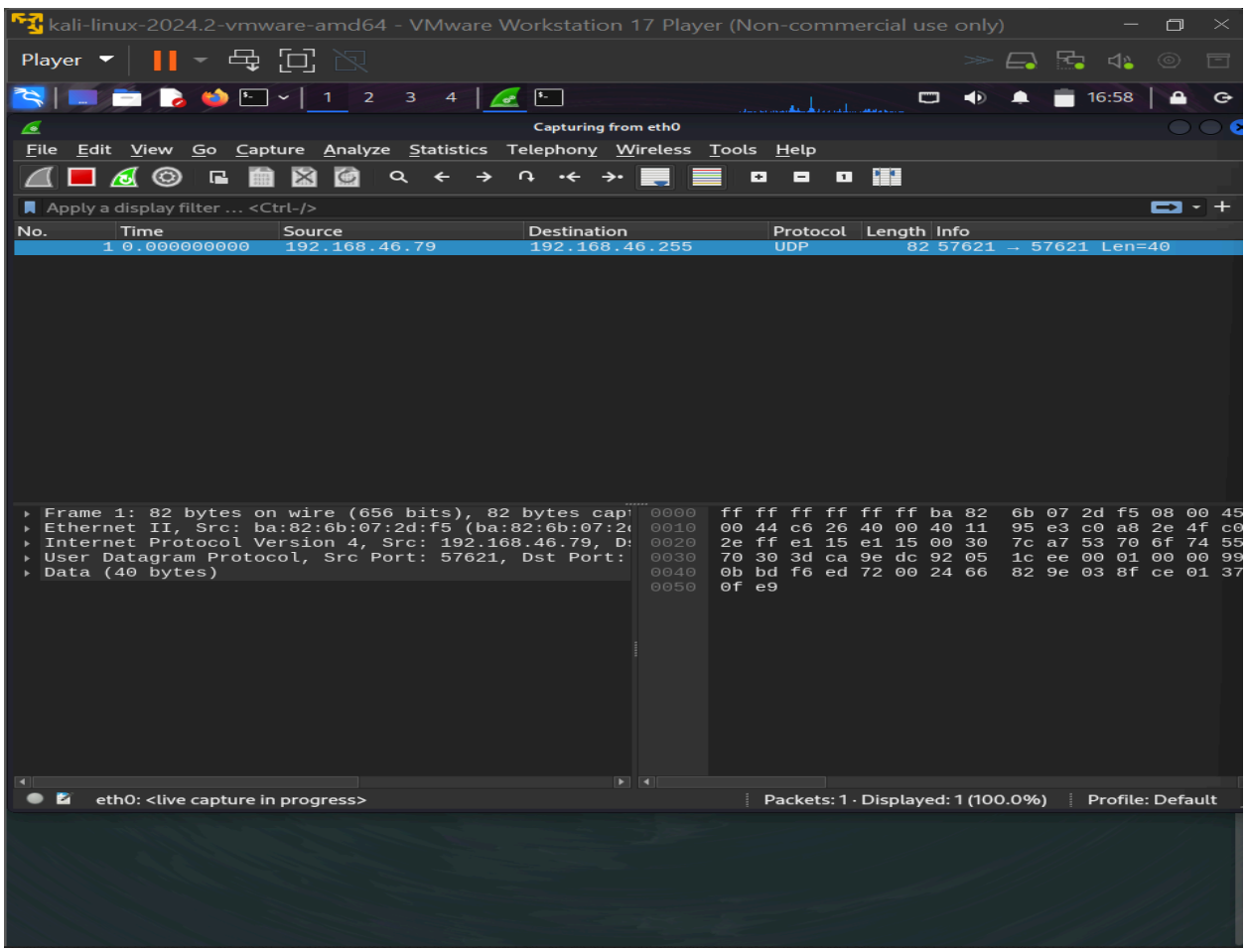
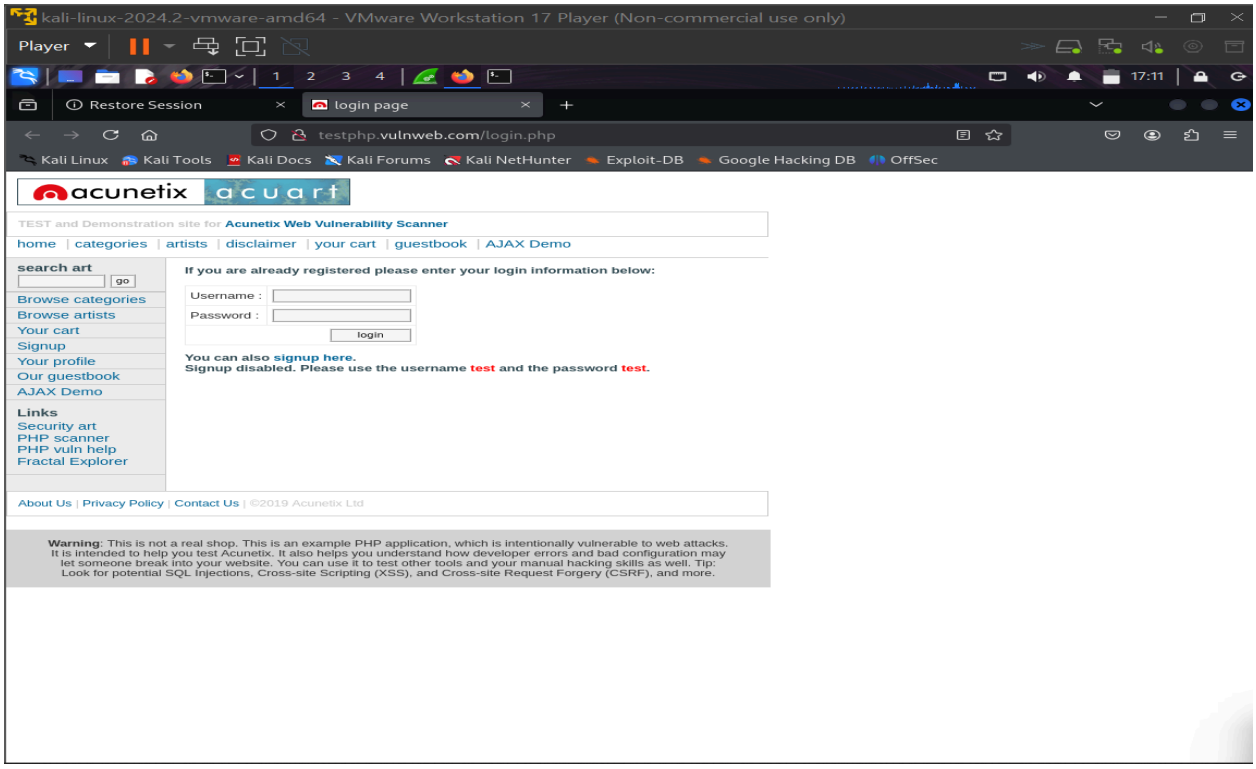
CAPTURING CREDENTIALS SUBMITTED THROUGH HTTP WITH WIRESHARK

