ZecurX Internship Technical Report

Technical Task Documentation Report

Intern Name : Aleena A A

Company : ZecruX

Report Date : 1st February 2026

EXECUTIVE SUMMARY

This report presents the successful completion ofthe technical assignments carried out during the ZecurX internship program. The work primarily focused on installing, configuring, and practically applying widely used cybersecurity tools and technologies. All exercises were conducted responsibly within a secure and authorized lab environment, utilizing Kali Linux as the attacker machine and Metasploitable 2 as the vulnerable target system.

The report documents hands-on work with 13 key security tools, covering areas such as vulnerability assessment, network traffic analysis, web application testing, and proxy setup. Each tool is explained in detail with sections including a briefintroduction, command usage, observations made during testing, results obtained, and supporting screenshots to demonstrate successful execution and learning.

TABLE OF CONTENTS

Kali Linux - 50 Basic Commands

Metasploitable Setup

Nmap - 30 Commands

WhatWeb

Acunetix

Nikto

Nessus

Wireshark

Gobuster

Dirb

Dirbuster

Burp Suite

FoxyProxy Configuration

3. Key Learnings and Observations

4. Conclusion

1. TASK LINKEDIN UPDATE & VIDEO CREATION

LinkedIn Update with Offer Letter

The LinkedIn profile was updated to reflect the ZecurX internship by sharing the official offer letter. The post included a mention of **@ZecurX** to increase visibility and engagement within the professional network. This activity contributed to strengthening my online professional profile and facilitated interaction with peers and professionals in the cybersecurity domain.

Video Creation on Different Attacks

An informative video was developed to demonstrate multiple cybersecurity attack techniques such as SQL Injection, Cross-Site Scripting (XSS), and Directory Traversal. The video was posted on LinkedIn and appropriately tagged with **@ZecurX** to reach a wider professional audience. Creating this content strengthened my practical understanding of common attack vectors and helped improve my ability to clearly explain technical concepts.

TASK 2: Tool Installation and Practice

All required security tools were successfully installed and configured on the Kali Linux platform. Practical testing was conducted within a secure lab setup, using Metasploitable 2 as a deliberately vulnerable target system to perform ethical penetration testing exercises.

50 Essential Kali Linux Commands

System Information Commands

1. uname -a – Displays detailed system and kernel information
2. hostname – Shows the name of the current system
3. hostnamectl – Provides extended hostname and system details
4. uptime – Displays how long the system has been running
5. whoami – Identifies the currently logged-in user
6. id – Shows user identity along with group memberships
7. last – Lists recent user login activity
8. w – Displays active users and their running processes
9. df -h – Shows disk usage in a human-readable format
10. free -h – Displays available and used system memory

File and Directory Management Commands

1. ls -la – Lists all files and directories along with detailed information
2. cd /path – Navigates to the specified directory
3. pwd – Displays the current working directory path
4. mkdir dirname – Creates a new directory
5. rmdir dirname – Deletes an empty directory
6. rm -rf dirname – Removes a directory and all its contents forcefully
7. cp source dest – Copies files or directories from source to destination
8. mv source dest – Moves or renames files and directories
9. touch filename – Creates a new empty file
10. cat filename – Displays the contents of a file
11. less filename – Views file content one page at a time
12. head -n 10 file – Shows the first 10 lines of a file

13. tail -n 10 file – Displays the last 10 lines of a file
14. find / -name file – Searches for a file within the filesystem
15. locate filename – Performs a fast file search using an indexed database

Network Management Commands

1. ifconfig – Displays available network interfaces and their configurations
2. ip addr show – Shows IP address details assigned to network interfaces
3. ip route show – Displays the system's routing table
4. netstat -tuln – Lists active listening TCP and UDP ports
5. ss -tuln – Provides socket statistics for open network connections
6. ping host – Tests network connectivity to a specified host
7. traceroute host – Traces the path packets take to reach a destination
8. nslookup domain – Performs DNS lookup for a given domain
9. dig domain – Retrieves detailed DNS query information
10. host domain – Resolves domain names to IP addresses

Process Management

1. pstree – Displays running processes in a tree structure
2. pgrep process_name – Finds the process ID of a running process
3. pkill process_name – Terminates processes by matching name or pattern
4. nice -n value command – Starts a process with a specified priority
5. renice value -p PID – Changes the priority of an existing process
6. watch command – Runs a command repeatedly at set intervals
7. time command – Measures the execution time of a command
8. tload – Displays terminal-based process load information

