CYBERSECURITY INTERN REPORT AT SHADOWFOX

Name [redacted]

Batch: [redacted]

Gmail:- ris [redacted]

Task Level:- Beginner & Intermediate Level

Task Level (Beginner)

Table of Content

List Of Figures

Introduction and information about the report and the machine

[Task Level (Beginner)]

❖ Introduction:-

In the domain of cybersecurity, it is vital to comprehend and detect vulnerabilities in web applications to protect against potential cyber threats. During my internship, I was assigned the responsibility of conducting several security evaluations on the website->

http://testphp.vulnweb.com/

❖ Information about the report:-

1. Port Scanning:-

   The initial task was to identify all open ports on the target website. By performing a port scan, we sought to reveal any possible entry points that malicious actors might exploit to gain unauthorized access to the web server.

2. Brute Forcing directories on website:-
   The second task involved executing a brute force attack to list all directories on the website. This method aimed to uncover hidden or unsecured directories that might contain sensitive information or could be used for further exploitation.

3. Network Traffic Interception:-

   Finally, we executed a network traffic interception by logging into the website and capturing network packets using Wireshark. This enabled us to examine the data being transmitted between the client and server, with the goal of identifying any credentials or sensitive information being sent in plaintext.

❖ Required machine:-

- Standard computer system with network connectivity
- Linux operating system or Windows operating system.

Task-1

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

❖ **Attack name:-** Port Scanning

❖ **Severity:-** High and its score 7.0 – 8.9

Reason:- HTTP port [redacted] is considered highly severe because it is unencrypted, making it extremely susceptible to eavesdropping and data interception by malicious actors. This vulnerability can expose sensitive information, such as login credentials and credit card details, when used for web browsing.

❖ **Steps to reproduce with screen shots:-**

➢ Step 1: Target IP identification-

The website's IP address was identified using the ping command, [redacted] a precursor to the port scanning pr[redacted]Nmap, a powerful network scanning to[redacted]as performed on the detected IP address. This scan aimed to uncover open ports and the services running on them.

➢ Step 2: Port scanning using nmap-

Using Nmap, a powerful network scanning tool, a port scan was conducted on the identified IP address. The goal was to uncover open ports and their associated services. Here are the results of the port scan:

