seq: 0, Len: 0
    Source Port: 36664
    Destination Port: 443
    [Stream index: 0]
    [Stream Packet Number: 1]
  › [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3950477984
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1010 .... = Header Length: 40 bytes (10)
  › Flags: 0x002 (SYN)
```

```
0000  00 04 00 01 00 06 08 00  27 31 e8 ab 00 00 08 00   ········ '1······
0010  45 00 00 3c 2e fd 40 00  40 06 53 c1 0a 00 02 0f   E··<.·@· @·S·····
0020  22 24 89 cb 8f 38 01 bb  eb 77 82 a0 00 00 00 00   "$···8·· ·w······
0030  a0 02 fa f0 b8 2c 00 00  02 04 05 b4 04 02 08 0a   ·····,·· ········
0040  14 96 8f 94 00 00 00 00  01 03 03 07               ········ ····
```

```
Transmission Control Protocol: Protocol    Packets: 3427 · Displayed: 3057 (89.2%) · Dropped: 0 (0.0%)    Profile: Default
```

This filtered capture shows only TCP protocol packets, which are the backbone of most internet communication. The initial SYN, SYN-ACK, and ACK packets visible here represent the three-way handshake used to establish a TCP connection. Wireshark displays detailed information such as sequence numbers, acknowledgment numbers, and TCP flags (SYN, ACK). The hex dump at the bottom shows raw packet content. The packets come from different IP sources, including Mozilla servers, indicating likely web browser traffic. Some packets show TLS negotiation, as TLS rides over TCP. This helps identify encrypted HTTPS connections. TCP analysis is useful for diagnosing issues like retransmissions, out-of-order packets, or latency. TCP packet structure includes headers and options that are vital in deep traffic inspection. Overall, this view is essential for understanding stateful connections and network flow.

HTTP Traffic Analysis


KALI Clone (Snapshot 1) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
11:06
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Right Ctrl

http

No.	Time	Source	Destination	Protocol	Length	TCP Segment Len	Info
40	0.345564147	10.0.2.15	184.26.54.227	OCSP	487	431	Request
43	0.371392452	184.26.54.227	10.0.2.15	OCSP	945	889	Response
45	0.398921700	10.0.2.15	184.26.54.227	OCSP	487	431	Request
54	0.425726220	184.26.54.227	10.0.2.15	OCSP	945	889	Response
132	1.145583437	10.0.2.15	34.107.221.82	HTTP	366	310	GET /success.txt?ipv4 HTTP/1.1
160	1.241085737	10.0.2.15	23.44.11.104	OCSP	487	431	Request
162	1.241823605	34.107.221.82	10.0.2.15	HTTP	272	216	HTTP/1.1 200 OK (text/plain)
167	1.263130681	23.44.11.104	10.0.2.15	OCSP	945	889	Response
218	1.361416331	10.0.2.15	142.251.220.99	OCSP	483	427	Request
220	1.385135573	10.0.2.15	23.44.11.104	OCSP	487	431	Request
222	1.406417985	23.44.11.104	10.0.2.15	OCSP	945	889	Response
224	1.411475284	142.251.220.99	10.0.2.15	OCSP	565	509	Response
304	1.538320922	10.0.2.15	142.250.77.131	OCSP	490	434	Request
328	1.591900062	142.250.77.131	10.0.2.15	OCSP	1159	1103	Response
368	1.617898856	10.0.2.15	34.107.221.82	HTTP	366	310	GET /success.txt?ipv4 HTTP/1.1
430	1.716396484	34.107.221.82	10.0.2.15	HTTP	272	216	HTTP/1.1 200 OK (text/plain)

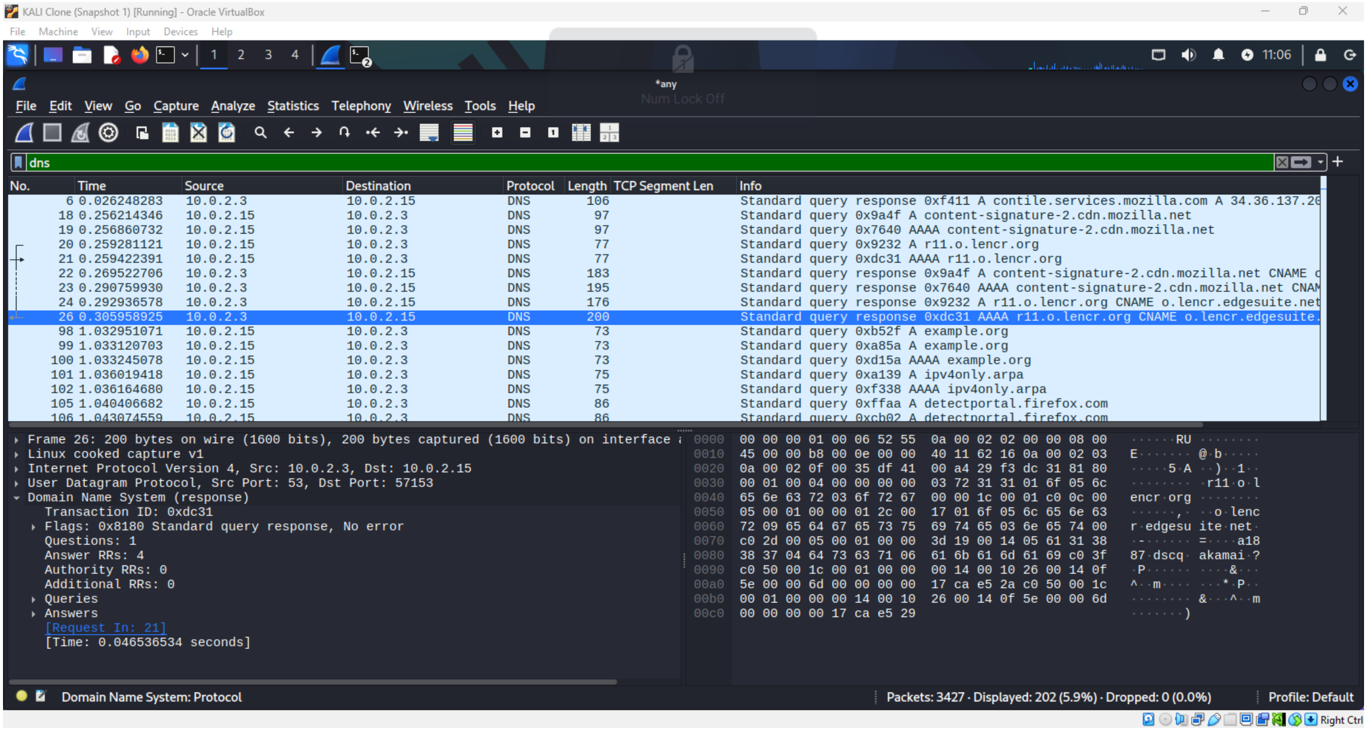
```
› Frame 40: 487 bytes on wire (3896 bits), 487 bytes captured (3896 bits) on interface
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 10.0.2.15, Dst: 184.26.54.227
› Transmission Control Protocol, Src Port: 46822, Dst Port: 80, Seq: 1, Ack: 1, Len: 43
⌄ Hypertext Transfer Protocol
  › POST / HTTP/1.1\r\n
    Host: r11.o.lencr.org\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
    Accept: */*\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: application/ocsp-request\r\n
  › Content-Length: 85\r\n
    Connection: keep-alive\r\n
    Priority: u=2\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
```