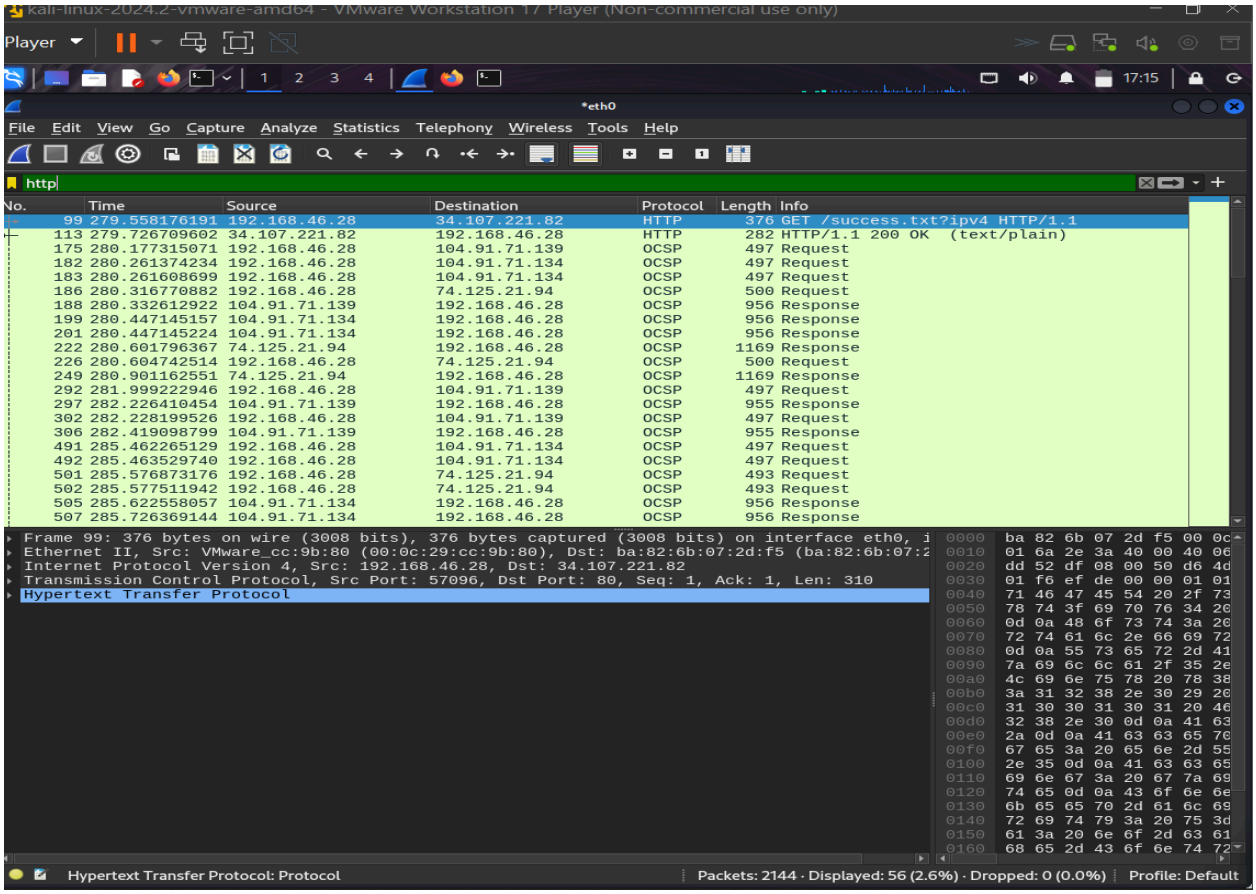
Tools : KALI LINUX, WIRESHARK
Site : http://testphp.vulnweb.com/login.php