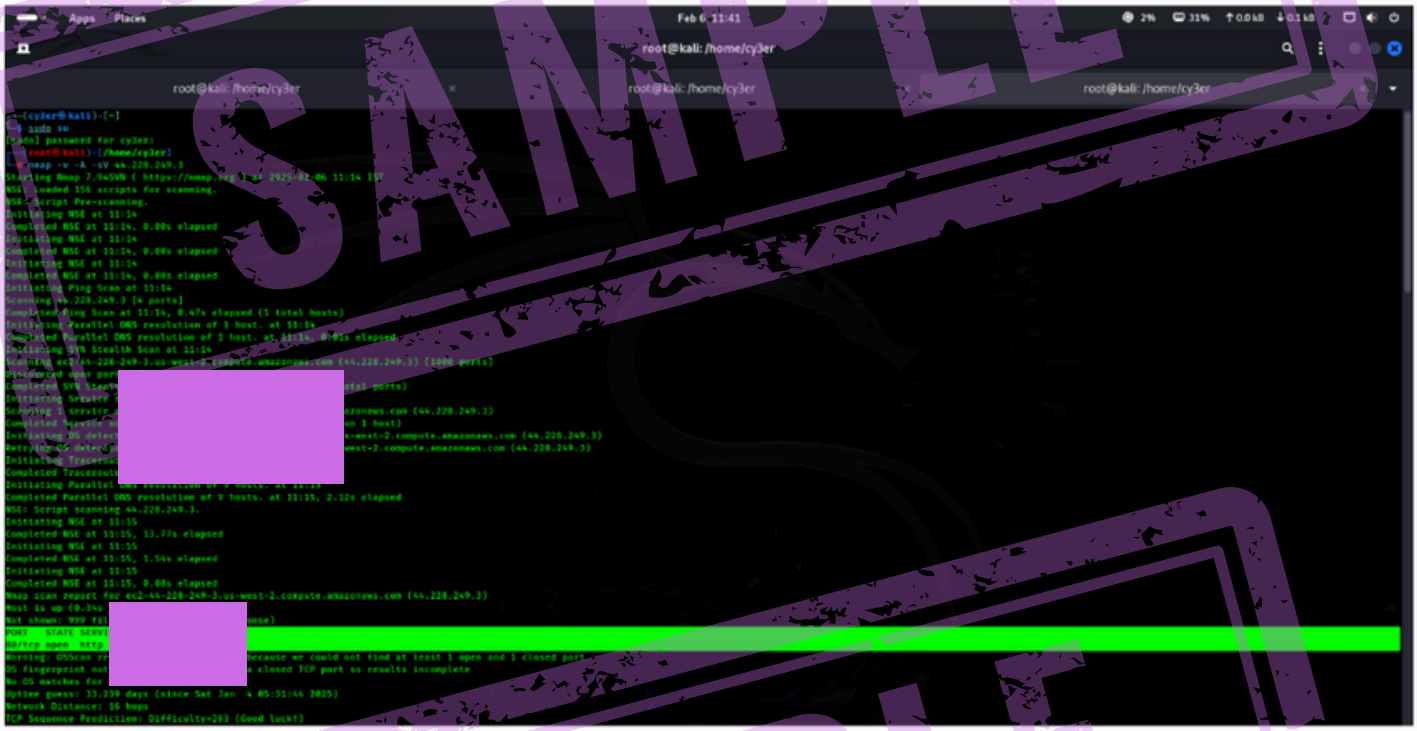
- Target website:- http://testphp.vulnweb.com/
- Target Ip address:- 44[redacted]
- Nmap command:- nm[redacted].3

Analysis:-

Port/Protocol	State	Service	Version
[redacted]/tcp	open	Http	nginx 1.19.0

Port [redacted] hosts an active HTTP service, which usually signifies a functioning web server. This server is powered by Nginx version 1.19.0. To reduce the risk of vulnerabilities, regularly applying the most recent security patches to web servers is crucial.

➢ Step 3: Nmap scanning fig(1):-



Fig(1): Nmap scanning

❖ **Impact:-**

➢ Advantages:-

1. Utilizing port [redacted] for HTTP traffic guarantees your website remains readily accessible, even for users who do not specify a port number in their browser.
2. Older browsers and some automated systems might rely on port 80 for accessing web resources.

➢ Disadvantages:-

1. Since HTTP traffic is unencrypted, it can be intercepted by malicious actors, leading to potential man-in-the-middle attacks.

This can compromise sensitive data like login credentials and personal information.

2. Open ports can be exploited by attackers to gain unauthorized access to your server.

❖ Mitigation steps:-

1. Make sure you're using the most recent version of Nginx. Regularly updating the software ensures you receive security patches that fix known vulnerabilities.
2. Redirecting HTTP traffic to HTTPS (po[redacted]3) ensures data encryption, providing a layer of protection against eavesdropping and man-in-the-middle attacks.This switch to HTTPS ensures that data sent between the server and client is securely encrypted, making it much harder for attackers to intercept or tamper with the information.
3. Implement lo[redacted]ring systems to quickly identify and respond to a[redacted]vities.These systems will help you track unusua[redacted]e swift action to maintain security.
4. Limit access to sensitive directories and files by configuring allow and deny directives in your Nginx settings. This step enhances security by ensuring only authorized users can access specific parts of your server.
5. Conduct regular security audits and vulnerability scans to identify and address potential issues.

Resources Used:-

Kali Linux operating system, network mapping(Nmap) and Ping Tool.