```
0000  00 04 00 01 00 06 08 00  27 31 e8 ab 00 00 08 00   ········ '1······
0010  45 00 01 d7 9b 0b 40 00  40 06 a3 09 0a 00 02 0f   E·····@· @·······
0020  b8 1a 36 e3 b6 e6 00 50  cb 1d 29 43 00 2f da 02   ··6····P ··)C·/··
0030  50 18 fa f0 fc d5 00 00  50 4f 53 54 20 2f 20 48   P······· POST / H
0040  54 54 50 2f 31 2e 31 0d  0a 48 6f 73 74 3a 20 72   TTP/1.1· ·Host: r
0050  31 31 2e 6f 2e 6c 65 6e  63 72 2e 6f 72 67 0d 0a   11.o.len cr.org··
0060  55 73 65 72 2d 41 67 65  6e 74 3a 20 4d 6f 7a 69   User-Age nt: Mozi
0070  6c 6c 61 2f 35 2e 30 20  28 58 31 31 3b 20 4c 69   lla/5.0  (X11; Li
0080  6e 75 78 20 78 38 36 5f  36 34 3b 20 72 76 3a 31   nux x86_ 64; rv:1
0090  32 38 2e 30 29 20 47 65  63 6b 6f 2f 32 30 31 30   28.0) Ge cko/2010
00a0  30 31 30 31 20 46 69 72  65 66 6f 78 2f 31 32 38   0101 Fir efox/128
00b0  2e 30 0d 0a 41 63 63 65  70 74 3a 20 2a 2f 2a 0d   .0··Acce pt: */*·
00c0  0a 41 63 63 65 70 74 2d  4c 61 6e 67 75 61 67 65   ·Accept- Language
00d0  3a 20 65 6e 2d 55 53 2c  65 6e 3b 71 3d 30 2e 35   : en-US, en;q=0.5
00e0  0d 0a 41 63 63 65 70 74  2d 45 6e 63 6f 64 69 6e   ··Accept -Encodin
00f0  67 3a 20 67 7a 69 70 2c  20 64 65 66 6c 61 74 65   g: gzip,  deflate
0100  0d 0a 43 6f 6e 74 65 6e  74 2d 54 79 70 65 3a 20   ··Conten t-Type: 
```

Hypertext Transfer Protocol: Protocol | Packets: 3427 · Displayed: 48 (1.4%) · Dropped: 0 (0.0%) | Profile: Default

This filtered view shows HTTP traffic, one of the most common and readable protocols captured by Wireshark. HTTP GET and POST methods are visible, indicating standard web browsing activity. The details pane displays headers like User-Agent, Host, Accept-Language, and Content-Length. This level of detail allows analysts to inspect what information a browser sends to a server. The source and destination IPs and ports are shown, allowing traceability. HTTP is a stateless protocol, so every request is independent. This helps in recognizing site visits, file downloads, or API communications. Here, the user is accessing URLs such as /success.txt and contacting Mozilla-related endpoints. HTTP captures can be used to extract requested URLs, detect potential attacks like SQL injection, or debug API issues. In real-world applications, security teams monitor HTTP traffic for anomalies and potential data leaks.

DNS Query and Response Analysis


KALI Clone (Snapshot 1) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
11:06
*any
Num Lock Off
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
Right Ctrl

