ShadowFox Internship

Name: Gujjari Sandeep

Batch: B1

(March to April {extended})

Domain: Cyber-Security

Report On Beginner Level Task

Table of content

1. Task Level: Beginner

- Find all the ports that are open on the website **http://testphp.vulnweb.com**/.

- Brute force the website **http://testphp.vulnweb.com/** and find the directories that are present on the website.

- Make a login on the website http://testphp.vulnweb.com/ and intercept the network traffic using Wireshark and find the credentials that were transferred through the network.

Beginner Level

➢ Find all the ports that are open on the website http://testphp.vulnweb.com/

Task Description:

The task involves identifying all open ports on the website http://testphp.vulnweb.com/.

By determining which ports are open, we gain insight into the services running on the target server. This information is crucial for understanding the attack surface and potential vulnerabilities of the website.

Tool Used: Nmap

Commands Used and steps involved:

1. At first, we need to find the IP address of the website, hence we use the command **nslookup** to find out the IP address of the website.

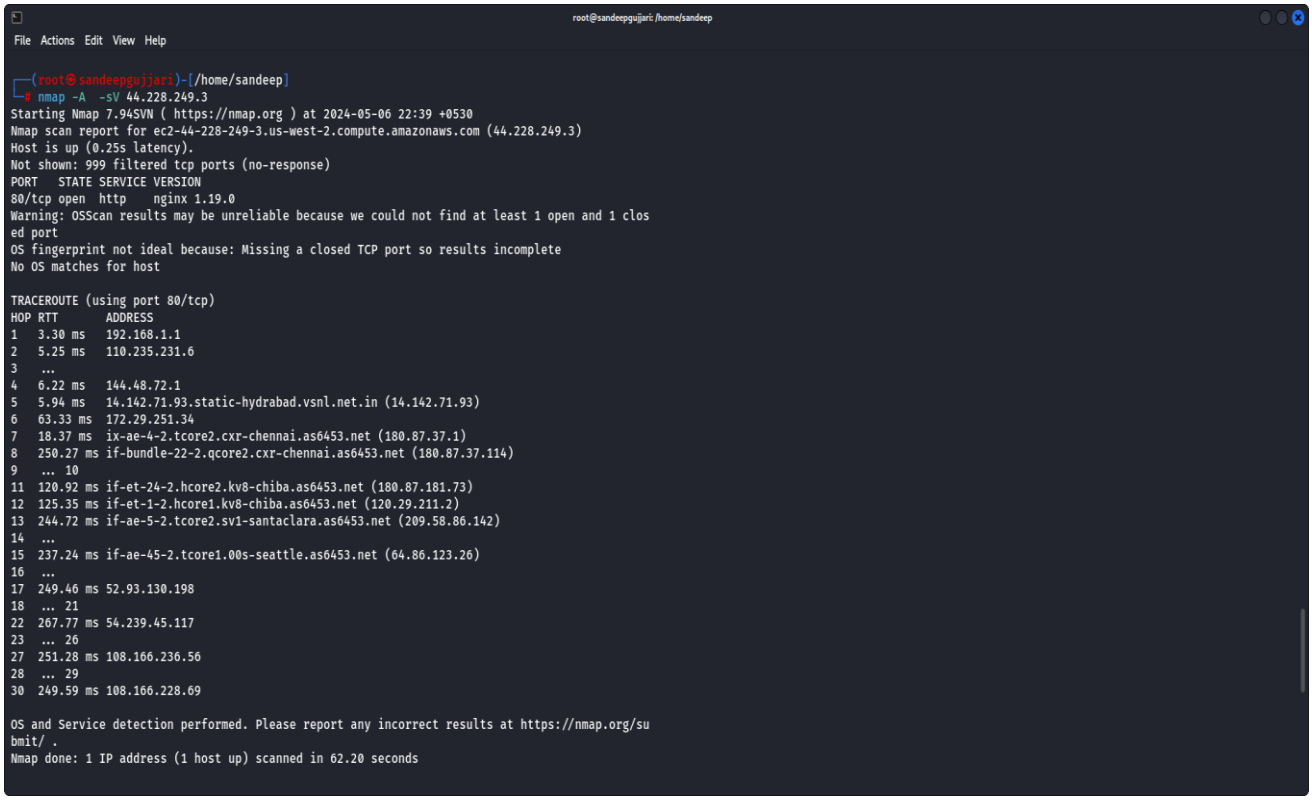
2. Finding the ports using the default command of Nmap