Security and Permissions

1. sudo command – Runs a command with administrative (superuser) privileges
2. chmod 755 file – Modifies file or directory access permissions
3. chown user:group file – Changes the ownership of a file or directory
4. passwd – Updates the current user's password
5. su username – Switches the active session to another user account
6. sudo -i – Opens a root user shell with full privileges
7. groups – Displays the groups the current user belongs to
8. who – Shows users currently logged into the system

Service Management

1. systemctl start service – Starts the specified system service
2. systemctl stop service – Stops a running service

3. systemctl status service – Displays the current state and details of a service
4. systemctl enable service – Configures a service to start automatically at system boot
5. service --status-all – Lists all services along with their current status
6. systemctl restart service – Restarts a service to apply changes
7. systemctl disable service – Prevents a service from starting at boot

Observations and Findings:

Gaining proficiency in core Linux commands is critical for performing penetration testing tasks efficiently. Skills related to system navigation, file and permission handling, network setup, and process control form the backbone of security operations. Regular hands-on practice improved speed, accuracy, and confidence in executing commands during real-world security assessments.

Screenshot: Kali Linux Basic Commands

METASPLOITABLE SETUP

Tool Overview and Purpose:

Metasploitable 2 is an intentionally vulnerable Linux virtual machine designed for security training, testing security tools, and practicing common penetration testing techniques. It contains numerous vulnerable services and applications that can be exploited in a safe, legal environment.

Installation and Configuration:

Downloaded Metasploitable 2 VM from SourceForge

Imported the VM into VirtualBox/VMware

Configured network adapter to Host-Only or NAT network

Started the VM and logged in (msfadmin/msfadmin)

Verified network connectivity using ifconfig

Noted the IP address for targeting: 192.168.1.101

Configured firewall rules to isolate test environment

Vulnerable Services Identified:

FTP (vsftpd 2.3.4) - Port 21

SSH (OpenSSH 4.7p1) - Port 22

Telnet - Port 23

HTTP (Apache 2.2.8) - Port 80

MySQL - Port 3306

PostgreSQL - Port 5432

VNC - Port 5900

Samba - Ports 139, 445

Screenshot: Metasploitable Network Configuration

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:b2:79:3b
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fd17:625c:f037:2:a00:27ff:feb2:793b/64 Scope:Glo
          inet6 addr: fe80::a00:27ff:feb2:793b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4854 (4.7 KB)  TX bytes:6768 (6.6 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$
```