```
No.  Time         Source      Destination  Protocol Length TCP Segment Len Info
   6 0.026248283  10.0.2.3    10.0.2.15    DNS      106                    Standard query response 0xf411 A contile.services.mozilla.com A 34.36.137.20
  18 0.256214346  10.0.2.15   10.0.2.3     DNS       97                    Standard query 0x9a4f A content-signature-2.cdn.mozilla.net
  19 0.256860732  10.0.2.15   10.0.2.3     DNS       97                    Standard query 0x7640 AAAA content-signature-2.cdn.mozilla.net
  20 0.259281121  10.0.2.15   10.0.2.3     DNS       77                    Standard query 0x9232 A r11.o.lencr.org
  21 0.259422391  10.0.2.15   10.0.2.3     DNS       77                    Standard query 0xdc31 AAAA r11.o.lencr.org
  22 0.269522706  10.0.2.3    10.0.2.15    DNS      183                    Standard query response 0x9a4f A content-signature-2.cdn.mozilla.net CNAME c
  23 0.290759930  10.0.2.3    10.0.2.15    DNS      195                    Standard query response 0x7640 AAAA content-signature-2.cdn.mozilla.net CNAM
  24 0.292936578  10.0.2.3    10.0.2.15    DNS      176                    Standard query response 0x9232 A r11.o.lencr.org CNAME o.lencr.edgesuite.net
  26 0.305958925  10.0.2.3    10.0.2.15    DNS      200                    Standard query response 0xdc31 AAAA r11.o.lencr.org CNAME o.lencr.edgesuite.
  98 1.032951071  10.0.2.15   10.0.2.3     DNS       73                    Standard query 0xb52f A example.org
  99 1.033120703  10.0.2.15   10.0.2.3     DNS       73                    Standard query 0xa85a A example.org
 100 1.033245078  10.0.2.15   10.0.2.3     DNS       73                    Standard query 0xd15a AAAA example.org
 101 1.036019418  10.0.2.15   10.0.2.3     DNS       75                    Standard query 0xa139 A ipv4only.arpa
 102 1.036164680  10.0.2.15   10.0.2.3     DNS       75                    Standard query 0xf338 AAAA ipv4only.arpa
 105 1.040406682  10.0.2.15   10.0.2.3     DNS       86                    Standard query 0xffaa A detectportal.firefox.com
 106 1.043074559  10.0.2.15   10.0.2.3     DNS       86                    Standard query 0xcb02 A detectportal.firefox.com
```

```
› Frame 26: 200 bytes on wire (1600 bits), 200 bytes captured (1600 bits) on interface a
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 10.0.2.3, Dst: 10.0.2.15
› User Datagram Protocol, Src Port: 53, Dst Port: 57153
⌄ Domain Name System (response)
    Transaction ID: 0xdc31
  › Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 4
    Authority RRs: 0
    Additional RRs: 0
  › Queries
  › Answers
    [Request In: 21]
    [Time: 0.046536534 seconds]
```

```
0000  00 00 00 01 00 06 52 55  0a 00 02 02 00 00 08 00   ······RU ········
0010  45 00 00 b8 00 0e 00 00  40 11 62 16 0a 00 02 03   E······· @·b····
0020  0a 00 02 0f 00 35 df 41  00 a4 29 f3 dc 31 81 80   ·····5·A ··)··1··
0030  00 01 00 04 00 00 00 00  03 72 31 31 01 6f 05 6c   ········ ·r11·o·l
0040  65 6e 63 72 03 6f 72 67  00 00 1c 00 01 c0 0c 00   encr·org ········
0050  05 00 01 00 00 01 2c 00  17 01 6f 05 6c 65 6e 63   ······,· ··o·lenc
0060  72 09 65 64 67 65 73 75  69 74 65 03 6e 65 74 00   r·edgesu ite·net·
0070  c0 2d 00 05 00 01 00 00  3d 19 00 14 05 61 31 38   ·-······ =····a18
0080  38 37 04 64 73 63 71 06  61 6b 61 6d 61 69 c0 3f   87·dscq· akamai·?
0090  c0 50 00 1c 00 01 00 00  00 14 00 10 26 00 14 0f   ·P······ ····&···
00a0  5e 00 00 6d 00 00 00 00  17 ca e5 2a c0 50 00 1c   ^··m···· ···*·P··
00b0  00 01 00 00 00 14 00 10  26 00 14 0f 5e 00 00 6d   ········ &···^··m
00c0  00 00 00 00 17 ca e5 29                            ·······)
```

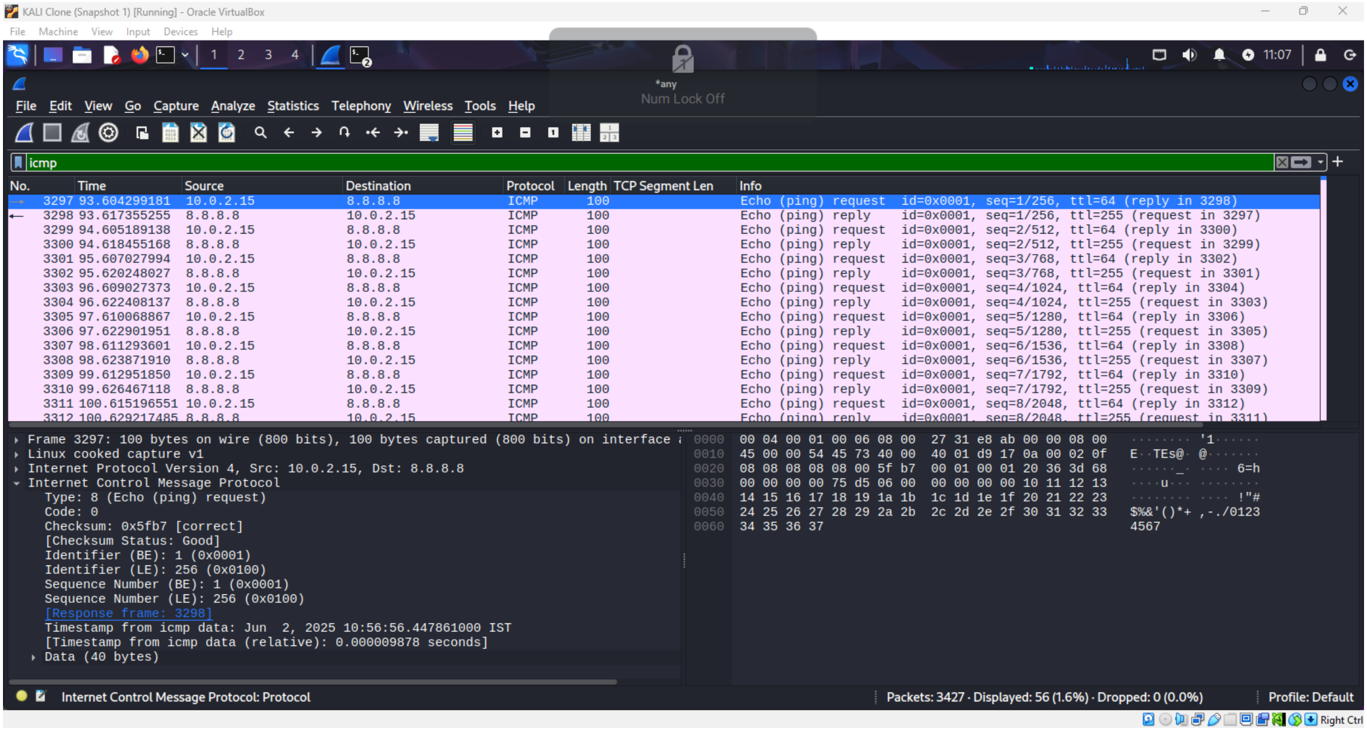
Domain Name System: Protocol

Packets: 3427 · Displayed: 202 (5.9%) · Dropped: 0 (0.0%)

Profile: Default

This capture highlights DNS traffic, a protocol used to resolve domain names into IP addresses. The filter displays standard DNS queries and responses, including AAAA and A records. The source IP (10.0.2.3) queries a DNS server at 10.0.2.15. The domain names resolved include mozilla.net and lencr.org. DNS is crucial for initiating any internet connection, as it is usually the first step a device takes before connecting to a website. In the details pane, transaction IDs, query types, and answer sections are visible. This lets analysts detect DNS poisoning, identify malware callbacks, or confirm if a domain is resolving correctly. Standard query responses indicate successful name resolution. DNS can use UDP or TCP, and in this case, it's shown over UDP. Monitoring DNS is essential for both security and troubleshooting internet access issues.

ICMP (Ping) Packet Analysis


KALI Clone (Snapshot 1) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
11:07
*any
Num Lock Off
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Right Ctrl

icmp