Task-2

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

- **Attack name:-** Brute Forcing directories on website
- **Severity:-** High and its score 7.0 – 8.9

  Reason:- The severity is high because directories like admin, secured, vendor, and CVS often contain critical information and vulnerabilities that can be exploited to compromise the entire system.
- **Steps to reproduce with screen shots:-**
  - Step 1: Target url:-
    Firstly, get the target webpage url., in my assessment the target url is http://testphp.vulnweb.com
  - Step 2: brute forcing url:-
    Using dirbuster, the kali Linux commandline tool. Scan that url using tool and you'll find all the hidden directories in that website. But here is the one catch,If you want to strong result then you need a strong directory wordlist for scanning.
    Here are the results of the scan:
    - Target website:- http://testphp.vulnweb.com
    - Co[redacted]scanning:-
      dir[redacted]hp.vulnweb.com

    Analysis:-

    ==> DIREC[redacted]dmin/

    + http://testp[redacted]

    + http://testp[redacted]

    + http://testp[redacted]

    ==> DIREC[redacted]VS/

+ http://[redacted]/Entries

+ http://[redacted]/Repository

+ http://[redacted]/Root

+ http://[redacted]con.ico

==> DI[redacted]ulnweb.com/images/

+ http://[redacted]x.php

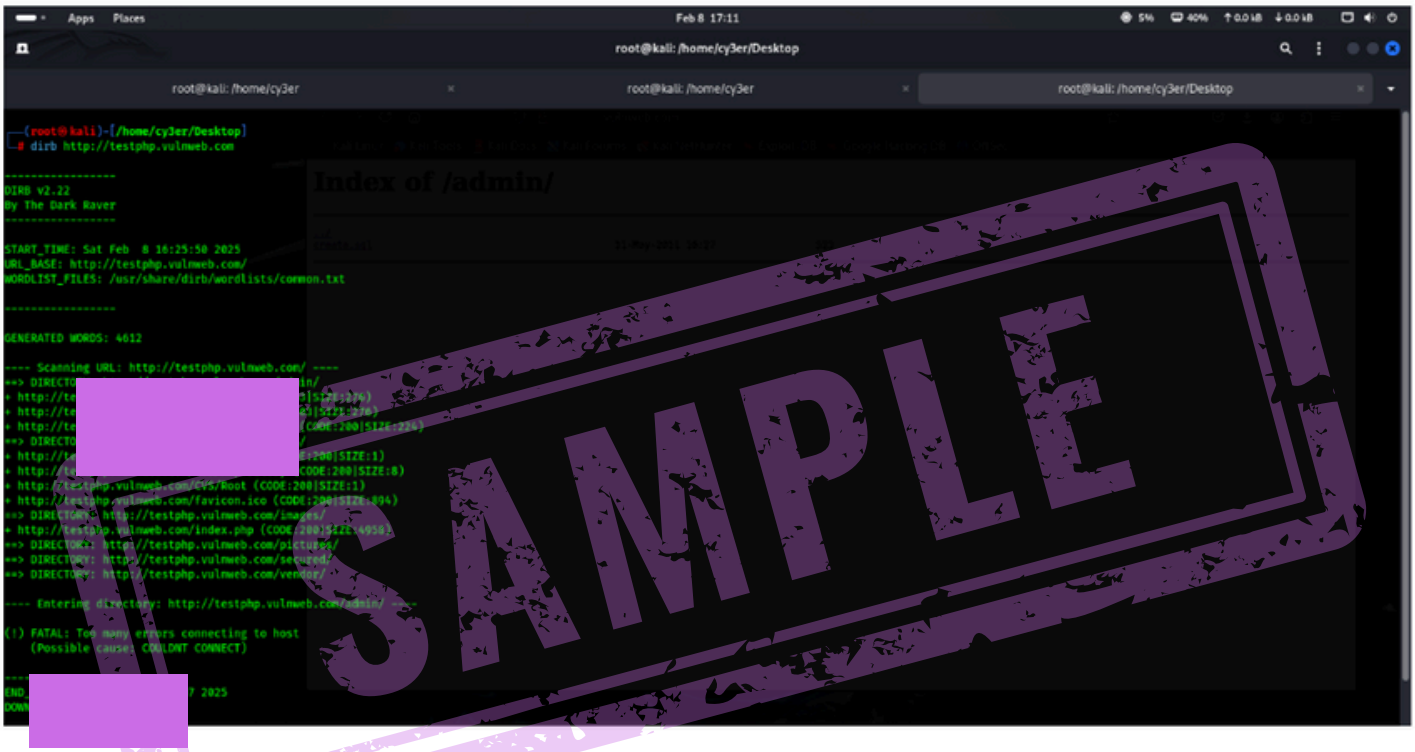
==> DI[redacted]ulnweb.com/pictures/

==> DI[redacted]ulnweb.com/secured/

==> DI[redacted]ulnweb.com/vendor/

Above that all directories and sub-directories we got in scanning. The most sever thing is in here, exposer of the admin, cvs, secured and vendor directories or database.

➢ Step 3: dirbuster scanning fig(2):-