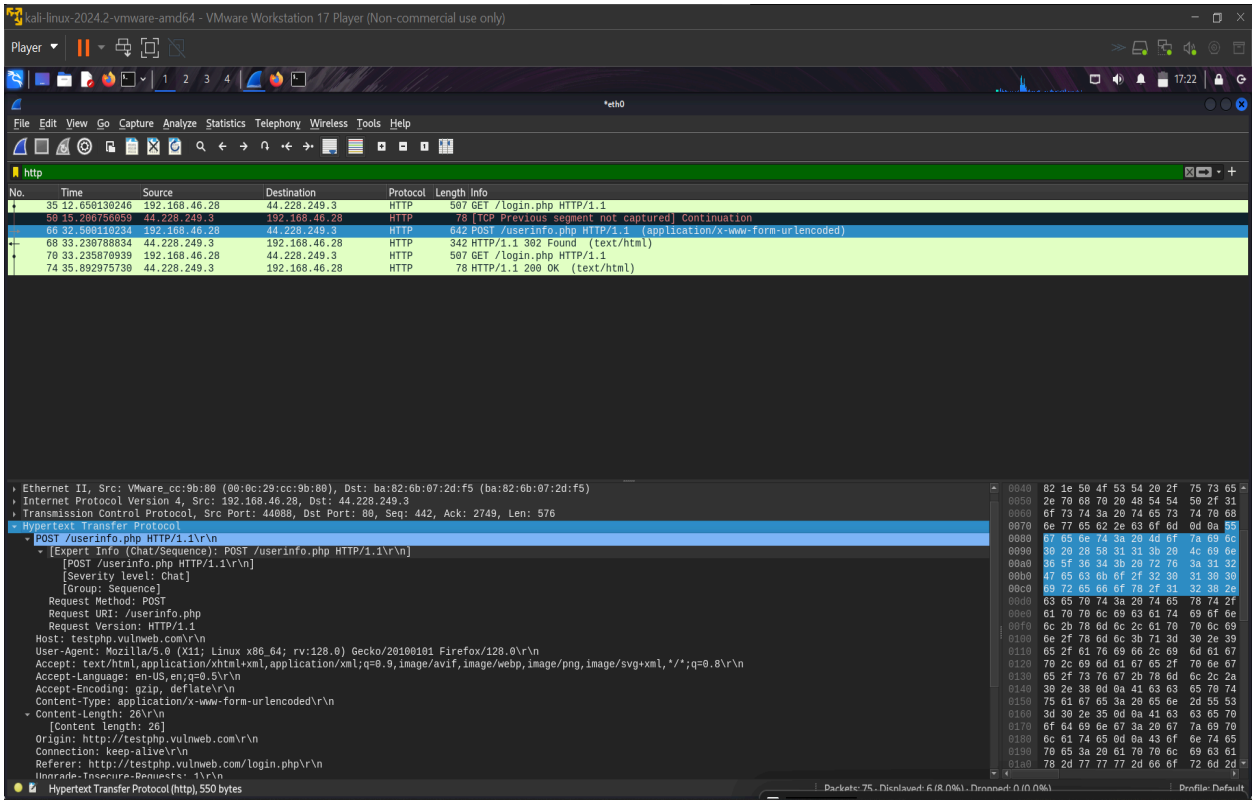
Wireshark is a popular open-source network protocol analyzer used for capturing and inspecting network traffic in real-time. It allows you to see what's happening on a network at a microscopic level and is widely used for troubleshooting, security analysis, and protocol development.