```
No.     Time          Source      Destination  Protocol Length TCP Segment Len Info
  3297 93.604299181  10.0.2.15   8.8.8.8      ICMP     100                    Echo (ping) request  id=0x0001, seq=1/256, ttl=64 (reply in 3298)
  3298 93.617355255  8.8.8.8     10.0.2.15    ICMP     100                    Echo (ping) reply    id=0x0001, seq=1/256, ttl=255 (request in 3297)
  3299 94.605189138  10.0.2.15   8.8.8.8      ICMP     100                    Echo (ping) request  id=0x0001, seq=2/512, ttl=64 (reply in 3300)
  3300 94.618455168  8.8.8.8     10.0.2.15    ICMP     100                    Echo (ping) reply    id=0x0001, seq=2/512, ttl=255 (request in 3299)
  3301 95.607027994  10.0.2.15   8.8.8.8      ICMP     100                    Echo (ping) request  id=0x0001, seq=3/768, ttl=64 (reply in 3302)
  3302 95.620248027  8.8.8.8     10.0.2.15    ICMP     100                    Echo (ping) reply    id=0x0001, seq=3/768, ttl=255 (request in 3301)
  3303 96.609027373  10.0.2.15   8.8.8.8      ICMP     100                    Echo (ping) request  id=0x0001, seq=4/1024, ttl=64 (reply in 3304)
  3304 96.622408137  8.8.8.8     10.0.2.15    ICMP     100                    Echo (ping) reply    id=0x0001, seq=4/1024, ttl=255 (request in 3303)
  3305 97.610068867  10.0.2.15   8.8.8.8      ICMP     100                    Echo (ping) request  id=0x0001, seq=5/1280, ttl=64 (reply in 3306)
  3306 97.622901951  8.8.8.8     10.0.2.15    ICMP     100                    Echo (ping) reply    id=0x0001, seq=5/1280, ttl=255 (request in 3305)
  3307 98.611293601  10.0.2.15   8.8.8.8      ICMP     100                    Echo (ping) request  id=0x0001, seq=6/1536, ttl=64 (reply in 3308)
  3308 98.623871910  8.8.8.8     10.0.2.15    ICMP     100                    Echo (ping) reply    id=0x0001, seq=6/1536, ttl=255 (request in 3307)
  3309 99.612951850  10.0.2.15   8.8.8.8      ICMP     100                    Echo (ping) request  id=0x0001, seq=7/1792, ttl=64 (reply in 3310)
  3310 99.626467118  8.8.8.8     10.0.2.15    ICMP     100                    Echo (ping) reply    id=0x0001, seq=7/1792, ttl=255 (request in 3309)
  3311 100.615196551 10.0.2.15   8.8.8.8      ICMP     100                    Echo (ping) request  id=0x0001, seq=8/2048, ttl=64 (reply in 3312)
  3312 100.629217485 8.8.8.8     10.0.2.15    ICMP     100                    Echo (ping) reply    id=0x0001, seq=8/2048, ttl=255 (request in 3311)
```

```
Frame 3297: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface a
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.8.8.8
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x5fb7 [correct]
  [Checksum Status: Good]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence Number (BE): 1 (0x0001)
  Sequence Number (LE): 256 (0x0100)
  [Response frame: 3298]
  Timestamp from icmp data: Jun  2, 2025 10:56:56.447861000 IST
  [Timestamp from icmp data (relative): 0.000009878 seconds]
  Data (40 bytes)
```