Fig(2): dirbuster scanning

❖ **Impact:-**

1. Directories such as these could contain configuration files, login credentials, or proprietary data. Accessing them might result in data breaches or unauthorized access.
2. These directories may contain files that can be targeted by attackers, enabling them to take advantage of these flaws to gain further access or execute harmful code.
3. Identifying and exploiting these directories can interrupt the website's regular functioning, leading to downtime or performance degradation.

❖ **Mitigation steps:-**

1. Limit access to important directories by implementing strong authentication and authorization measures. Make certain that only approved users are permitted to access these directories.
2. Regularly review and update the configuration files to ensure they [redacted] Remove default or unne[redacted] exploited.
3. Encr[redacted]ese directories to protect it from unauthorized access. Use strong encryption algorithms and secure key management practices.
4. Implement comprehensive monitoring and logging to detect and respond to unauthorized access attempts. Analyze logs regularly to identify suspicious activities.
5. Set appropriate file permissions to limit access to critical files. Ensure that sensitive files are not accessible to the public or unauthorized users.
6. Maintain regular backups of your data and have a well-defined recovery plan in place in case of a security incident.

Resources Used:-

Kali Linux operating system, dirbuster tool

Task-3

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

- **Attack name:-** Network sniffing
- **Severity:-** High and its score 7.0 – 8.9