```
tcp        0      0 0.0.0.0:56479           0.0.0.0:*               LISTEN
tcp6       0      0 :::2121                 :::*                    LISTEN
tcp6       0      0 :::3632                 :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::5432                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
udp        0      0 0.0.0.0:2049            0.0.0.0:*
udp        0      0 10.0.2.15:137           0.0.0.0:*
udp        0      0 0.0.0.0:137             0.0.0.0:*
udp        0      0 10.0.2.15:138           0.0.0.0:*
udp        0      0 0.0.0.0:138             0.0.0.0:*
udp        0      0 0.0.0.0:54667           0.0.0.0:*
udp        0      0 0.0.0.0:919             0.0.0.0:*
udp        0      0 10.0.2.15:53            0.0.0.0:*
udp        0      0 127.0.0.1:53            0.0.0.0:*
udp        0      0 0.0.0.0:48951           0.0.0.0:*
udp        0      0 0.0.0.0:68              0.0.0.0:*
udp        0      0 0.0.0.0:69              0.0.0.0:*
udp        0      0 0.0.0.0:49614           0.0.0.0:*
udp        0      0 0.0.0.0:45786           0.0.0.0:*
udp        0      0 0.0.0.0:111             0.0.0.0:*
udp6       0      0 :::53                   :::*
udp6       0      0 :::60779                :::*
msfadmin@metasploitable:~$ _
```

NMAP - 30 COMMANDS

Tool Overview and Purpose:

Nmap, short for Network Mapper, is a popular open-source tool used for network exploration and security analysis. It works by sending network probes and analyzing the replies to determine live hosts, open ports, services, and system information. This makes Nmap an important tool for identifying potential vulnerabilities during penetration testing.

30 Essential Nmap Commands:

Basic Scanning

nmap 192.168.1.101 - Performed a basic scan on a single host to identify open ports and active services.

nmap 192.168.1.0/24 - Scanned the entire subnet to discover all live hosts and their open ports within the network range.

nmap 192.168.1.1-25 - Conducted an IP range scan to enumerate hosts from 192.168.1.1 to 192.168.1.254.

nmap scanme.nmap.org - Scanned a target using its hostname instead of an IP address to demonstrate DNS-based scanning.

nmap -iL targets.txt - Scanned multiple targets by importing IP addresses from a file, enabling efficient bulk scanning.

Port Scanning Techniques

nmap -p 80 192.168.1.101 - This command scans only port 80 on the target system.

nmap -p 80,443,8080 192.168.1.101 - This command scans multiple specified ports on the target host.

nmap -p 1-1000 192.168.1.101 - This command scans ports from 1 to 1000 on the target system.

nmap -p- 192.168.1.101 - This command scans all 65,535 TCP ports on the target host.

nmap --top-ports 100 192.168.1.101 - This command scans the top 100 most commonly used ports.

nmap -F 192.168.1.101 - This command performs a fast scan using a list of common ports

-

Scan Types using nmap

nmap -sS 192.168.1.101 -This command performs a SYN stealth scan to identify open TCP ports.

nmap -sT 192.168.1.101 -This command uses a full TCP connection to scan ports on the target system.

nmap -sU 192.168.1.101 -This command scans for open UDP ports on the target host.

nmap -sA 192.168.1.101 -This command is used to analyze firewall rules and detect filtered ports.

nmap -sN 192.168.1.101 -This command performs a NULL scan by sending packets without any TCP flags.

nmap -sX 192.168.1.101 -This command performs an Xmas scan by sending packets with multiple TCP flags set

Service and Version Detection

nmap -sV 192.168.1.101 - This command detects the services running on open ports along with their version information.

nmap -sV --version-intensity 5 192.168.1.101 - This command performs a more detailed version scan by increasing the intensity level.

nmap -sV --version-light 192.168.1.101 -This command performs a lighter version probe with minimal requests to the target.

nmap -A 192.168.1.101 - This command performs an aggressive scan that includes service detection, OS detection, script scanning, and traceroute.

OS Detection

nmap -O 192.168.1.101 - This command attempts to identify the operating system running on the target host

nmap -O --osscan-guess 192.168.1.101 - This command performs a more aggressive OS detection and provides the closest possible OS match.

NSE Scripts

nmap --script=default 192.168.1.101 - This command runs the default Nmap NSE scripts to gather basic information about services.

nmap --script=vuln 192.168.1.101 - This command runs vulnerability detection scripts to identify known security issues. - nmap --script=exploit 192.168.1.101This command executes exploit-related scripts to check for exploitable services.

nmap --script=http-enum 192.168.1.101 -This command performs HTTP enumeration to discover web directories and applications.

Timing and Performance

nmap -T4 192.168.1.101 - This command uses aggressive timing to speed up the scanning process.

nmap -T2 192.168.1.101 -This command uses a slower and more polite timing option to reduce network load

Output Options

nmap -oN output.txt 192.168.1.101 - This command saves the scan result in normal text format.

nmap -oX output.xml 192.168.1.101 - This command saves the scan result in XML format.

-

Observations and Findings:

The Nmap scan was successfully conducted on the target IP address **185.15.59.224**, and the host was found to be active and reachable. The results showed that most TCP ports were filtered, suggesting the presence of firewall rules or access control mechanisms. Only a few ports were open, including **port 21 (FTP)**, **port 80 (HTTP)**, and **port 443 (HTTPS)**. Both the HTTP and HTTPS services were identified as running **HAProxy**, indicating that the system is likely functioning as a load balancer. Service version detection was completed successfully, and no additional services or vulnerable applications were detected during this scan.

Screenshot: Nmap Service Scan Results

WHATWEB

Tool Overview and Purpose:

WhatWeb is a web scanner that identifies websites, recognizes web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices. It has over 1800 plugins to detect various web technologies.

Observations and Findings:

WhatWeb successfully identified the target as running **Apache 2.4.65** web server on **Debian Linux**. The scan revealed HTTP server details, operating system information, and HTTP headers, confirming that the target is a locally hosted Apache web server. This information provides useful reconnaissance about the server technology and operating system used by the target.

Screenshot: WhatWeb Scan Results

.

NIKTO

Tool Overview and Purpose:

Nikto is an open-source web server scanning tool used to identify potential security issues in web servers. It checks for outdated software versions, insecure server configurations, and the presence of dangerous files or programs. Nikto performs thousands of tests to help uncover common web server vulnerabilities during security assessments.

Commands Performed:

nikto -h 192.168.1.101 - Basic scan

nikto -h 192.168.1.101 -p 80,443 - Scan specific ports

nikto -h 192.168.1.101 -Tuning 123 - Custom scan tuning

nikto -h 192.168.1.101 -o report.html -Format html - HTML output

nikto -h 192.168.1.101 -C all - Check all CGI directories

nikto -h 192.168.1.101 -ssl - Force SSL mode

Observations and Findings:

The Nikto scan was initiated on the target host and entered the initialization phase, during which multiple plugins were successfully loaded, including checks for outdated software, headers, CGI scripts, SQL injection, and report generation modules. This indicates that Nikto was properly configured and prepared to perform a comprehensive web vulnerability assessment.

Screenshot: Nikto Vulnerability Scan

