Name: Aditya Baheti

Roll Number: 22BCE10521

Course / Department: BTech Computer science

Project Title: Web vulnerability scan

Executive Summary

This project focused on identifying and analyzing vulnerabilities present in web applications through the process of web vulnerability scanning. Using ethical hacking tools like Kali Linux and scanning environments such as Nikto and Drib, the project aimed to simulate real-world cyberattacks in a controlled setup. Various vulnerabilities, including SQL injection, and insecure login forms, were detected and documented. The final outcome of the project included detailed findings, along with actionable security recommendations for preventing these vulnerabilities.

Project Overview

- **Problem Statement:**

Web applications are increasingly targeted by attackers due to their exposure over the internet. Unpatched vulnerabilities in these applications can lead to data theft, service disruption, and unauthorized access. This project aims to simulate attacks and scan for such weaknesses to understand how they can be identified and mitigated.

- **Objectives:**
    - It set up a secure lab environment with vulnerable targets for scanning.
    - It perform vulnerability scans using professional-grade tools.
    - It analyze scan results, document findings, and provide recommendations.

- **Scope of Work:**

    This project was strictly confined to a controlled lab environment designed for educational and research purposes. The core focus was on identifying and understanding common web application vulnerabilities using scanning tools. Specifically, the work included:

    - Setting up a virtual lab consisting of a secure attacker machine (Kali Linux) and vulnerable target machines (Nikto) within an isolated network.

    - Installing, configuring, and using security tools such as , Nikto, and drib to perform active and passive vulnerability scans.

    - Simulating realistic web interactions (e.g., form submissions, login attempts) to observe how web applications respond to inputs and identify potential flaws.

    - Capturing and analyzing network traffic to detect insecure transmissions and potential information leaks.

    - Documenting identified vulnerabilities along with evidence (e.g., screenshots, logs, scan reports) and researching best practices for mitigation.

    Excluded from the project:

    - Scanning or testing any live or production web applications.

- Exploiting discovered vulnerabilities beyond what is necessary to confirm their existence.
- Performing denial-of-service (DoS) attacks or other disruptive activities.

Tools & Lab Setup

- **Primary Tools Used:** Kali Linux, Nikto, gobuster
- **Environment Details:**
  - Virtual Machine Setup: (e.g., VirtualBox, VMware)
  - Target VM: (e.g. Acunetix)
  - Network Mode: (Host-only / NAT / Bridged)

Tool Configuration & Commands :

- **`Command used`** **`nikto -h http://<target-ip>`**
- **`Gobuster  for packet capture and inspection`**

Implementation & Execution Summary

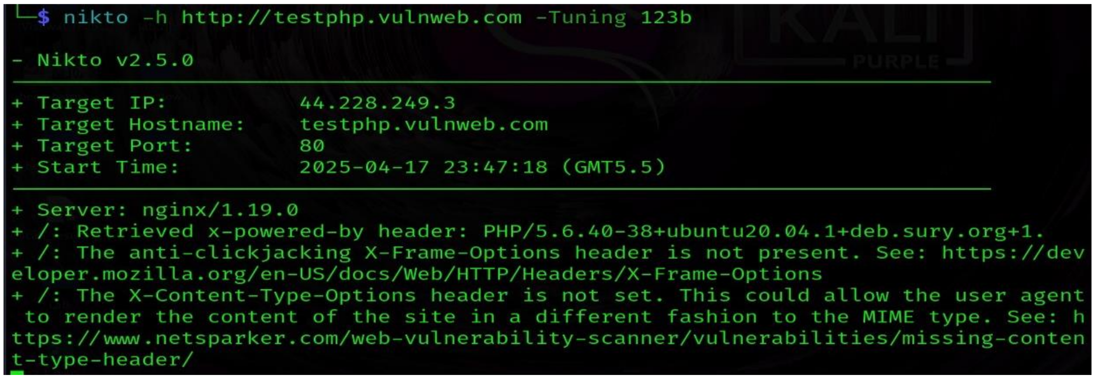
Firstly, Installing Kali Linux and target VMs on VirtualBox. Configured them on a Host-only network to isolate them from the internet.