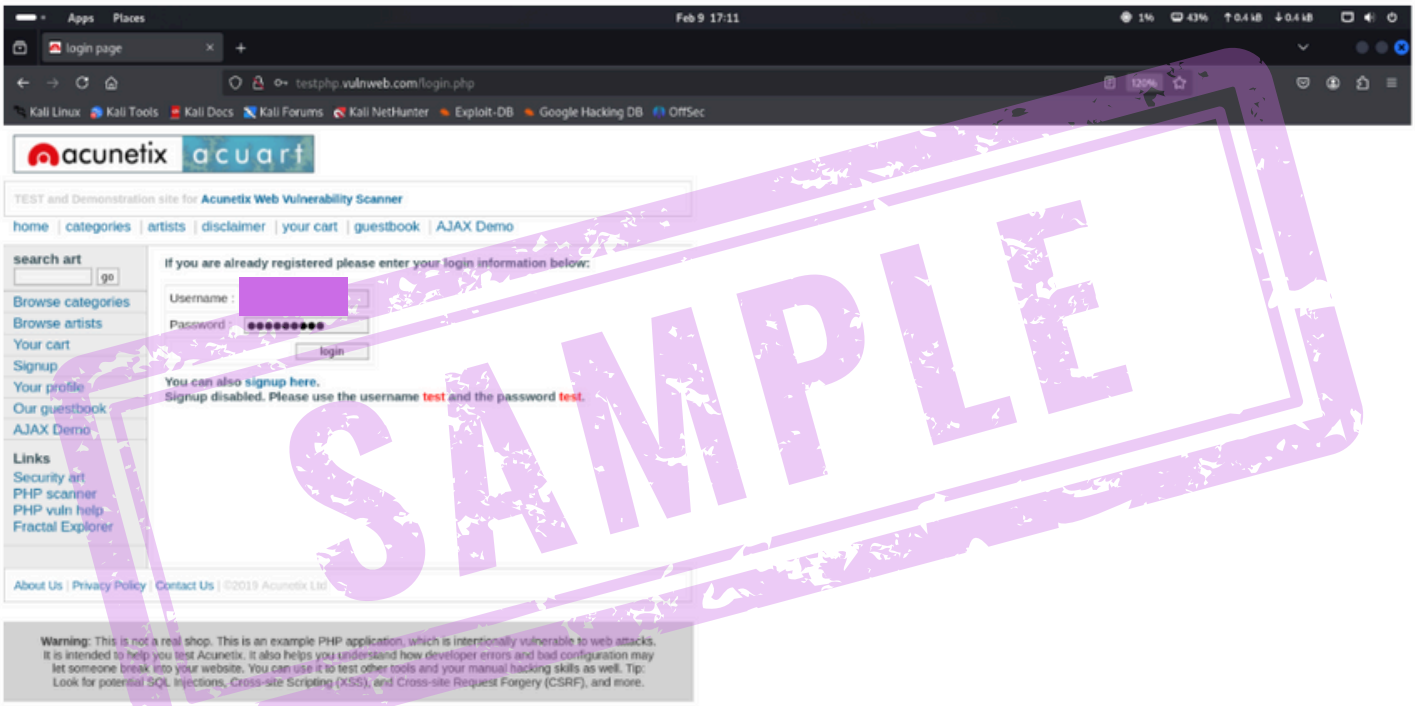
  Reason:- Network sniffing is highly severe because it can reveal sensitive information such as passwords and personal data, breach privacy, exploit network weaknesses, disrupt regular operations, and result in legal and compliance complications.
- **Steps to reproduce with screen shots:-**
  - Step 1: Launching Wireshark:-

    Launch the Wire[redacted]n your Linux environment. Once Wireshark [redacted]e network capture process to monitor and record network traffic. Make sure you have the necessary permissions and select the appropriate network interface to capture the data. This will allow you to analyze the packets traveling through your network and identify any potential issues or vulnerabilities also.
  - Step 2: Attempting to login:-

    Once we initiate packet capturing with Wireshark, we will navigate to the website http://testphp.vulnweb.com/login.php After reaching the login page, we will proceed to enter the credentials and attempt to log in. This process will allow us to capture the packets that are transmitted during the login attempt. By doing so, we can analyze the captured data to identify any potential vulnerabilities or security weaknesses in the authentication process.
    - Here I am giving:-

      Username: [redacted]