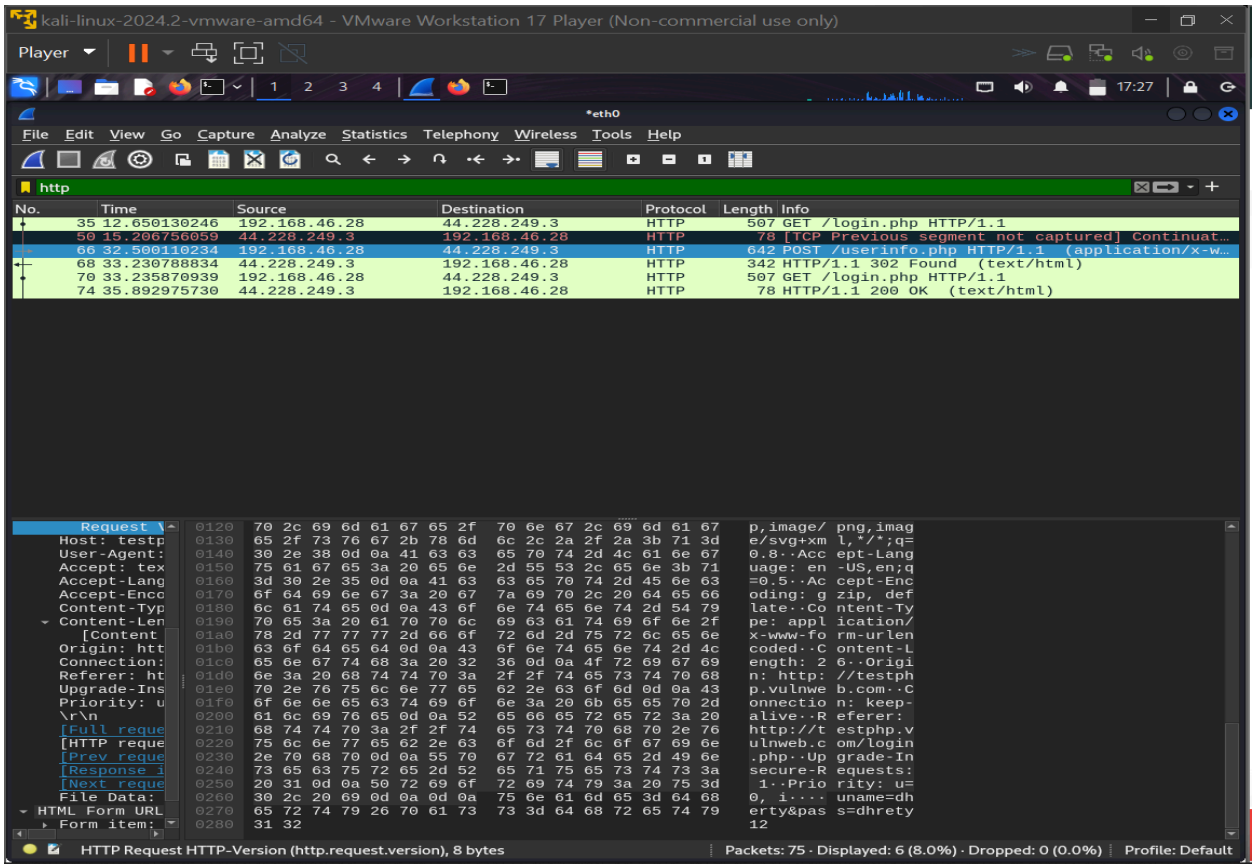
Input from kali :