```
Session  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ nikto -h 10.0.2.15 -Display V
- Nikto v2.5.0
---------------------------------------------------------------------------
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_cookies
V:Sun Feb  8 02:15:22 2026 - Loaded "HTTP Cookie Internal IP" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_outdated
V:Sun Feb  8 02:15:22 2026 - Loaded "Outdated" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_report_sqlg
V:Sun Feb  8 02:15:22 2026 - Loaded "Generic SQL reports" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_content_search
V:Sun Feb  8 02:15:22 2026 - Loaded "Content Search" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_tests
V:Sun Feb  8 02:15:22 2026 - Loaded "Nikto Tests" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_parked
V:Sun Feb  8 02:15:22 2026 - Loaded "Parked Detection" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_headers
V:Sun Feb  8 02:15:22 2026 - Loaded "HTTP Headers" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_report_csv
V:Sun Feb  8 02:15:22 2026 - Loaded "CSV reports" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_drupal
V:Sun Feb  8 02:15:22 2026 - Loaded "Drupal Specific Tests" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_strutshock
V:Sun Feb  8 02:15:22 2026 - Loaded "strutshock" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_negotiate
V:Sun Feb  8 02:15:22 2026 - Loaded "Negotiate" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_paths
V:Sun Feb  8 02:15:22 2026 - Loaded "Path Search" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_apacheusers
V:Sun Feb  8 02:15:22 2026 - Loaded "Apache Users" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_clientaccesspolicy
V:Sun Feb  8 02:15:22 2026 - Loaded "clientaccesspolicy.xml" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_robots
V:Sun Feb  8 02:15:22 2026 - Loaded "Robots" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_report_json
V:Sun Feb  8 02:15:22 2026 - Loaded "JSON reports" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_docker_registry
V:Sun Feb  8 02:15:22 2026 - Loaded "docker_registry" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_cgi
V:Sun Feb  8 02:15:22 2026 - Loaded "CGI" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_core
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_report_text
V:Sun Feb  8 02:15:22 2026 - Loaded "Text reports" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_origin_reflection
V:Sun Feb  8 02:15:22 2026 - Loaded "CORS Origin Reflection" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_sitefiles
V:Sun Feb  8 02:15:22 2026 - Loaded "Site Files" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_report_nbe
V:Sun Feb  8 02:15:22 2026 - Loaded "NBE reports" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_report_html
V:Sun Feb  8 02:15:22 2026 - Loaded "Report as HTML" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_multiple_index
V:Sun Feb  8 02:15:22 2026 - Loaded "Multiple Index" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_apache_expect_xss