Command Format: nmap [Scan Type(s)] [Options] {target specification}

Given command: nmap -A -sV 44.228.249.3

-A: Enable OS detection, version detection, script scanning, and traceroute.

-sV: Probe open ports to determine service/version info



Result: After executing the Nmap command, it was found that only port 80 is open on the target website http://testphp.vulnweb.com/.

Conclusion:

The Nmap scan revealed that only port 80, the default HTTP port, is open on the website http://testphp.vulnweb.com/. This indicates that the server is primarily serving web content and not running any other accessible services. Using Nmap for port scanning proved effective in identifying the open port and provided valuable information about the target website's network configuration.

Mitigations:

1. Implementing HTTPS: Enabling HTTPS encrypts the data transmitted between the client and the server, making it more challenging for attackers to intercept and analyse the traffic. While HTTPS itself doesn't prevent port scanning, it adds a layer of encryption that protects sensitive information, even if the attacker manages to discover open ports.

2. Web Application Firewall (WAF): A WAF can help detect and block malicious traffic, including port scanning attempts. By setting up rules in the WAF to monitor and block suspicious activities, you can reduce the effectiveness of port scanning tools.

4. Access Control: Proper access controls limit access to sensitive areas of your website and administrative interfaces. By restricting access to authorized users only, you reduce the chances of attackers gaining information about open ports through unauthorized means.

- **Brute force the website http://testphp.vulnweb.com/ and find the directories that are present on the website.**

Task Description:

The task is to discover directories present on the website

http://testphp.vulnweb.com/ using directory brute forcing techniques.

Tool Used: GObuster

Commands Used and steps involved:

1.finding the subdirectories using GObuster.

Command Format:

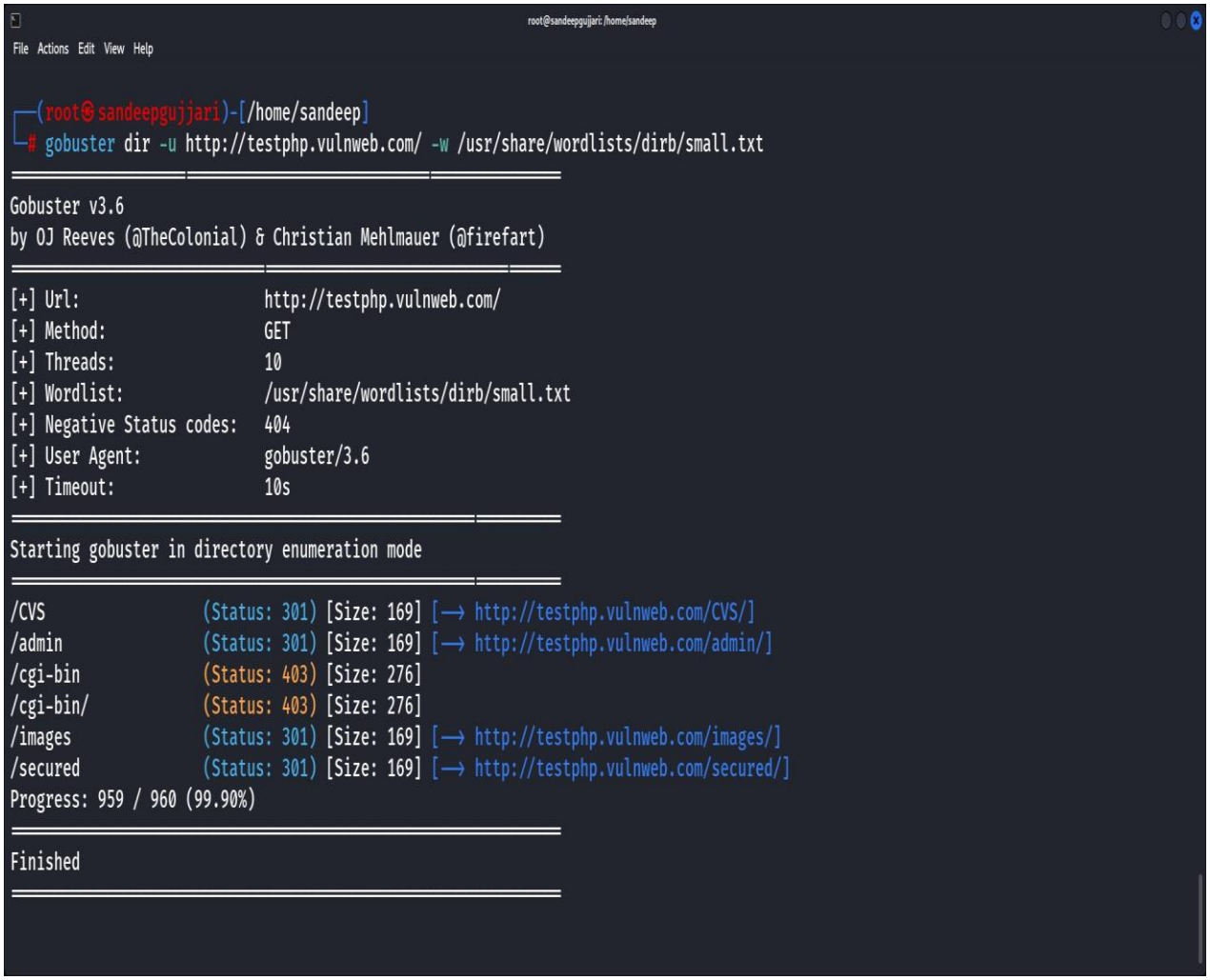
gobuster dir -u [website URL] -w [wordlist path]

Command Used:

```
gobuster dir -u http://testphp.vulnweb.com/ -w /usr/share/wordlists/dirb/small.txt
```

-u tag for the URL of the website.

-w the wordlist path tag.



Result:

gobuster will start brute forcing directories using the provided wordlist and display any discovered directories along with their HTTP status codes.

Conclusion:

By utilizing gobuster for directory brute forcing, potential directories present on the website http://testphp.vulnweb.com/ can be discovered. This process helps in identifying hidden or non-linked directories, which may contain sensitive or vulnerable information.

Mitigation:

1. Rate Limiting: Implement rate limiting mechanisms to restrict the number of HTTP requests that can be made within a certain time frame. This helps prevent automated tools from launching brute-force attacks by slowing down the rate of requests.

2. CAPTCHA Challenges: Introduce CAPTCHA challenges on login pages or any sensitive endpoints to differentiate between human users and automated bots. CAPTCHA challenges require users to solve a challenge, such as identifying objects in images, before gaining access.

3. Account Lockout Policies: Implement account lockout policies to lock user accounts temporarily or permanently after a certain number of failed login attempts. This prevents brute-force attacks by making it difficult for attackers to guess passwords.

4. Strong Authentication Mechanisms: Enforce the use of strong authentication mechanisms, such as multi-factor authentication (MFA) or two-factor authentication (2FA), to add an extra layer of security beyond passwords. This makes it harder for attackers to gain unauthorized access even if they manage to guess a password.

➢ Make a login on the website http://testphp.vulnweb.com/ and intercept the network traffic using Wireshark and find the credentials that were transferred through the network.