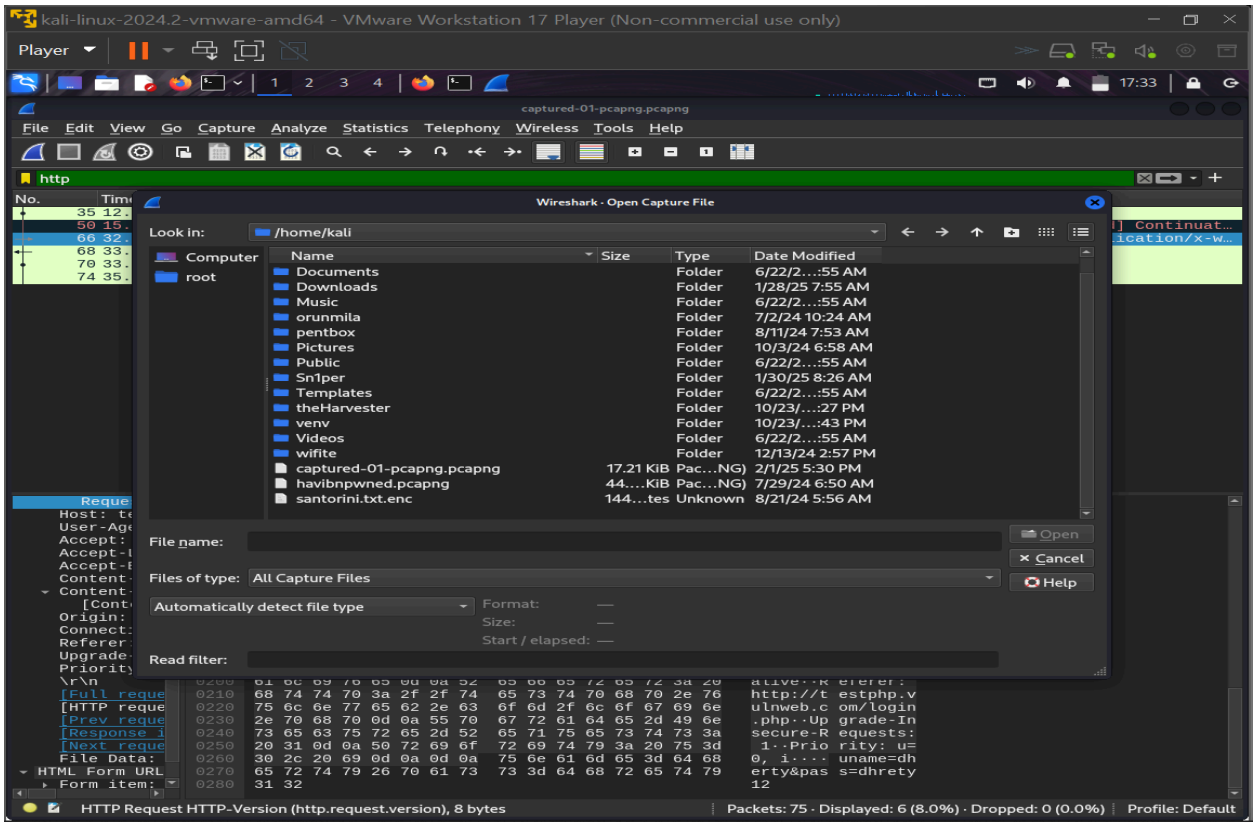


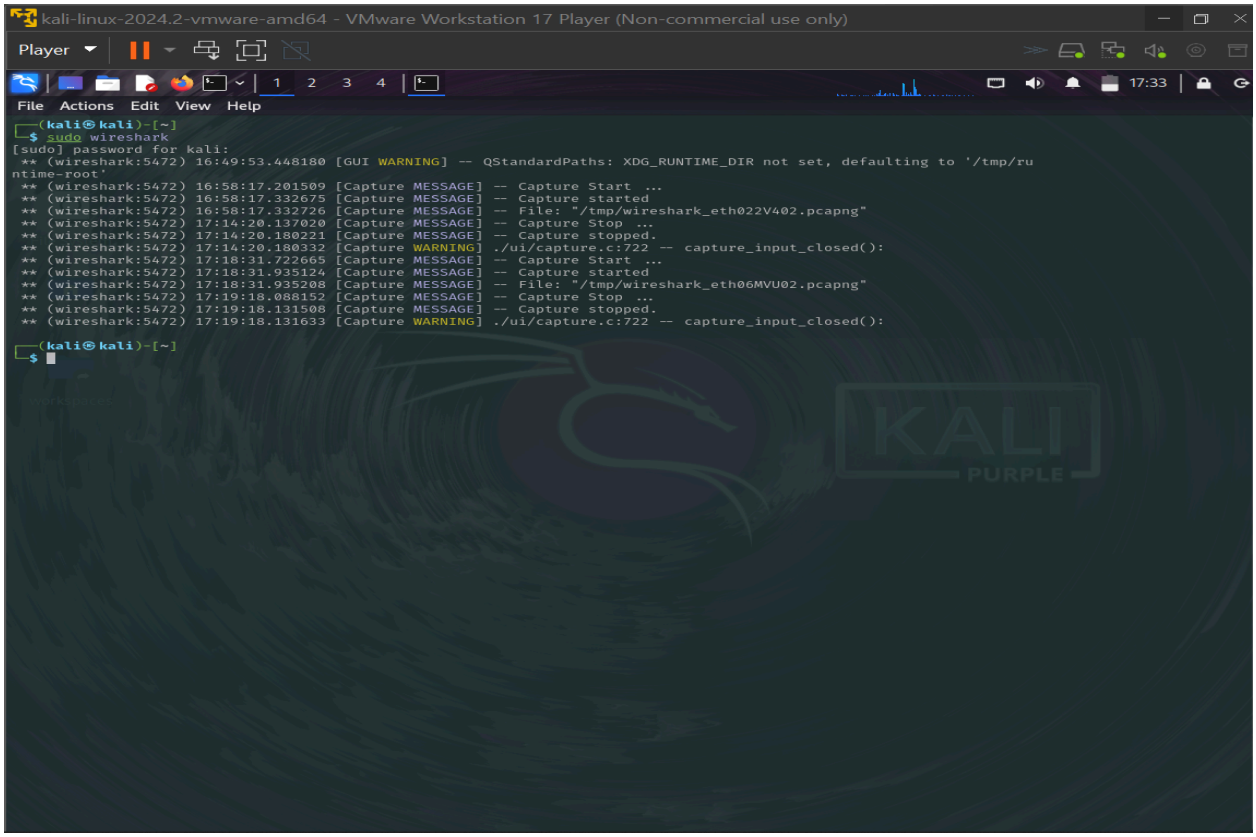

kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
Restore Session
login page
testphp.vulnweb.com/login.php
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
acunetix
acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
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
If you are already registered please enter your login information below:
Username :
Password :
login
You can also signup here.
Signup disabled. Please use the username test and the password test.
About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd
Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
99 279.558176191 192.168.46.28 34.107.221.82 HTTP 376 GET /success.txt?ipv4 HTTP/1.1
113 279.726709602 34.107.221.82 192.168.46.28 HTTP 282 HTTP/1.1 200 OK (text/plain)
175 280.177315071 192.168.46.28 104.91.71.139 OCSP 497 Request
182 280.261374234 192.168.46.28 104.91.71.134 OCSP 497 Request
183 280.261608699 192.168.46.28 104.91.71.134 OCSP 497 Request
186 280.316770882 192.168.46.28 74.125.21.94 OCSP 500 Request
188 280.332612922 104.91.71.139 192.168.46.28 OCSP 956 Response
199 280.447145157 104.91.71.134 192.168.46.28 OCSP 956 Response
201 280.447145224 104.91.71.134 192.168.46.28 OCSP 956 Response
222 280.601796367 74.125.21.94 192.168.46.28 OCSP 1169 Response
226 280.604742514 192.168.46.28 74.125.21.94 OCSP 500 Request
249 280.901162551 74.125.21.94 192.168.46.28 OCSP 1169 Response
292 281.999222946 192.168.46.28 104.91.71.139 OCSP 497 Request
297 282.226410454 104.91.71.139 192.168.46.28 OCSP 955 Response
302 282.228199526 192.168.46.28 104.91.71.139 OCSP 497 Request
306 282.419098799 104.91.71.139 192.168.46.28 OCSP 955 Response
491 285.462265129 192.168.46.28 104.91.71.134 OCSP 497 Request
492 285.463529740 192.168.46.28 104.91.71.134 OCSP 497 Request
501 285.576873176 192.168.46.28 74.125.21.94 OCSP 493 Request
502 285.577511942 192.168.46.28 74.125.21.94 OCSP 493 Request