Traffic Simulation / Attack Execution:
Accessed vulnerable applications through a browser and triggered various actions like login, form submissions, and search queries to simulate user behavior.

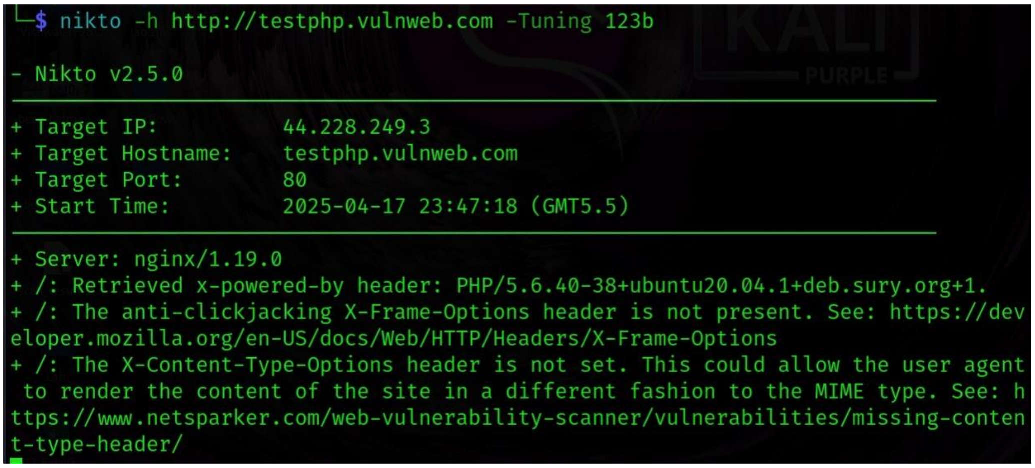
Packet Capture / Vulnerability Scanning:
Used Nikto to perform vulnerability scanning and Gobuster to capture packets during interaction.
Packet Analysis & Data Extraction:
Analyzed intercepted packets and scan reports to identify vulnerabilities such as plain text login credentials, outdated software, and input validation flaws.

Findings & Analysis

http://testphp.vulnweb.com/ is the website,
Using nikto the website is scanned, and result is below:

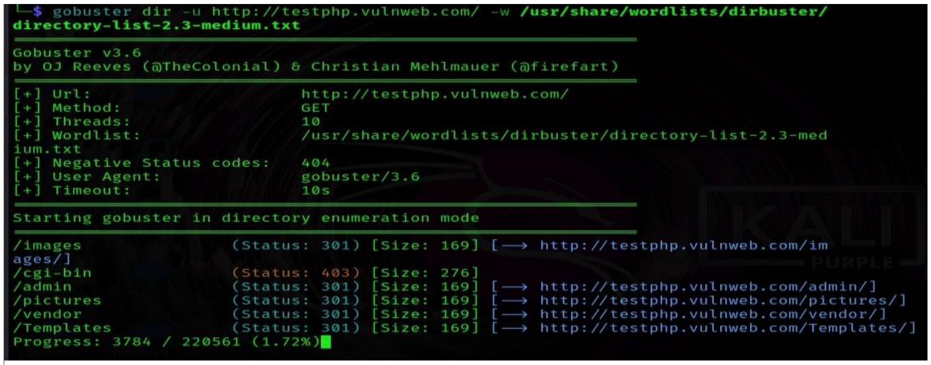


```
└─$ nikto -h http://testphp.vulnweb.com -Tuning 123b

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          44.228.249.3
+ Target Hostname:    testphp.vulnweb.com
+ Target Port:        80
+ Start Time:         2025-04-17 23:47:18 (GMT5.5)
---------------------------------------------------------------------------
+ Server: nginx/1.19.0
+ /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://dev
eloper.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent
 to render the content of the site in a different fashion to the MIME type. See: h
ttps://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-conten
t-type-header/
```

