

Capturing HTTP Credentials using Wireshark



**Model Institute of Engineering & Technology (Autonomous) Permanently
Affiliated to the University of Jammu Accredited by NAAC with “A” Grade
Jammu, India 2025**

Capturing HTTP Credentials using Wireshark

Environment: Kali Linux (Attacker), Windows
Server (Victim)

Tools: Wireshark, Browser, ping

Author: Adarshvardhan singh

Date: 26-07-2025



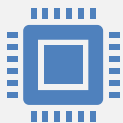
Objective



- Demonstrate how unencrypted HTTP login credentials can be captured over the network.



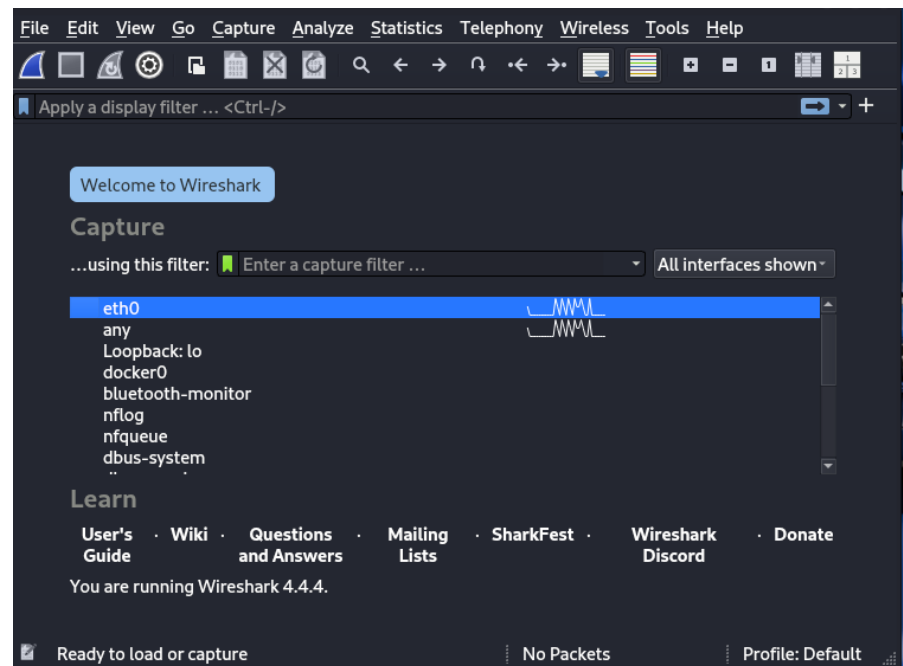
- Use Wireshark on Kali Linux to monitor traffic from a Windows machine.



- Analyze packets to retrieve sensitive data like usernames and passwords.

Starting Wireshark

- - Launch Wireshark with `sudo wireshark`.
- - Select the `eth0` interface to capture packets.



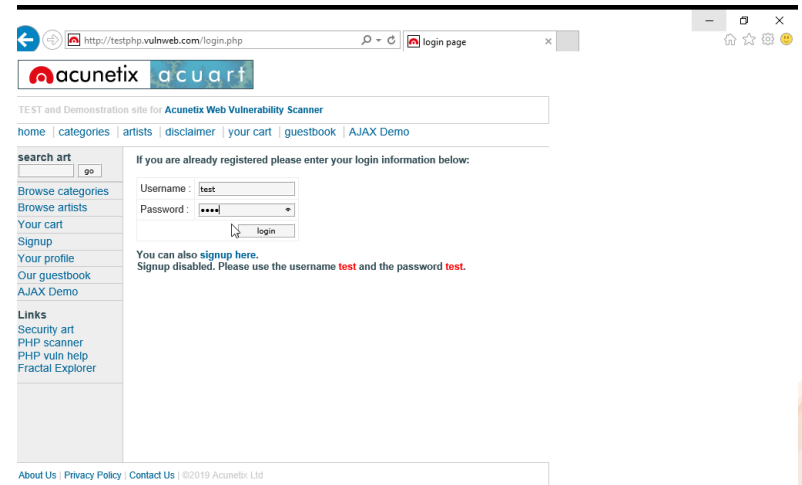
Confirming Target is Live

- - Use `ping 192.168.9.129` to verify the Windows machine is active.
- - Replies confirm it's reachable for packet sniffing.

```
root@kali: /home/kali
File Actions Edit View Help
(root@kali)-[/home/kali]
# ping 192.168.9.129
PING 192.168.9.129 (192.168.9.129) 56(84) bytes of data.
64 bytes from 192.168.9.129: icmp_seq=1 ttl=128 time=1.55 ms
64 bytes from 192.168.9.129: icmp_seq=2 ttl=128 time=1.75 ms
64 bytes from 192.168.9.129: icmp_seq=3 ttl=128 time=1.37 ms
64 bytes from 192.168.9.129: icmp_seq=4 ttl=128 time=0.990 ms
64 bytes from 192.168.9.129: icmp_seq=5 ttl=128 time=1.34 ms
64 bytes from 192.168.9.129: icmp_seq=6 ttl=128 time=0.963 ms
64 bytes from 192.168.9.129: icmp_seq=7 ttl=128 time=1.13 ms
64 bytes from 192.168.9.129: icmp_seq=8 ttl=128 time=0.304 ms
64 bytes from 192.168.9.129: icmp_seq=9 ttl=128 time=1.02 ms
64 bytes from 192.168.9.129: icmp_seq=10 ttl=128 time=1.57 ms
64 bytes from 192.168.9.129: icmp_seq=11 ttl=128 time=1.10 ms
64 bytes from 192.168.9.129: icmp_seq=12 ttl=128 time=0.917 ms
64 bytes from 192.168.9.129: icmp_seq=13 ttl=128 time=0.654 ms
64 bytes from 192.168.9.129: icmp_seq=14 ttl=128 time=1.28 ms
64 bytes from 192.168.9.129: icmp_seq=15 ttl=128 time=1.16 ms
64 bytes from 192.168.9.129: icmp_seq=16 ttl=128 time=1.13 ms
^[[B^[[B^[[B^[[B^[[B64 bytes from 192.168.9.129: icmp_seq=17 ttl=128 time
=0.626 ms
64 bytes from 192.168.9.129: icmp_seq=18 ttl=128 time=1.65 ms
```