505 285.622558057 104.91.71.134 192.168.46.28 OCSP 956 Response
507 285.726369144 104.91.71.134 192.168.46.28 OCSP 956 Response
Frame 99: 376 bytes on wire (3008 bits), 376 bytes captured (3008 bits) on interface eth0, i
Ethernet II, Src: VMware_cc:9b:80 (00:0c:29:cc:9b:80), Dst: ba:82:6b:07:2d:f5 (ba:82:6b:07:2
Internet Protocol Version 4, Src: 192.168.46.28, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 57096, Dst Port: 80, Seq: 1, Ack: 1, Len: 310
Hypertext Transfer Protocol
Hypertext Transfer Protocol: Protocol
Packets: 2144 · Displayed: 56 (2.6%) · Dropped: 0 (0.0%)
Profile: Default


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
35 12.650130246 192.168.46.28 44.228.249.3 HTTP 507 GET /login.php HTTP/1.1
50 15.206756059 44.228.249.3 192.168.46.28 HTTP 78 [TCP Previous segment not captured] Continuation
66 32.500110234 192.168.46.28 44.228.249.3 HTTP 642 POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
68 33.230788834 44.228.249.3 192.168.46.28 HTTP 342 HTTP/1.1 302 Found (text/html)
70 33.235870939 192.168.46.28 44.228.249.3 HTTP 507 GET /login.php HTTP/1.1
74 35.892975730 44.228.249.3 192.168.46.28 HTTP 78 HTTP/1.1 200 OK (text/html)
Ethernet II, Src: VMware_cc:9b:80 (00:0c:29:cc:9b:80), Dst: ba:82:6b:07:2d:f5 (ba:82:6b:07:2d:f5)
Internet Protocol Version 4, Src: 192.168.46.28, Dst: 44.228.249.3
Transmission Control Protocol, Src Port: 44088, Dst Port: 80, Seq: 442, Ack: 2749, Len: 576
Hypertext Transfer Protocol
POST /userinfo.php HTTP/1.1\r\n
[Expert Info (Chat/Sequence): POST /userinfo.php HTTP/1.1\r\n]
[POST /userinfo.php HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: POST
Request URI: /userinfo.php
Request Version: HTTP/1.1
Host: testphp.vulnweb.com\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 26\r\n
[Content length: 26]
Origin: http://testphp.vulnweb.com\r\n
Connection: keep-alive\r\n
Referer: http://testphp.vulnweb.com/login.php\r\n
Hypertext Transfer Protocol (http), 550 bytes