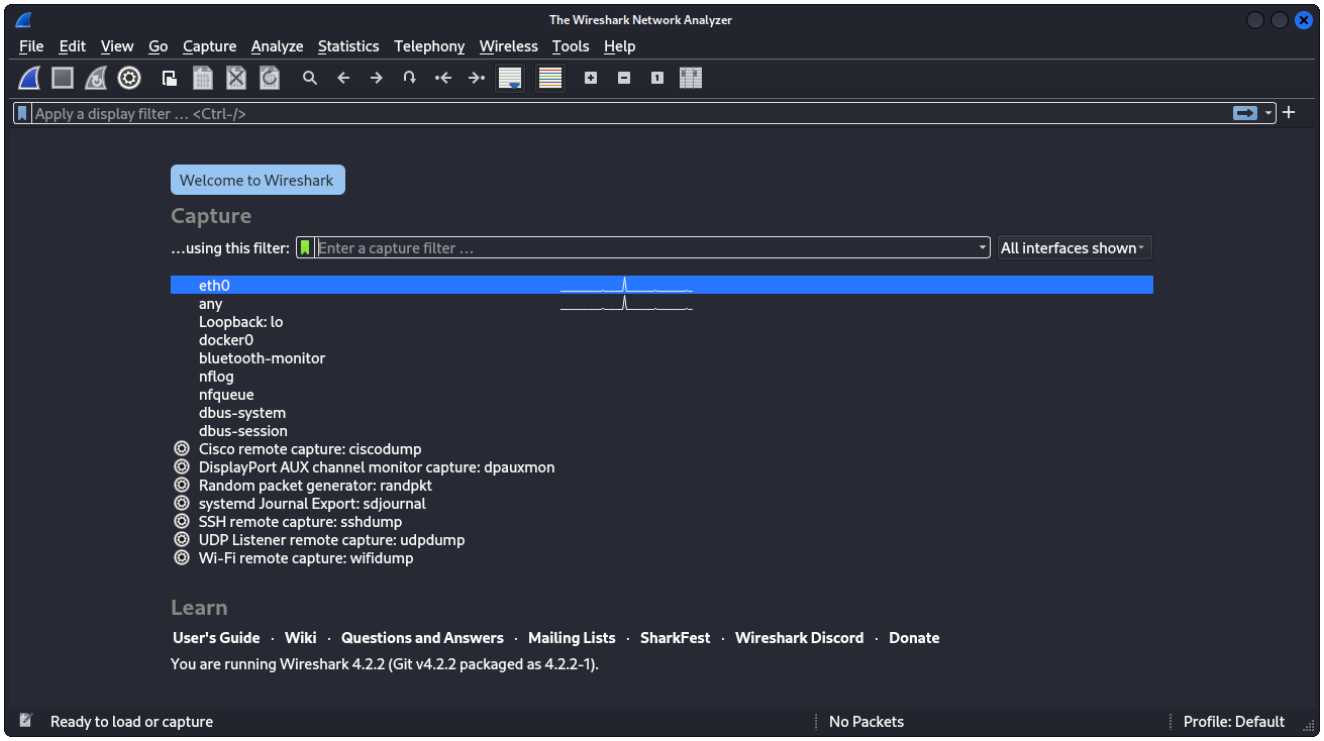
Task Description:

The task is to intercept network traffic using Wireshark during a login attempt on the website http://testphp.vulnweb.com/ and identify any credentials transferred over the network.

Tool Used: Wireshark

Commands Used and steps involved:

1. **Start Network Capture:** Open Wireshark and begin capturing network traffic on the network interface connected to the internet.