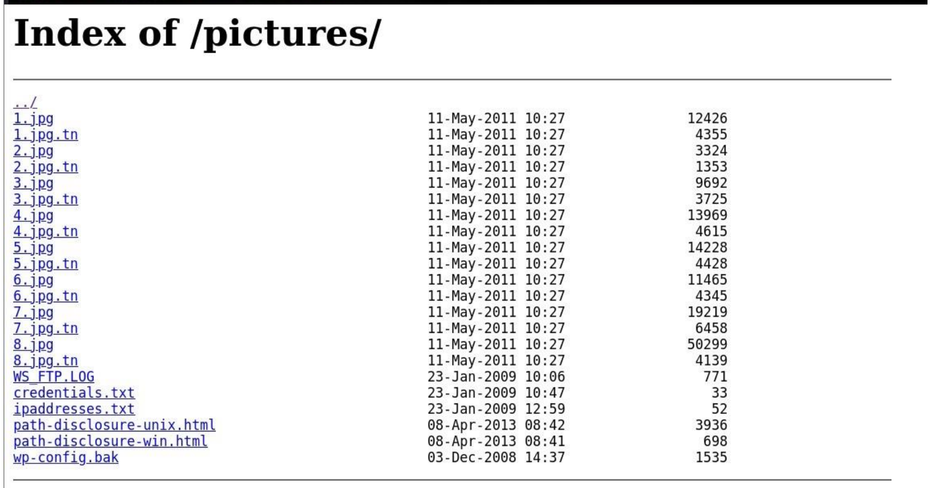


```
└─$ nikto -h http://testphp.vulnweb.com -Tuning 123b

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          44.228.249.3
+ Target Hostname:    testphp.vulnweb.com
+ Target Port:        80
+ Start Time:         2025-04-17 23:47:18 (GMT5.5)
---------------------------------------------------------------------------
+ Server: nginx/1.19.0
+ /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://dev
eloper.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent
 to render the content of the site in a different fashion to the MIME type. See: h
ttps://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-conten
t-type-header/
```

Using gobuster, we got to know about the hidden directories on the website.

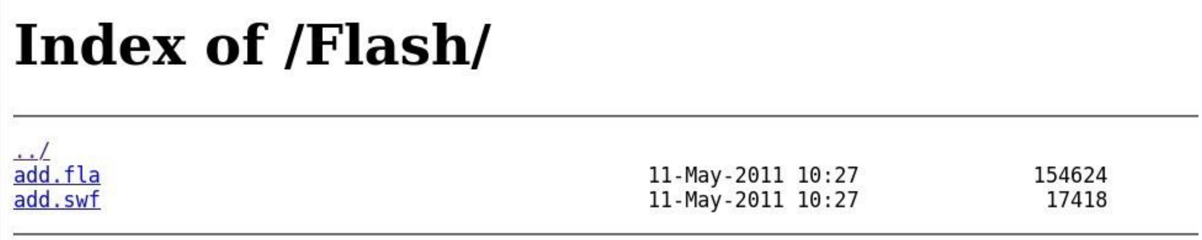


```
└─$ gobuster dir -u http://testphp.vulnweb.com/ -w /usr/share/wordlists/dirbuster/
directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://testphp.vulnweb.com/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-med
ium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/im
ages/]
/cgi-bin              (Status: 403) [Size: 276]
/admin                (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/admin/]
/pictures             (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/pictures/]
/vendor               (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/vendor/]
/Templates            (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/Templates/]
Progress: 3784 / 220561 (1.72%)
```