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
35 12.650130246 192.168.46.28 44.228.249.3 HTTP 507 GET /login.php HTTP/1.1
50 15.206756059 44.228.249.3 192.168.46.28 HTTP 78 [TCP Previous segment not captured] Continuat…
66 32.500110234 192.168.46.28 44.228.249.3 HTTP 642 POST /userinfo.php HTTP/1.1 (application/x-w…
68 33.230788834 44.228.249.3 192.168.46.28 HTTP 342 HTTP/1.1 302 Found (text/html)
70 33.235870939 192.168.46.28 44.228.249.3 HTTP 507 GET /login.php HTTP/1.1
74 35.892975730 44.228.249.3 192.168.46.28 HTTP 78 HTTP/1.1 200 OK (text/html)
HTTP Request HTTP-Version (http.request.version), 8 bytes
Packets: 75 · Displayed: 6 (8.0%) · Dropped: 0 (0.0%)
Profile: Default


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
17:33
captured-01-pcapng.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Wireshark · Open Capture File
Look in: /home/kali
Computer
root
Name Size Type Date Modified
Documents Folder 6/22/2...:55 AM
Downloads Folder 1/28/25 7:55 AM
Music Folder 6/22/2...:55 AM
orunmila Folder 7/2/24 10:24 AM
pentbox Folder 8/11/24 7:53 AM
Pictures Folder 10/3/24 6:58 AM
Public Folder 6/22/2...:55 AM
Sn1per Folder 1/30/25 8:26 AM
Templates Folder 6/22/2...:55 AM
theHarvester Folder 10/23/...:27 PM
venv Folder 10/23/...:43 PM
Videos Folder 6/22/2...:55 AM
wifite Folder 12/13/24 2:57 PM
captured-01-pcapng.pcapng 17.21 KiB Pac...NG) 2/1/25 5:30 PM
havibnpwned.pcapng 44....KiB Pac...NG) 7/29/24 6:50 AM
santorini.txt.enc 144...tes Unknown 8/21/24 5:56 AM
File name:
Files of type: All Capture Files
Automatically detect file type
Format:
Size:
Start / elapsed:
Read filter:
Open
Cancel
Help
HTTP Request HTTP-Version (http.request.version), 8 bytes
Packets: 75 · Displayed: 6 (8.0%) · Dropped: 0 (0.0%)
Profile: Default


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
17:33
File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo wireshark
[sudo] password for kali:
** (wireshark:5472) 16:49:53.448180 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:5472) 16:58:17.201509 [Capture MESSAGE] -- Capture Start ...
** (wireshark:5472) 16:58:17.332675 [Capture MESSAGE] -- Capture started
** (wireshark:5472) 16:58:17.332726 [Capture MESSAGE] -- File: "/tmp/wireshark_eth022V402.pcapng"
** (wireshark:5472) 17:14:20.137020 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:5472) 17:14:20.180221 [Capture MESSAGE] -- Capture stopped.
** (wireshark:5472) 17:14:20.180332 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
** (wireshark:5472) 17:18:31.722665 [Capture MESSAGE] -- Capture Start ...
** (wireshark:5472) 17:18:31.935124 [Capture MESSAGE] -- Capture started
** (wireshark:5472) 17:18:31.935208 [Capture MESSAGE] -- File: "/tmp/wireshark_eth06MVU02.pcapng"
** (wireshark:5472) 17:19:18.088152 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:5472) 17:19:18.131508 [Capture MESSAGE] -- Capture stopped.
** (wireshark:5472) 17:19:18.131633 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
(kali㉿kali)-[~]
$

Overview

This experiment will **simulate a real-world attack scenario** in which a malicious actor could intercept login credentials transmitted over an **unencrypted HTTP website**. We will analyze the captured packets to extract usernames and passwords entered into a login form.

Important Note: This guide is for **ethical penetration testing and cybersecurity research** only. Capturing network traffic without authorization is **illegal** and punishable under various cybersecurity laws.

1. Setting Up the Lab Environment

Before we begin, ensure that you have:
Kali Linux installed (either on a virtual machine or bare-metal).
Wireshark installed (it comes pre-installed on Kali Linux).
An active internet connection to access the HTTP website.
Firefox or any other browser to submit login credentials.