```
0000  00 04 00 01 00 06 08 00  27 31 e8 ab 00 00 08 00   ········ '1······
0010  45 00 00 54 45 73 40 00  40 01 d9 17 0a 00 02 0f   E··TEs@· @·······
0020  08 08 08 08 08 00 5f b7  00 01 00 01 20 36 3d 68   ······_· ···· 6=h
0030  00 00 00 00 75 d5 06 00  00 00 00 00 10 11 12 13   ····u··· ········
0040  14 15 16 17 18 19 1a 1b  1c 1d 1e 1f 20 21 22 23   ········ ···· !"#
0050  24 25 26 27 28 29 2a 2b  2c 2d 2e 2f 30 31 32 33   $%&'()*+ ,-./0123
0060  34 35 36 37                                        4567
```

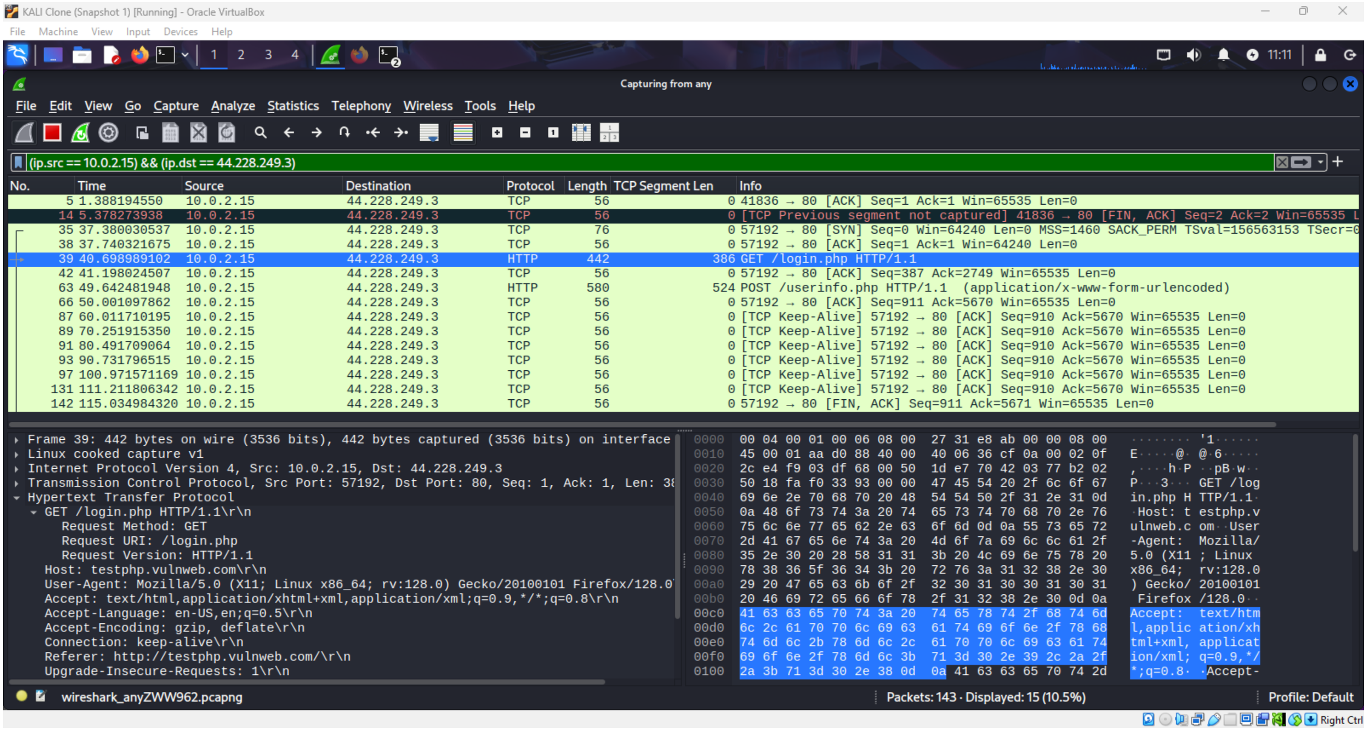
Internet Control Message Protocol: Protocol

Packets: 3427 · Displayed: 56 (1.6%) · Dropped: 0 (0.0%)

Profile: Default

This section captures ICMP packets, which are used primarily for diagnostic purposes like 'ping'. The packet information shows Echo (ping) requests sent to 8.8.8.8, Google's public DNS, and replies received. Each entry includes identifiers, sequence numbers, and TTL values. This type of traffic helps verify network connectivity and latency. If a host is unreachable, ICMP replies can provide clues about the cause. The frame and IP layer details show the type and code of the ICMP message, checksum values, and other metadata. Analyzing ICMP packets can help determine if packets are being dropped or delayed along the route. In cybersecurity, monitoring unexpected ICMP traffic is important, as it could indicate reconnaissance activity or covert data exfiltration. The consistent pattern shown here implies a healthy ping operation between the host and target.

Traffic Between Two Specific IPs


KALI Clone (Snapshot 1) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
11:11
Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Right Ctrl

(ip.src == 10.0.2.15) && (ip.dst == 44.228.249.3)