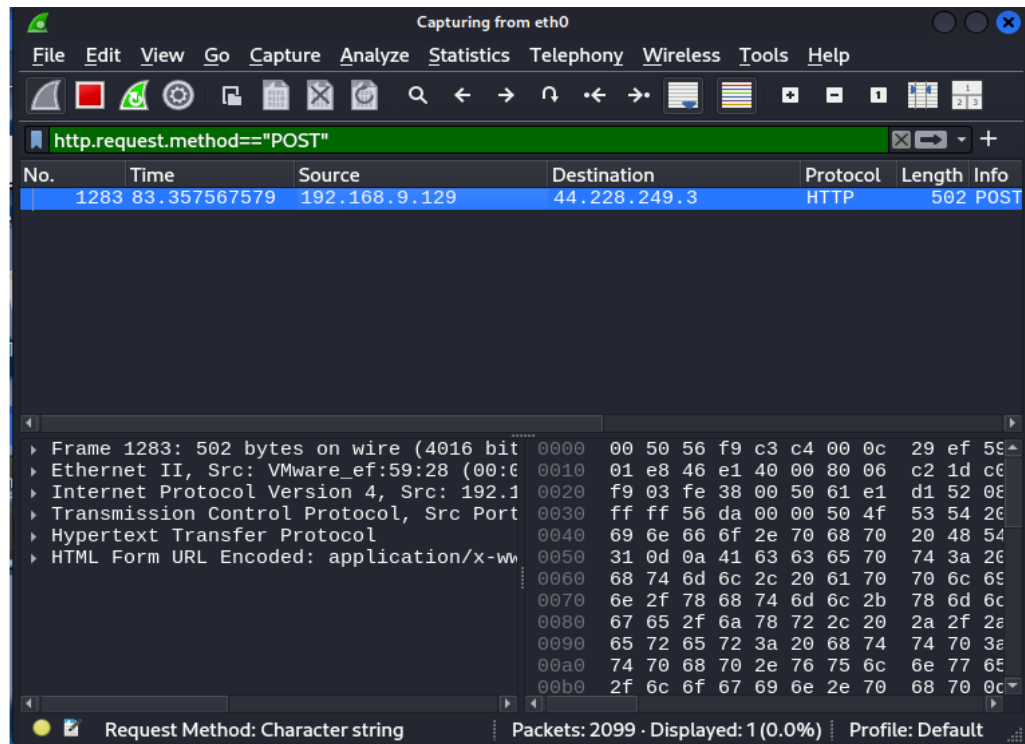
Target Accesses Vulnerable Web Page

- - Victim logs into
`http://testphp.vulnweb.com`.
- - Credentials: Username = test,
- Password = test
- - Site uses HTTP (insecure).



Capture HTTP POST Packet

- - Apply Wireshark filter:
- `http.request.method == "POST"`
- - Observe traffic from 192.168.9.129 to 44.228.249.3



Analyze Packet Details

- Packet reveals: `uname=test&pass=test`
- Confirms credentials sent in plain text (insecure HTTP).

The image shows a Wireshark packet capture interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. A green filter bar at the top of the packet list contains the text `http.request.method=="POST"`. The packet list on the left shows a single packet, No. 1283, at time 83.357567579, from source 192.168.9.129. The packet details pane on the right shows the following structure:

- Ethernet II, Src: VMware_ef:59:28 (00:0c:29:ef:59:28), Dst: 44:22:8b:12:1b:05
- Internet Protocol Version 4, Src: 192.168.9.129, Dst: 44.228.12.1
- Transmission Control Protocol, Src Port: 65080, Dst Port: 80
- Hypertext Transfer Protocol
- HTML Form URL Encoded: application/x-www-form-urlencoded
 - Form item: "uname" = "test"
 - Form item: "pass" = "test"

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII. The first 10 bytes are 00 50 56 f9 c3 c4 00 0c 29 ef, which correspond to the Ethernet II frame type (0x0000) and the source MAC address (00:0c:29:ef:59:28). The ASCII column shows the corresponding characters: PV.

Wireshark Launch Command

- Command used to start Wireshark with root privileges:
- `sudo wireshark`

```
(root@kali)-[/home/kali]  
# sudo wireshark
```

Conclusion & Recommendation

- - Captured credentials show the danger of unencrypted HTTP login forms.
- - Recommendations:
 - Use HTTPS to secure communications.
 - Avoid logging into sensitive accounts on insecure networks.
 - Monitor internal traffic for potential data leaks.