      password:-



Fig(3): log in attempt

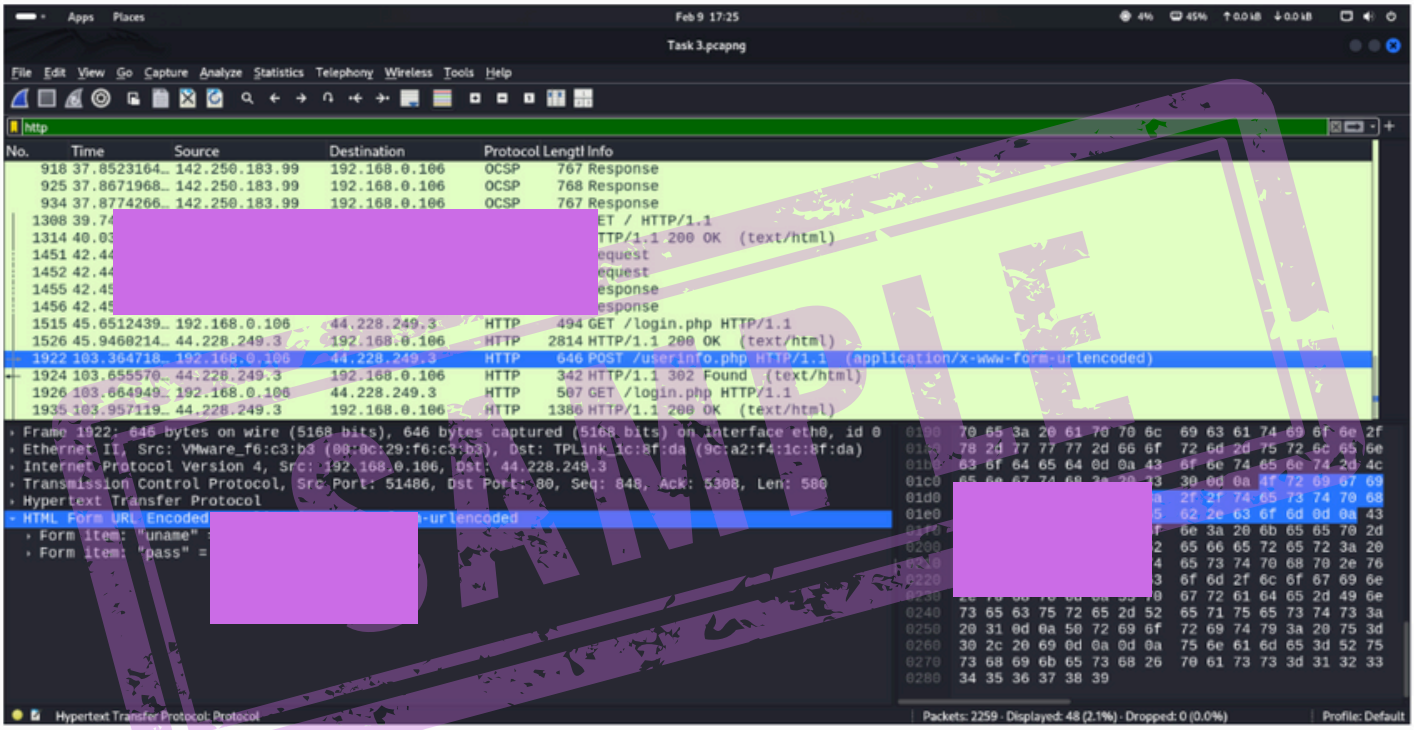
- Step 3: Stop!!:-

  After we have captured the necessary packets, we will stop the [redacted] process in Wireshark. This will allow us [redacted] analyze the recorded data. By halting the capt[redacted]us on examining the specific packets [redacted] particularly those related to the login attempt.

  Analysis:-

Wireshark has captured numerous packets, but we are specifically interested in the HTTP packets. To filter these out, we can use the display filter option within Wireshark. By entering "http" in the filter section, we can isolate and view all the captured HTTP packets. This allows us to focus our analysis on the relevant traffic and ignore the rest.
after that, I found the actual packet of that login credential attempt, You can see in fig (4).

Here are the results:



Fig(4): wireshark result

❖ Impact:-

1. If login credentials are exposed in network traffic, unauthorized individuals could easily intercept and use them to gain access to user accounts.
2. If the credentials be linked to accounts with financial information, attackers could misuse them to commit financial fraud, which could involve unauthorized transactions or accessing sensitive financial data.
3. Attackers can use the captured credentials to log in to user accounts, potentially leading to unauthorized access, data theft, and misuse of personal information.
4. Exposing login credentials puts user privacy at risk by giving attackers access to personal information, messages, and sensitive data connected to the accounts.

❖ **Mitigation steps:-**

1. Make certain that all communication between the client
2. encryption ensures that credentials are not sent in plain text, thus protecting them from exposure.
3. Adding an extra layer of security with MFA can help protect accounts even if credentials are compromised.
4. Consistently keep an eye on network traffic to identify any unusual activities or unauthorized access attempts. Use intrusion detection systems (IDS) to help spot potential threats.
5. Keep all systems, software, and applications up to date with the latest security patches to prevent vulnerabilities that could be exploited.
6. Limit a[redacted] directories and data exclusively to author[redacted]ementing strong authentication and au[redacted]ols.

Resources Used:-

Kali Linux operating system, Wireshark tool and Firefox browser.

***