```
No.   Time          Source      Destination    Protocol Length TCP Segment Len Info
    5 1.388194550   10.0.2.15   44.228.249.3   TCP      56       0 41836 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
   14 5.378273938   10.0.2.15   44.228.249.3   TCP      56       0 [TCP Previous segment not captured] 41836 → 80 [FIN, ACK] Seq=2 Ack=2 Win=65535 L
   35 37.380030537  10.0.2.15   44.228.249.3   TCP      76       0 57192 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=156563153 TSecr=0
   38 37.740321675  10.0.2.15   44.228.249.3   TCP      56       0 57192 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
   39 40.698989102  10.0.2.15   44.228.249.3   HTTP     442    386 GET /login.php HTTP/1.1
   42 41.198024507  10.0.2.15   44.228.249.3   TCP      56       0 57192 → 80 [ACK] Seq=387 Ack=2749 Win=65535 Len=0
   63 49.642481948  10.0.2.15   44.228.249.3   HTTP     580    524 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
   66 50.001097862  10.0.2.15   44.228.249.3   TCP      56       0 57192 → 80 [ACK] Seq=911 Ack=5670 Win=65535 Len=0
   87 60.011710195  10.0.2.15   44.228.249.3   TCP      56       0 [TCP Keep-Alive] 57192 → 80 [ACK] Seq=910 Ack=5670 Win=65535 Len=0
   89 70.251915350  10.0.2.15   44.228.249.3   TCP      56       0 [TCP Keep-Alive] 57192 → 80 [ACK] Seq=910 Ack=5670 Win=65535 Len=0
   91 80.491709064  10.0.2.15   44.228.249.3   TCP      56       0 [TCP Keep-Alive] 57192 → 80 [ACK] Seq=910 Ack=5670 Win=65535 Len=0
   93 90.731796515  10.0.2.15   44.228.249.3   TCP      56       0 [TCP Keep-Alive] 57192 → 80 [ACK] Seq=910 Ack=5670 Win=65535 Len=0
   97 100.971571169 10.0.2.15   44.228.249.3   TCP      56       0 [TCP Keep-Alive] 57192 → 80 [ACK] Seq=910 Ack=5670 Win=65535 Len=0
  131 111.211806342 10.0.2.15   44.228.249.3   TCP      56       0 [TCP Keep-Alive] 57192 → 80 [ACK] Seq=910 Ack=5670 Win=65535 Len=0
  142 115.034984320 10.0.2.15   44.228.249.3   TCP      56       0 57192 → 80 [FIN, ACK] Seq=911 Ack=5671 Win=65535 Len=0
```

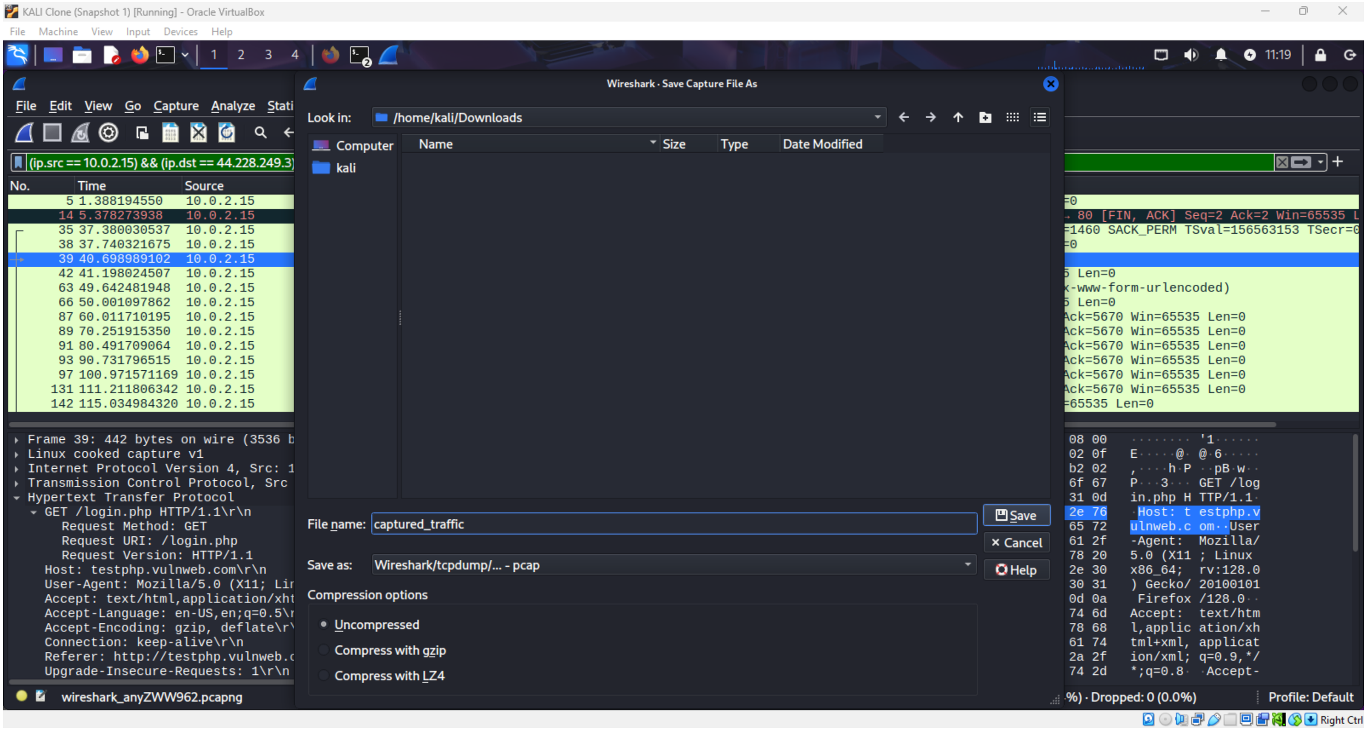
```
› Frame 39: 442 bytes on wire (3536 bits), 442 bytes captured (3536 bits) on interface
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 10.0.2.15, Dst: 44.228.249.3
› Transmission Control Protocol, Src Port: 57192, Dst Port: 80, Seq: 1, Ack: 1, Len: 38
⌄ Hypertext Transfer Protocol
  ⌄ GET /login.php HTTP/1.1\r\n
      Request Method: GET
      Request URI: /login.php
      Request Version: HTTP/1.1
    Host: testphp.vulnweb.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Referer: http://testphp.vulnweb.com/\r\n
    Upgrade-Insecure-Requests: 1\r\n
```