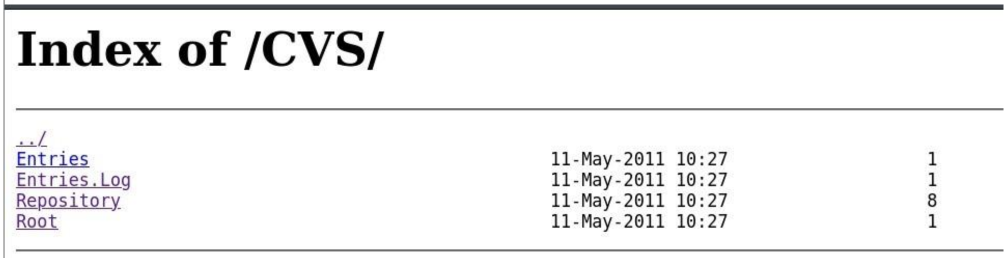
Index of /pictures/

```
../
1.jpg                                              11-May-2011 10:27               12426
1.jpg.tn                                           11-May-2011 10:27                4355
2.jpg                                              11-May-2011 10:27                3324
2.jpg.tn                                           11-May-2011 10:27                1353
3.jpg                                              11-May-2011 10:27                9692
3.jpg.tn                                           11-May-2011 10:27                3725
4.jpg                                              11-May-2011 10:27               13969
4.jpg.tn                                           11-May-2011 10:27                4615
5.jpg                                              11-May-2011 10:27               14228
5.jpg.tn                                           11-May-2011 10:27                4428
6.jpg                                              11-May-2011 10:27               11465
6.jpg.tn                                           11-May-2011 10:27                4345
7.jpg                                              11-May-2011 10:27               19219
7.jpg.tn                                           11-May-2011 10:27                6458
8.jpg                                              11-May-2011 10:27               50299
8.jpg.tn                                           11-May-2011 10:27                4139
WS_FTP.LOG                                         23-Jan-2009 10:06                 771
credentials.txt                                    23-Jan-2009 10:47                  33
ipaddresses.txt                                    23-Jan-2009 12:59                  52
path-disclosure-unix.html                          08-Apr-2013 08:42                3936
path-disclosure-win.html                           08-Apr-2013 08:41                 698
wp-config.bak                                      03-Dec-2008 14:37                1535
```



Index of /Flash/

```
../
add.fla                                            11-May-2011 10:27              154624
add.swf                                            11-May-2011 10:27               17418
```



Index of /CVS/

```
../
Entries                                            11-May-2011 10:27                   1
Entries.Log                                        11-May-2011 10:27                   1
Repository                                         11-May-2011 10:27                   8
Root                                               11-May-2011 10:27                   1
```

- Detected **SQL injection** vulnerability in login fields of **Acunetix .**
- Found **cross-site scripting (XSS)** flaws in input fields.
- Login credentials sent over **HTTP** (unencrypted), easily captured in nikto.
- Exposed server information and software versions identified via Nikto.

Security Recommendations

- Use encrypted protocols like **HTTPS** to protect data in transit.
- **Disable unused ports/services** to reduce the attack surface.

- Enforce **strong password policies** and account lockout mechanisms.
- Conduct **regular vulnerability scans** and apply patches promptly.
- Use **input validation** and sanitize user inputs to prevent injection attacks.

Learning Outcomes

Technically, I learned how to configure and use ethical hacking tools to identify real vulnerabilities in a web application environment. I also gained hands-on experience in packet analysis and vulnerability research. Professionally, I learned the importance of structured documentation and how to communicate technical findings clearly.

Future Scope

This project could be extended by integrating automated patch management tools and exploring more advanced vulnerability scanning techniques using AI-based scanners. Another enhancement could be the simulation of social engineering attacks or real-time alert systems for intrusion detection.

Conclusion

This project gave me hands-on experience in performing basic vulnerability assessments of a web application. I learned how simple tools can reveal critical weaknesses even in internal apps. It emphasized the importance of proactive scanning and secure configuration. These skills are directly applicable to real-world cybersecurity practices, ethical hacking.