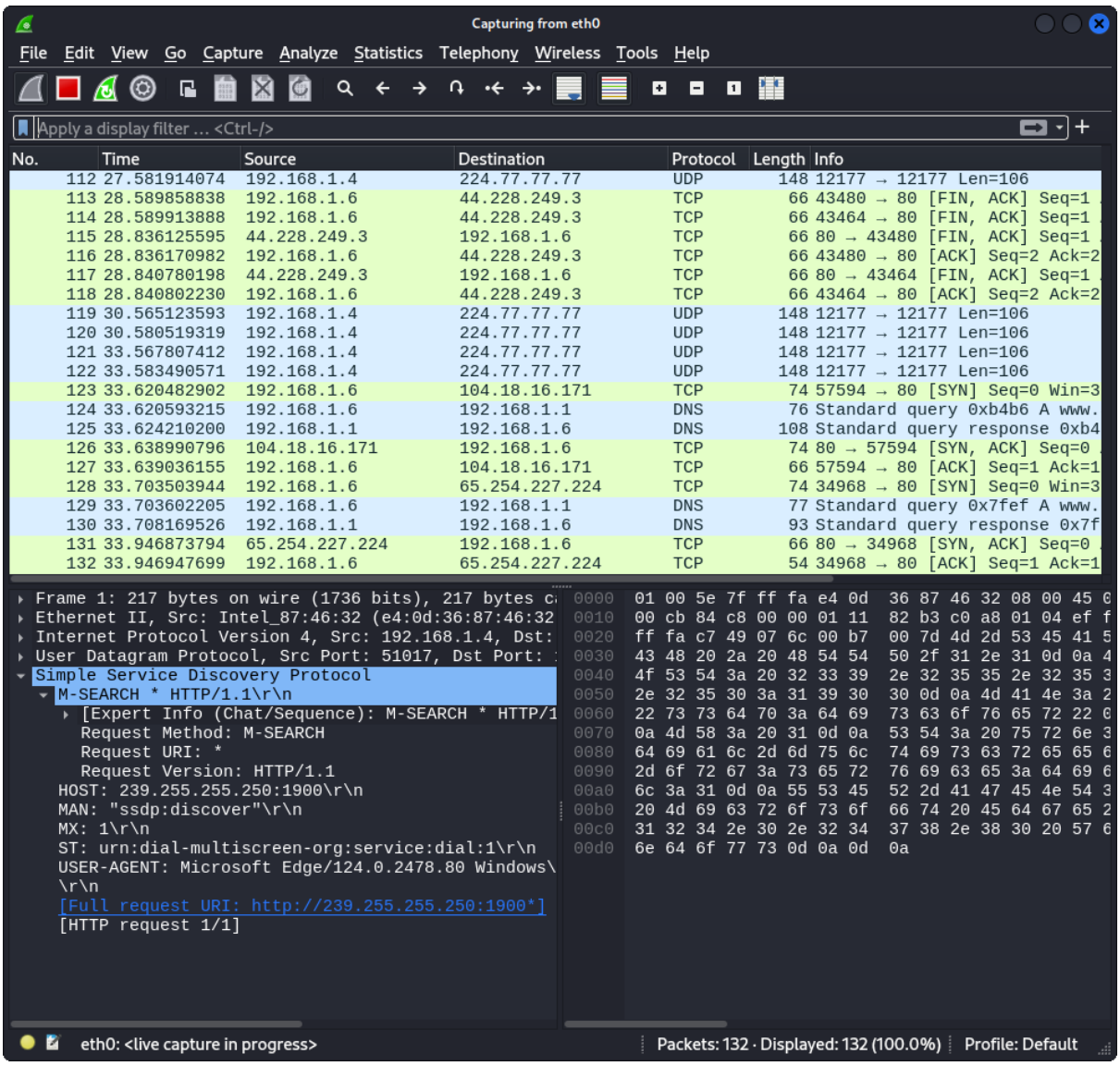
2. **Log in to the Website:** Navigate to the login page of the website http://testphp.vulnweb.com/ in your web browser. Log in using a username and password.

   Username given: SANDEEP.