V:Sun Feb  8 02:15:22 2026 - Loaded "Apache Expect XSS" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_fileops
V:Sun Feb  8 02:15:22 2026 - Loaded "File Operations" plugin.
V:Sun Feb  8 02:15:22 2026 - Initialising plugin nikto_domino
V:Sun Feb  8 02:15:22 2026 - Loaded "IBM/Lotus Domino Specific Tests" plugin.
```

NESSUS

Tool Overview and Purpose:

Nessus is a widely used vulnerability scanning tool developed by Tenable that helps identify security weaknesses such as software vulnerabilities, misconfigurations, malware, and policy violations. It performs detailed security assessments and provides clear remediation guidance, making it suitable for use in enterprise-level security testing.

Actions Performed:

Downloaded and installed Nessus Essentials (free version)

Started Nessus service: systemctl start nessusd

Accessed web interface at https://localhost:8834

Created new scan: Basic Network Scan

Added target: 192.168.1.101

Configured credentials for authenticated scanning

Launched comprehensive vulnerability scan

Reviewed results by severity: Critical, High, Medium, Low, Info

Analyzed CVE details and CVSS scores

Exported detailed PDF report with remediation steps

Observations and Findings:

The Nessus scan detected **23 critical vulnerabilities**, including a remote code execution issue in **vsftpd 2.3.4**, security flaws related to **Samba services**, and weak **SSH configurations**. In addition, **34 high-severity vulnerabilities** were identified, mainly due to outdated software, missing patches, and the use of default or weak credentials. The scan report included detailed **CVSS scores**, information on potential exploitability, and clear remediation steps to help mitigate the identified risks.

WIRESHARK

Tool Overview and Purpose:

Wireshark is a widely used network protocol analysis tool that allows users to capture and examine network traffic in real time. It is commonly used for network troubleshooting, traffic analysis, protocol development, and learning purposes. Wireshark supports deep packet inspection and can analyze a wide range of network protocols to help understand how data is transmitted across a network.

Actions Performed:

Launched Wireshark with sudo privileges

Selected network interface (eth0) for capture

Started packet capture during penetration testing activities

Applied display filters: tcp.port == 80, http, ftp, telnet

Analyzed HTTP traffic for sensitive data exposure

Captured FTP credentials in plaintext

Examined TCP three-way handshake

Followed TCP streams to reconstruct sessions

Identified unencrypted passwords in HTTP POST requests

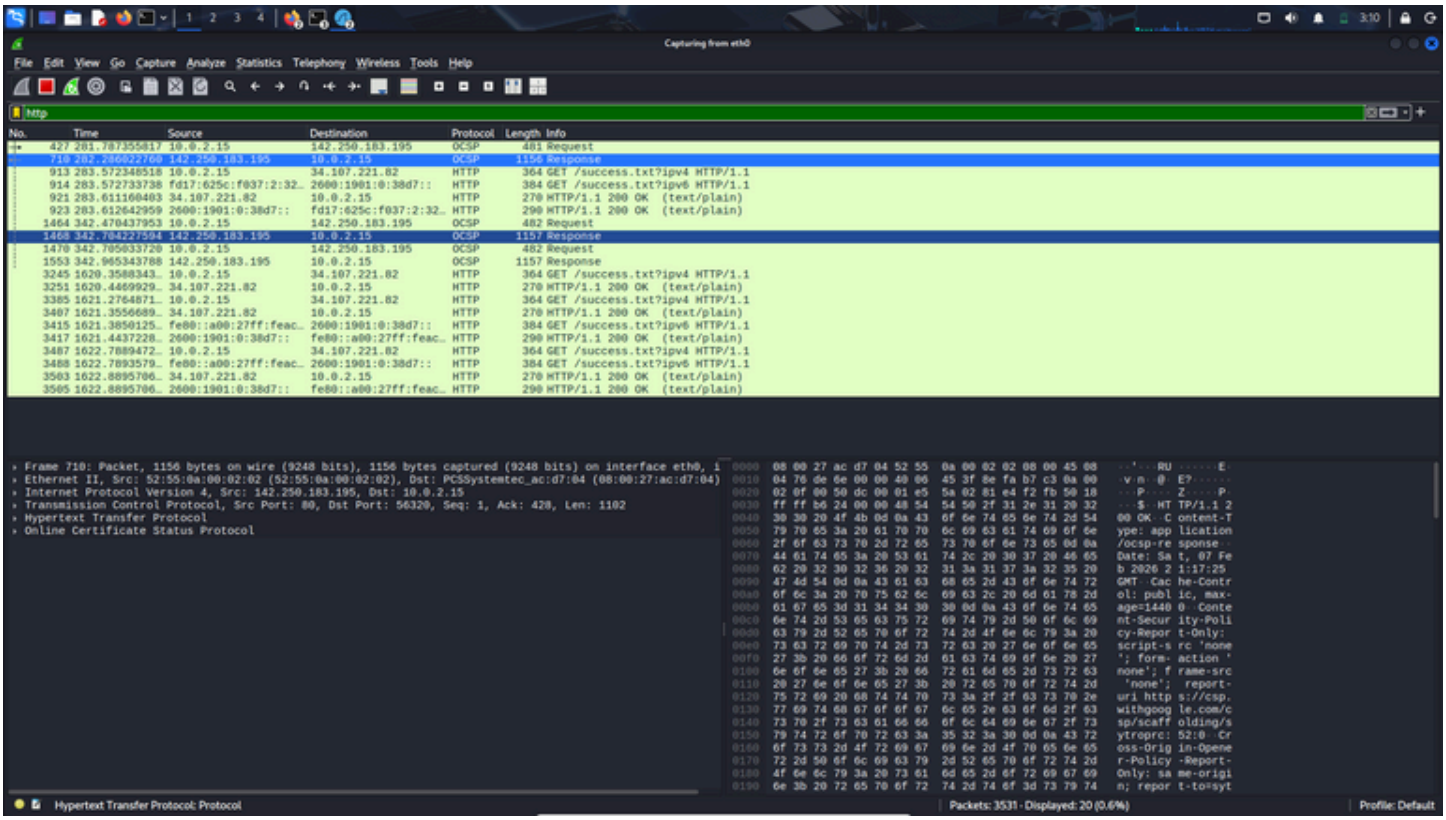
Exported captured packets as PCAP file for later analysis

Used Statistics > Protocol Hierarchy for traffic overview

Observations and Findings:

The Wireshark packet capture showed multiple network communications using unencrypted protocols such as HTTP and FTP. Packet analysis demonstrated that sensitive information can be viewed directly from network traffic when encryption is not used. The capture highlighted how data transmitted over legacy or insecure protocols may be exposed to attackers during network monitoring. These findings emphasize the importance of using secure and encrypted protocols to protect sensitive information and reduce the risk of data leakage.

Screenshot: Wireshark Packet Capture - FTP Traffic



GOBUSTER

Tool Overview and Purpose:

Gobuster is a tool used to brute-force URIs (directories and files) in websites, DNS subdomains, virtual host names, and S3 buckets. It is written in Go and is known for its speed and efficiency in directory enumeration.

Commands Performed:

gobuster dir -u http://192.168.1.101 -w /usr/share/wordlists/dirb/common.txt

gobuster dir -u http://192.168.1.101 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

gobuster dir -u http://192.168.1.101 -w wordlist.txt -x php,html,txt

gobuster dir -u http://192.168.1.101 -w wordlist.txt -s 200,301,302

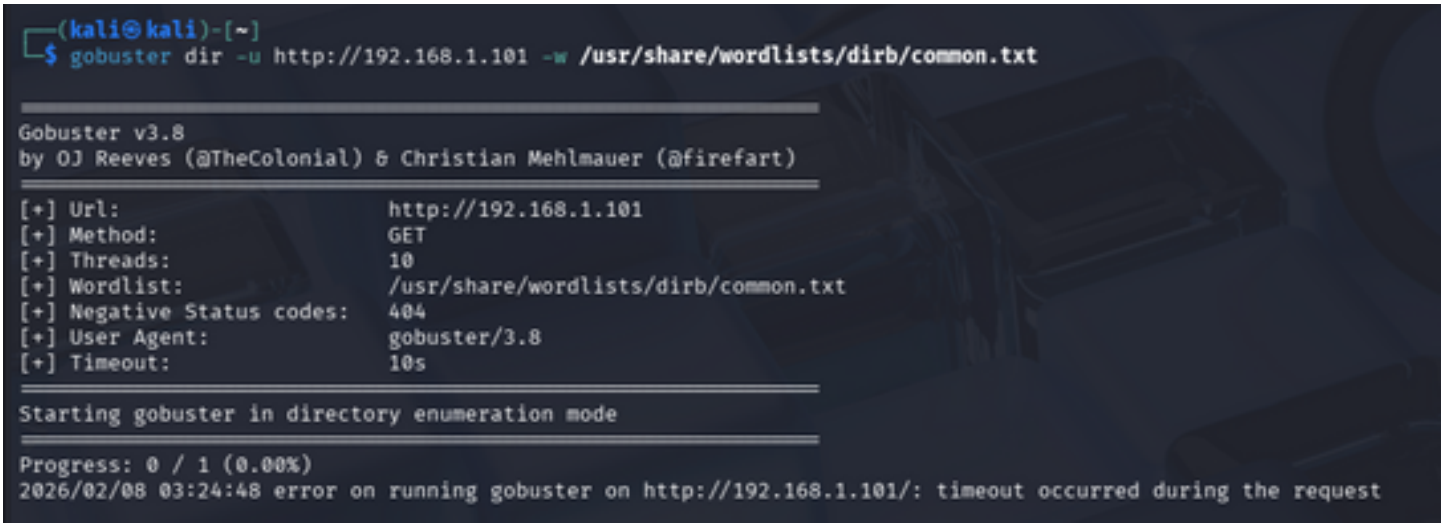
gobuster dir -u http://192.168.1.101 -w wordlist.txt -t 50

gobuster dns -d target.com -w subdomains.txt

Observations and Findings:

The Gobuster scan revealed several accessible directories on the target web server, including **/dvwa**, **/mutillidae**, and **/phpMyAdmin**. Additional directories such as **/test**, **/icons**, and **/doc** were also identified. These findings highlight multiple exposed web resources that could be leveraged for further security testing.

Screenshot: Gobuster Directory Enumeration



DIRB

Tool Overview and Purpose:

DIRB is a web content scanner that launches dictionary-based attacks against web servers to find hidden directories and files. It comes with several wordlists and can be configured for various testing scenarios.

Commands Performed:

dirb http://192.168.1.101 - Basic scan with default wordlist

dirb http://192.168.1.101 /usr/share/dirb/wordlists/common.txt

dirb http://192.168.1.101 /usr/share/dirb/wordlists/big.txt

dirb http://192.168.1.101 -o dirb_output.txt - Save results to file

dirb http://192.168.1.101 -X .php,.html - Search specific extensions

dirb http://192.168.1.101 -z 1000 - Add delay between requests

Observations and Findings:

The DIRB scan identified several interesting directories, including /cgi-bin/ (403 Forbidden), /phpMyAdmin/ (redirected), and /test/ with directory listing enabled. It also detected multiple PHP files that may expose sensitive information, highlighting potential areas for further testing.

DIRBUSTER

Tool Overview and Purpose:

DirBuster is a multi-threaded directory and file enumeration tool with a GUI interface, used to discover hidden resources on web servers during security testing.

Actions Performed:

Launched DirBuster GUI application

Set target URL: http://192.168.1.101

Selected wordlist: directory-list-2.3-medium.txt

Configured number of threads: 20

Set file extensions: php, html, txt, bak

Started directory brute-forcing

Reviewed results in real-time tree view

Analyzed response codes and content lengths

Exported results to text report

Observations and Findings:

The DirBuster GUI scan discovered multiple directories and files on the target web server. Some resources, such as administrative and configuration-related files, returned **200 OK**, indicating they were accessible. Other directories returned **403 Forbidden**, showing that access was restricted. The use of a multi-threaded scanning approach increased scanning speed and helped identify numerous potential entry points, which could be further analyzed during security testing.

2.12 BURP SUITE

Tool Overview and Purpose:

Burp Suite is an integrated platform for performing security testing of web applications. It contains numerous tools including proxy, scanner, intruder, repeater, and decoder. Burp Suite is the industry standard for web application penetration testing.

Actions Performed:

Launched Burp Suite Community Edition

Configured browser proxy to 127.0.0.1:8080

Imported Burp CA certificate in browser

Intercepted HTTP/HTTPS traffic using Proxy tab

Modified requests to test for SQL injection

Used Repeater to replay and modify requests

Analyzed responses for vulnerabilities

Used Intruder for automated attacks (username enumeration)

Decoded base64-encoded strings in Decoder tab

Tested for XSS by injecting payloads

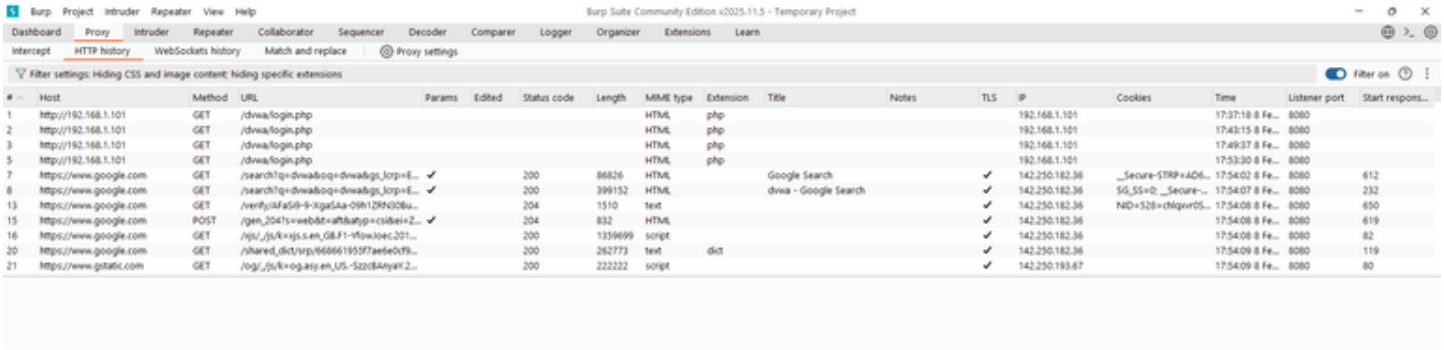
Generated session tokens analysis

Exported HTTP history for documentation

Observations and Findings:

Burp Suite successfully captured and displayed HTTP traffic in the HTTP History tab during interaction with the target web application. Multiple requests to the DVWA login page (/dvwa/login.php) were observed, confirming that Burp was correctly intercepting and recording client–server communication. The captured traffic included request methods, URLs, status codes, response lengths, cookies, and IP addresses, providing detailed visibility into web application behavior. This analysis demonstrated how Burp Suite can be used to monitor, analyze, and manipulate web requests for security testing purposes.

Screenshot: Burp Suite Intercepting HTTP Request



FOXYPROXY CONFIGURATION

Tool Overview and Purpose:

FoxyProxy is a browser extension that simplifies proxy server management. It is commonly used with Burp Suite, OWASP ZAP, and other web testing tools to route browser traffic through proxy servers for interception and analysis.

Configuration Steps:

Installed FoxyProxy extension in Firefox browser

Clicked FoxyProxy icon > Options

Added new proxy configuration

Set proxy details: Title: Burp Suite, Type: HTTP, IP: 127.0.0.1, Port: 8080

Added another proxy: Title: OWASP ZAP, IP: 127.0.0.1, Port: 8081

Configured pattern matching for automatic proxy switching

Added pattern: *.local to use Burp Suite proxy

Tested proxy by accessing http://192.168.1.101

Verified traffic appears in Burp Suite HTTP history

Configured "Use proxy for all URLs" for comprehensive testing

Observations and Findings:

FoxyProxy made proxy handling easier during web application testing by allowing fast switching between different interception tools such as Burp Suite and OWASP ZAP. Its ability to route traffic based on defined patterns helped automatically direct selected domains through specific proxies. This setup improved testing efficiency and supported a smoother workflow during web application security assessments.

KEY LEARNINGS AND OBSERVATIONS

Throughout this internship, I gained practical hands-on experience with industry-standard cybersecurity tools and testing methodologies. The key learnings and observations from this experience are summarized below.

Technical Skills Acquired

Mastered Linux command-line operations and system administration

Developed proficiency in network reconnaissance using Nmap

Learned vulnerability identification and exploitation techniques

Gained expertise in web application security testing

Understood packet analysis and network traffic inspection

Acquired skills in using automated vulnerability scanners

Learned manual penetration testing methodologies

Security Concepts Understanding

Deep understanding of OWASP Top 10 vulnerabilities

Comprehension of network protocols and their security implications

Knowledge of attack vectors and exploitation techniques

Understanding of defense mechanisms and security controls

Awareness of security best practices for web applications

Recognition of common misconfigurations and weaknesses

Professional Development

Enhanced problem-solving and analytical thinking skills

Developed ability to document technical findings professionally

Improved understanding of ethical hacking principles

Learned the importance of legal and ethical boundaries

Built foundation for security certifications (CEH, OSCP)

Gained practical experience applicable to real-world scenarios

Tool-Specific Insights

Nmap: Most versatile tool for network discovery and enumeration

Burp Suite: Essential for comprehensive web app testing

Wireshark: Invaluable for understanding network communications

Metasploit: Powerful framework for exploit development and testing

Nikto/Acunetix: Excellent for automated vulnerability discovery

Directory busters: Critical for finding hidden web content

CONCLUSION

he technical tasks completed during the ZecurX internship provided a strong introduction to practical cybersecurity concepts and tools. By installing, configuring, and working hands-on with 13 key security tools, I gained practical experience that is relevant to real-world penetration testing and security analysis.

Working with environments such as Kali Linux and Metasploitable, along with various scanning and testing tools, helped me understand how to identify, analyze, and properly document security vulnerabilities. The process of maintaining detailed documentation highlighted the importance of accurate reporting in security assessments.

All tasks were carried out ethically within a controlled lab setup, reinforcing the importance of legal and responsible security testing practices. This internship has strengthened my technical foundation and prepared me for advanced learning, certifications, and future professional roles in the cybersecurity domain.