Red Team Playbook

Table of Contents

1. Revision History

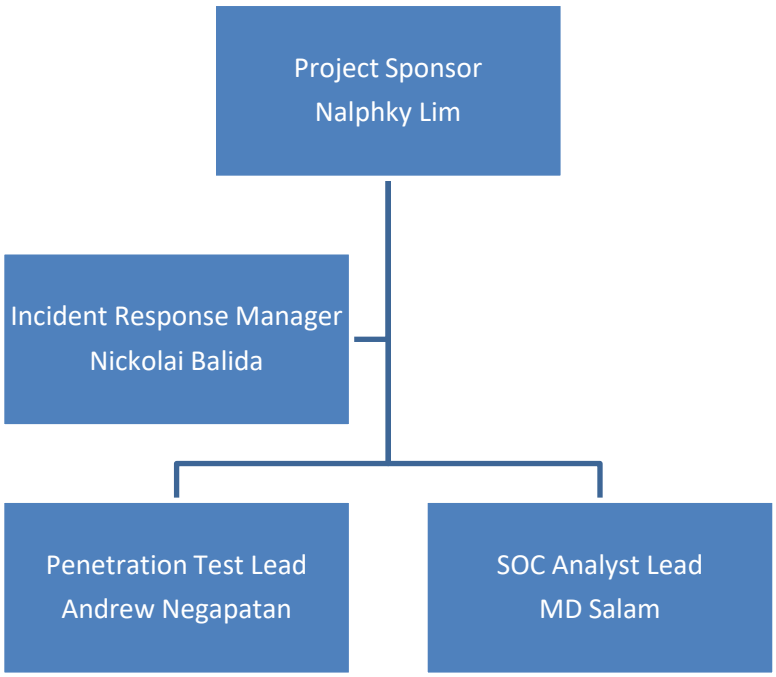
This Security Incident Response Plan has been modified as follows:

Date	Version	Modification	Modifier
2025-11-10	*1.0*	*Plan created*	*Md Salam, Andrew Negapatan, Nickolai Balida*

Review Cycle

This Security Incident Response Plan must be reviewed at least annually.

2. Team structure



3. Overview

3.1 Purpose

The Red Team Playbook serves as a comprehensive guide for conducting simulated attacks and evaluating an organisation's security posture. It provides a structured framework detailing the strategies, techniques, and procedures (TTPs) employed by the Red Team to assess defensive capabilities and identify potential vulnerabilities.

The playbook outlines a systematic approach to testing security measures through realistic attack simulations that replicate real-world scenarios. Its purpose is to evaluate the effectiveness of security controls, incident response mechanisms, and overall resilience to potential threats. By adhering to the Red Team Playbook, organisations can emulate the behaviours and methodologies of actual adversaries to uncover weaknesses, strengthen defences, and enhance their overall security posture.

The primary purpose of the Red Team Playbook is to reinforce an organisation's cybersecurity maturity and resilience in the following ways:

1) Identifying Vulnerabilities
Through realistic simulated attacks, the Red Team systematically uncovers weaknesses within the organisation's infrastructure, applications, and processes. These findings are essential for prioritising remediation efforts, strengthening defences, and reducing the organisation's overall risk exposure.

2) Testing Defence Capabilities
The playbook evaluates the organisation's ability to detect, respond to, and contain security incidents effectively. It assesses the performance of existing security controls, monitoring tools, and response procedures, while also measuring staff awareness and the effectiveness of training programs.

3) Providing Realistic Scenarios
By replicating real-world threats using advanced tactics, techniques, and procedures (TTPs) observed in actual adversaries, the playbook ensures simulations reflect genuine attack behaviour. This process helps identify potential blind spots in security strategies and offers actionable insights into the organisation's preparedness against evolving threats.

4) Enhancing Incident Response

The playbook outlines structured guidelines for managing security incidents detected during simulations. It supports continuous improvement of incident response plans, communication channels, and coordination between teams, ultimately enhancing the organisation's ability to respond swiftly and effectively to real-world cyber incidents.

5) Promoting Continuous Improvement
By conducting regular Red Team exercises and using the playbook as an ongoing reference, the organisation can continuously evaluate and enhance its security posture. Insights gained from each engagement serve as valuable feedback to refine security measures, update policies and procedures, and strengthen preparedness against emerging and evolving threats.