```
0000  00 04 00 01 00 06 08 00  27 31 e8 ab 00 00 08 00   ········ '1······
0010  45 00 01 aa d0 88 40 00  40 06 36 cf 0a 00 02 0f   E·····@· @·6·····
0020  2c e4 f9 03 df 68 00 50  1d e7 70 42 03 77 b2 02   ,····h·P ··pB·w··
0030  50 18 fa f0 33 93 00 00  47 45 54 20 2f 6c 6f 67   P···3··· GET /log
0040  69 6e 2e 70 68 70 20 48  54 54 50 2f 31 2e 31 0d   in.php H TTP/1.1·
0050  0a 48 6f 73 74 3a 20 74  65 73 74 70 68 70 2e 76   ·Host: t estphp.v
0060  75 6c 6e 77 65 62 2e 63  6f 6d 0d 0a 55 73 65 72   ulnweb.c om··User
0070  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
0080  35 2e 30 20 28 58 31 31  3b 20 4c 69 6e 75 78 20   5.0 (X11 ; Linux 
0090  78 38 36 5f 36 34 3b 20  72 76 3a 31 32 38 2e 30   x86_64;  rv:128.0
00a0  29 20 47 65 63 6b 6f 2f  32 30 31 30 30 31 30 31   ) Gecko/ 20100101
00b0  20 46 69 72 65 66 6f 78  2f 31 32 38 2e 30 0d 0a    Firefox /128.0··
00c0  41 63 63 65 70 74 3a 20  74 65 78 74 2f 68 74 6d   Accept:  text/htm
00d0  6c 2c 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 68   l,applic ation/xh
00e0  74 6d 6c 2b 78 6d 6c 2c  61 70 70 6c 69 63 61 74   tml+xml, applicat
00f0  69 6f 6e 2f 78 6d 6c 3b  71 3d 30 2e 39 2c 2a 2f   ion/xml; q=0.9,*/
0100  2a 3b 71 3d 30 2e 38 0d  0a 41 63 63 65 70 74 2d   *;q=0.8· ·Accept-
```

wireshark_anyZWW962.pcapng | Packets: 143 · Displayed: 15 (10.5%) | Profile: Default

This capture filters traffic between two specific IP addresses: the Kali VM and an external web server. The filtered display isolates only relevant communication, eliminating background noise. It reveals HTTP GET requests such as '/login.php' and POST requests like '/userinfo.php'. This kind of targeted filtering is useful for analyzing user sessions or investigating malicious activity. By inspecting packet contents, we can see what kind of data is being transmitted, such as credentials or form inputs. HTTP headers show browser and request details, while the TCP stream maintains context. Security analysts often use such filtering to correlate actions with user behavior. This approach is also used in forensic investigations to identify how a system was accessed or what data was exchanged.

Saving Capture in .pcap Format



KALI Clone (Snapshot 1) [Running] - Oracle VirtualBox

Wireshark · Save Capture File As

Look in: /home/kali/Downloads

Computer
kali

Name	Size	Type	Date Modified

File name: captured_traffic

Save as: Wireshark/tcpdump/... - pcap

Compression options

- ● Uncompressed
- ○ Compress with gzip
- ○ Compress with LZ4

Save · Cancel · Help

(ip.src == 10.0.2.15) && (ip.dst == 44.228.249.3)

```
No.  Time             Source
  5 1.388194550    10.0.2.15
 14 5.378273938    10.0.2.15
 35 37.380030537   10.0.2.15
 38 37.740321675   10.0.2.15
 39 40.698989102   10.0.2.15
 42 41.198024507   10.0.2.15
 63 49.642481948   10.0.2.15
 66 50.001097862   10.0.2.15
 87 60.011710195   10.0.2.15
 89 70.251915350   10.0.2.15
 91 80.491709064   10.0.2.15
 93 90.731796515   10.0.2.15
 97 100.971571169  10.0.2.15
131 111.211806342  10.0.2.15
142 115.034984320  10.0.2.15
```

```
=0
→ 80 [FIN, ACK] Seq=2 Ack=2 Win=65535 L
=1460 SACK_PERM TSval=156563153 TSecr=0
=0