2. Launching Wireshark and Starting a Capture

Step 1: Open Wireshark

Wireshark can be launched via the terminal or the graphical interface:

Using the Terminal:
Open a terminal in Kali and enter the following command:

```
sudo wireshark
```

1. The **sudo** command ensures that Wireshark runs with administrative privileges, allowing it to capture all network traffic.
2. **Using the GUI Menu:**
    - Navigate to the **top left menu** in Kali Linux.
    - Hover over **Sniffing & Spoofing**.
    - Click on **Wireshark** to launch the application.

Step 2: Select the Network Interface for Capture

Once Wireshark opens, you need to select the **network interface** you want to monitor:

1. On the **Wireshark main screen**, you will see a list of available network interfaces.
2. Locate and **double-click** on the interface labeled **eth0** (this represents your wired network connection).
    - If using **Wi-Fi**, select **wlan0** instead.
3. Wireshark will now start capturing **live network traffic**.

At this stage, the screen will populate with packets moving through the network. Each row represents a captured packet, containing **source and destination addresses, protocol types, timestamps, and packet details**.

3. Simulating an HTTP Login Submission

To capture **login credentials**, we need to submit them through an **unencrypted** HTTP website.

Step 1: Open an Insecure HTTP Login Page

1. Open **Firefox** or any web browser.

Enter the following URL in the address bar:

```
http://testphp.vulnweb.com/login.php
```

2.
3. Press **Enter** to load the page.

Notice: A **red warning icon** (🔓) will appear in the browser's address bar, indicating that this website is not secure. This means **any data submitted here is transmitted in plaintext**, making it **vulnerable to interception**.

Step 2: Enter Fake Login Credentials

On the login page:

In the **Username** field, enter:
`testuser`

1.

In the **Password** field, enter:
`password123`

2.
3. Click the **Login** button.

Since this is a **test site**, the login attempt will fail. However, the **plaintext credentials** have already been transmitted over the network.

4. Capturing and Extracting Credentials in Wireshark

Now that we have submitted login credentials, we will locate them in Wireshark's captured packets.

Step 1: Stop Packet Capture

1. Switch back to **Wireshark**.
2. Click the **red square** (🟧) in the top-left corner to stop capturing traffic.

Step 2: Apply an HTTP Filter

To isolate HTTP traffic and eliminate unnecessary packets, apply a display filter:

1. Locate the **Display Filter** bar at the top of the Wireshark window.

Type the following filter:
`http`

2.
3. Press **Enter**.

This filter will **remove non-HTTP traffic**, leaving only HTTP packets in the packet list.

Step 3: Identify the Login Request

1. Look for a packet where the **Info** column contains **POST**.
    - **POST** requests are used to send form data (such as login credentials) to a server.
2. Click on the packet to **select it**.

Step 4: Extract Login Credentials

1. In the **Packet Details Pane** (middle section of Wireshark), expand the section labeled **Hypertext Transfer Protocol**.
2. Locate and expand the section labeled **HTML Form URL Encoded**.
3. Inside this section, you will find:
    - **username:** `testuser`
    - **password:** `password123`

Alternatively, you can view the raw data:

1. Scroll down to the **Packet Bytes Pane** (bottom section of Wireshark).
2. Locate the plaintext representation of the captured credentials.

You will see something like:

```
username=testuser&password=password123
```

5. Exporting Captured Data for Further Analysis

Wireshark allows you to **save and analyze captured traffic later**.

1. Click **File** > **Save As**.
2. Choose a filename and save the capture as a **.pcap** file.
3. You can later reopen the capture file in Wireshark for further investigation.

6. Understanding the Security Risks

This exercise demonstrates a **critical security vulnerability**:

- **Any data submitted via an HTTP website is visible in plaintext.**
- Attackers with network access can **easily intercept usernames and passwords**.
- This method works **on public Wi-Fi, office networks, and even compromised home routers**.

How to Protect Against This Attack

✔ **Always use HTTPS:** Ensure websites use **TLS encryption** (indicated by 🔒 in the address bar).
✔ **Use VPNs:** Encrypt all network traffic when on **untrusted networks**.
✔ **Enable HSTS (HTTP Strict Transport Security):** Forces browsers to connect only via HTTPS.
✔ **Monitor Network Traffic:** Use **Wireshark** to regularly inspect traffic for security vulnerabilities.

7. Conclusion

In this detailed lab, we successfully:
✅ Installed and launched **Wireshark**.
✅ Captured HTTP traffic on the **eth0** interface.
✅ Submitted login credentials via an **unencrypted HTTP page**.
✅ **Extracted usernames and passwords** from captured packets.
✅ **Learned about security risks and mitigation strategies**.

This guide highlights **the dangers of insecure HTTP websites** and the **ease of intercepting credentials on an unencrypted network**. Understanding these vulnerabilities allows penetration testers and security professionals to **identify, mitigate, and educate users about these risks**.