   Password given: SHADOWFOX@SANDEEP370

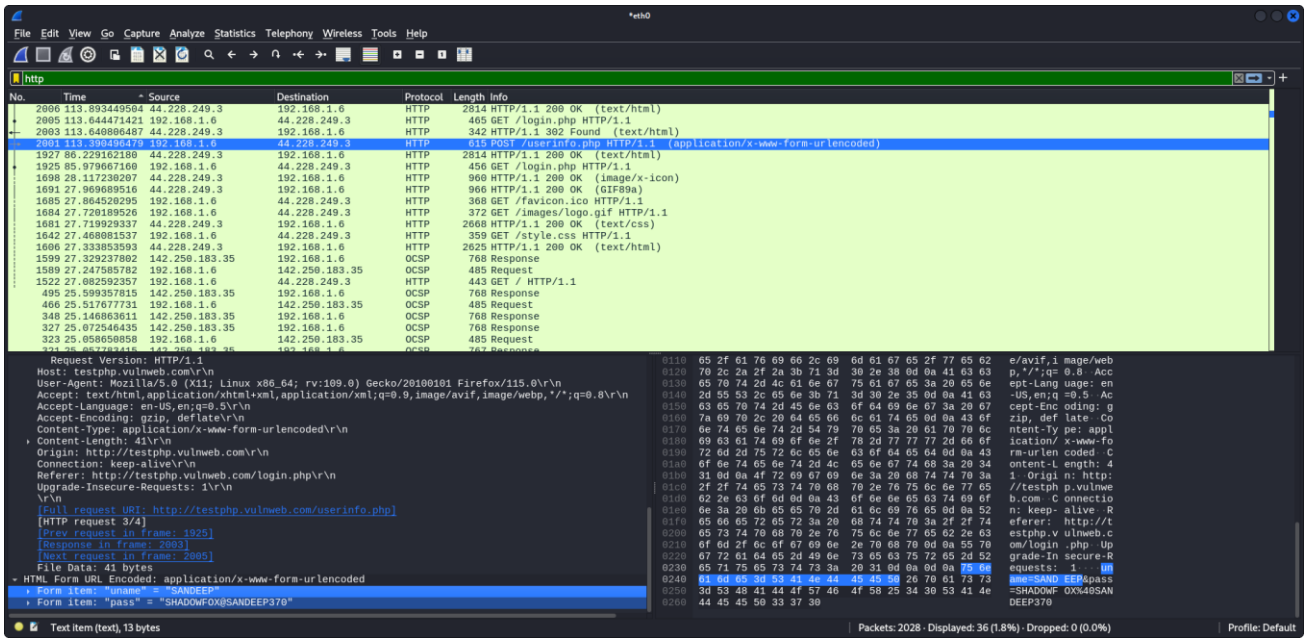
3. **End the Capture**: After completing the login process, stop capturing network traffic in Wireshark.

4. **Apply Filters:** In Wireshark, use a filter to focus on HTTP and HTTPS traffic. These protocols are commonly used to transmit login credentials.



5. **Inspect Captured Packets:** Examine the captured packets to identify any HTTP POST requests containing form data. These requests may hold the login credentials submitted during the login process.

6. **Extract and Analyse Credentials**: Carefully review the content of HTTP POST requests to locate and extract any usernames and passwords transmitted over the network.



Result:

The login credentials were.

username: SANDEEP and password: SHADOWFOX@SANDEEP370

The packets and data sent over HTTP protocol were in plaintext and hence we could capture/sniff them and gain the victims username and password by capturing over that protocol.

Conclusion:

Capturing and analysing network traffic can be an effective method for identifying vulnerabilities in the transmission of login credentials. By addressing these vulnerabilities through encryption, secure protocols, and other best practices, you can significantly strengthen the security of your network. These measures protect user privacy and help maintain the integrity of your system and websites.

Mitigations:

1. **Use HTTPS:** Ensure that all communication, including login forms, is encrypted using HTTPS. This encrypts data during transmission, protecting it from interception.
2. **Avoid HTTP:** Disable HTTP access on your website to prevent unencrypted data transmission. Redirect all HTTP traffic to HTTPS.
3. **Strong Password Policies:** Implement strong password policies to increase security. Require users to use complex passwords and encourage the use of multi-factor authentication.
4. **Secure Cookies:** Use secure cookie attributes (such as HttpOnly and Secure) to protect session cookies from being accessed or tampered with by malicious actors.
5. **Limit Packet Sniffing:** Use network security measures such as firewalls and intrusion detection/prevention systems to monitor and limit packet sniffing attempts on your network.
6. **Secure Network**: Implement network-level security measures, such as VPNs, to provide secure, encrypted tunnels for data transmission.