5 Len=0
x-www-form-urlencoded)
5 Len=0
Ack=5670 Win=65535 Len=0
Ack=5670 Win=65535 Len=0
Ack=5670 Win=65535 Len=0
Ack=5670 Win=65535 Len=0
Ack=5670 Win=65535 Len=0
Ack=5670 Win=65535 Len=0
=65535 Len=0
```

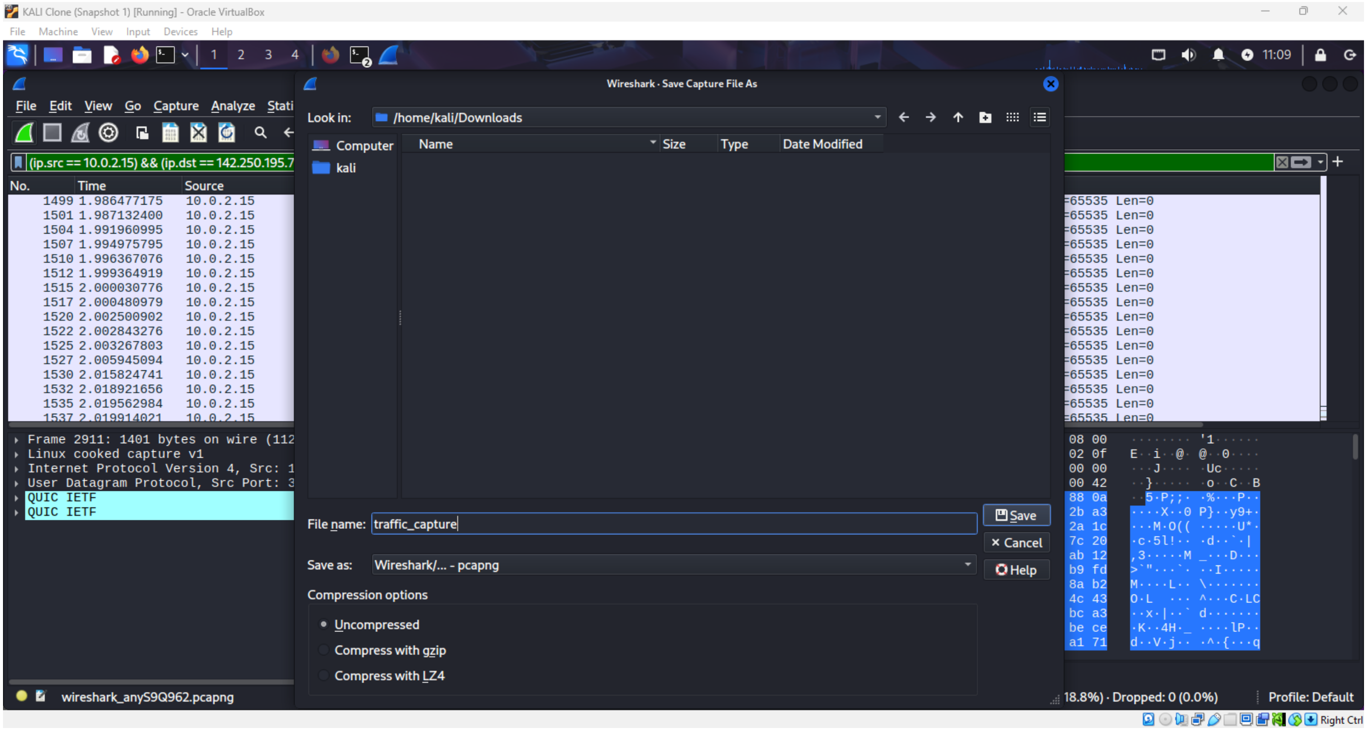
```
▸ Frame 39: 442 bytes on wire (3536 b
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 1
▸ Transmission Control Protocol, Src
▾ Hypertext Transfer Protocol
  ▾ GET /login.php HTTP/1.1\r\n
      Request Method: GET
      Request URI: /login.php
      Request Version: HTTP/1.1
    Host: testphp.vulnweb.com\r\n
    User-Agent: Mozilla/5.0 (X11; Lir
    Accept: text/html,application/xht
    Accept-Language: en-US,en;q=0.5\r
    Accept-Encoding: gzip, deflate\r\
    Connection: keep-alive\r\n
    Referer: http://testphp.vulnweb.c
    Upgrade-Insecure-Requests: 1\r\n
```

```
08 00   ········ '1······
02 0f   E·····@· @·6·····
b2 02   ,····h·P ··pB·w··
6f 67   P···3··· GET /log
31 0d   in.php H TTP/1.1·
2e 76   ·Host: t estphp.v
65 72   ulnweb.c om··User
61 2f   -Agent:  Mozilla/
78 20   5.0 (X11 ; Linux
2e 30   x86_64;  rv:128.0
30 31   ) Gecko/ 20100101
0d 0a    Firefox /128.0··
74 6d   Accept:  text/htm
78 68   l,applic ation/xh
61 74   tml+xml, applicat
2a 2f   ion/xml; q=0.9,*/
74 2d   *;q=0.8·  ·Accept-
```

wireshark_anyZWW962.pcapng · %) · Dropped: 0 (0.0%) · Profile: Default

Here, Wireshark is used to save a capture session in .pcap format. The file is named 'captured_traffic' and saved in the /home/kali/Downloads directory. The .pcap format is a standard across tools like tcpdump, Wireshark, and Snort. Saving captures allows offline analysis, which is important for large datasets or forensic archiving. Uncompressed saving ensures maximum compatibility with other tools. The interface allows selecting other formats too, but .pcap remains the most used due to its widespread support. Proper file naming and organization help during multi-session investigations. Saving the file ensures that the evidence or captured data is preserved for reports, compliance, or further inspection.

Saving Capture in .pcapng Format



Wireshark · Save Capture File As

Look in: /home/kali/Downloads

File name: traffic_capture

Save as: Wireshark/... - pcapng

Compression options

- ● Uncompressed
- ○ Compress with gzip
- ○ Compress with LZ4

```
(ip.src == 10.0.2.15) && (ip.dst == 142.250.195.7
No.   Time         Source
 1499 1.986477175  10.0.2.15
 1501 1.987132400  10.0.2.15
 1504 1.991960995  10.0.2.15
 1507 1.994975795  10.0.2.15
 1510 1.996367076  10.0.2.15
 1512 1.999364919  10.0.2.15
 1515 2.000030776  10.0.2.15
 1517 2.000480979  10.0.2.15
 1520 2.002500902  10.0.2.15
 1522 2.002843276  10.0.2.15
 1525 2.003267803  10.0.2.15
 1527 2.005945094  10.0.2.15
 1530 2.015824741  10.0.2.15
 1532 2.018921656  10.0.2.15
 1535 2.019562984  10.0.2.15
 1537 2.019914021  10.0.2.15

Frame 2911: 1401 bytes on wire (112
Linux cooked capture v1
Internet Protocol Version 4, Src: 1
User Datagram Protocol, Src Port: 3
QUIC IETF
QUIC IETF
```

wireshark_anyS9Q962.pcapng · 18.8%) · Dropped: 0 (0.0%) · Profile: Default

KALI Clone (Snapshot 1) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
11:09
File Edit View Go Capture Analyze Stati
Computer
kali
Name
Size
Type
Date Modified
Save
Cancel
Help
Right Ctrl

In this case, the capture is being saved as a .pcapng file, an enhanced version of .pcap. This format supports extended metadata like interface stats, annotations, and custom blocks. It is recommended for modern analysis as it allows multi-interface capture in a single file. The capture is named 'traffic_capture' and stored in the default Downloads folder. Compression options like gzip and LZ4 are also available, useful for reducing file size during long sessions. The .pcapng format is supported by most modern network tools. Saving in this format allows interoperability and richer session context. This screen shows best practices for export settings in Wireshark.