In summary, the Red Team Playbook is a critical resource that empowers organisations to proactively assess their security defences, uncover vulnerabilities, and build resilience against cyber threats. By implementing the playbook's guidelines, organisations can strengthen their defensive capabilities, reduce risks, and maintain a proactive stance against potential adversaries.

3.2 Scope

- Inclusions (within scope)

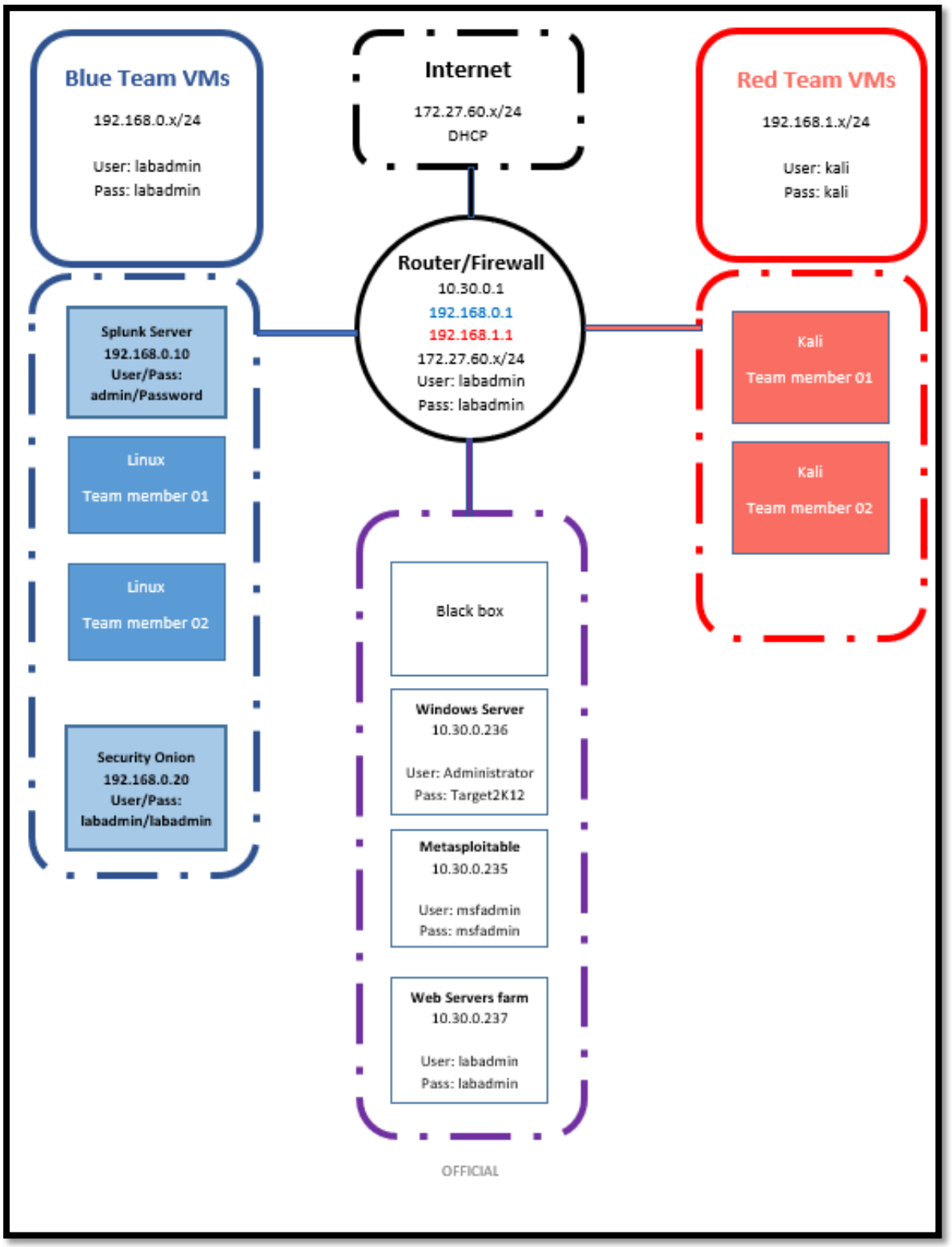
Assessment	Details
Internal Penetration Test	10.30.0.250 - Blackbox Host 10.30.0.235 - Metasploitable 10.30.0.236 - Windows Server 10.30.0.237 - Web Servers Farm

- Exclusions (outside of scope)]

  The team does not perform any of the following attacks during testing:

  - Denial of Service (DoS)
  - Phishing/Social Engineering

3.3 Topology


Blue Team VMs
192.168.0.x/24
User: labadmin
Pass: labadmin
Internet
172.27.60.x/24
DHCP
Red Team VMs
192.168.1.x/24
User: kali
Pass: kali
Router/Firewall
10.30.0.1
192.168.0.1
192.168.1.1
172.27.60.x/24
User: labadmin
Pass: labadmin
Splunk Server
192.168.0.10
User/Pass:
admin/Password
Linux
Team member 01
Linux
Team member 02
Security Onion
192.168.0.20
User/Pass:
labadmin/labadmin
Kali
Team member 01
Kali
Team member 02
Black box
Windows Server
10.30.0.236
User: Administrator
Pass: Target2K12
Metasploitable
10.30.0.235
User: msfadmin
Pass: msfadmin
Web Servers farm
10.30.0.237
User: labadmin
Pass: labadmin
OFFICIAL

4. Attacks

4.1 Vulnerability Scanning

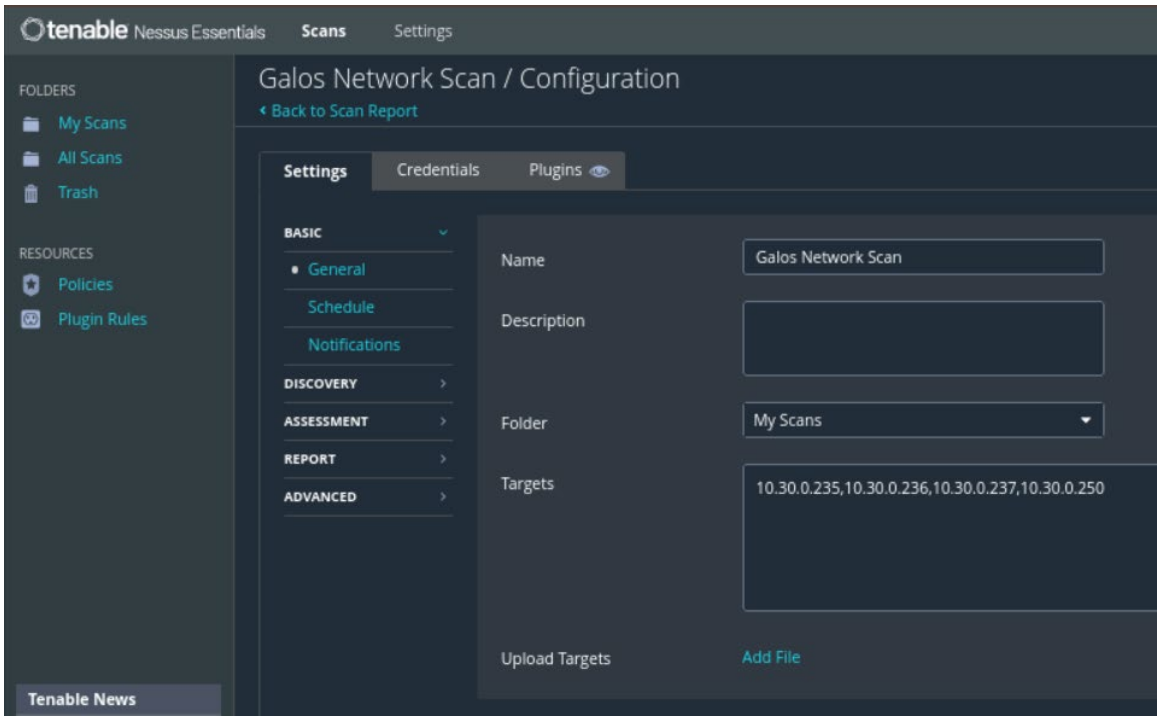
Target: 10.30.0.235 (Metaspoitable), 10.30.0.236 (Windows Server), 10.30.0.237 (Web Servers Farm), 10.30.0.250 (Black Box)

Objective: The purpose of this activity is to conduct a systematic vulnerability assessment of the target systems using Nessus. This process identifies known security weaknesses that could be exploited by attackers. Through vulnerability scanning, organisations can better understand their security posture, prioritise remediation actions, and strengthen overall system defences to reduce the likelihood of successful attacks.

Type of Attack: Vulnerability Scanning using Nessus

Attack steps and screenshots:

Open Nessus and Launch Advanced Scan against the targets.



4.2 Port Scanning

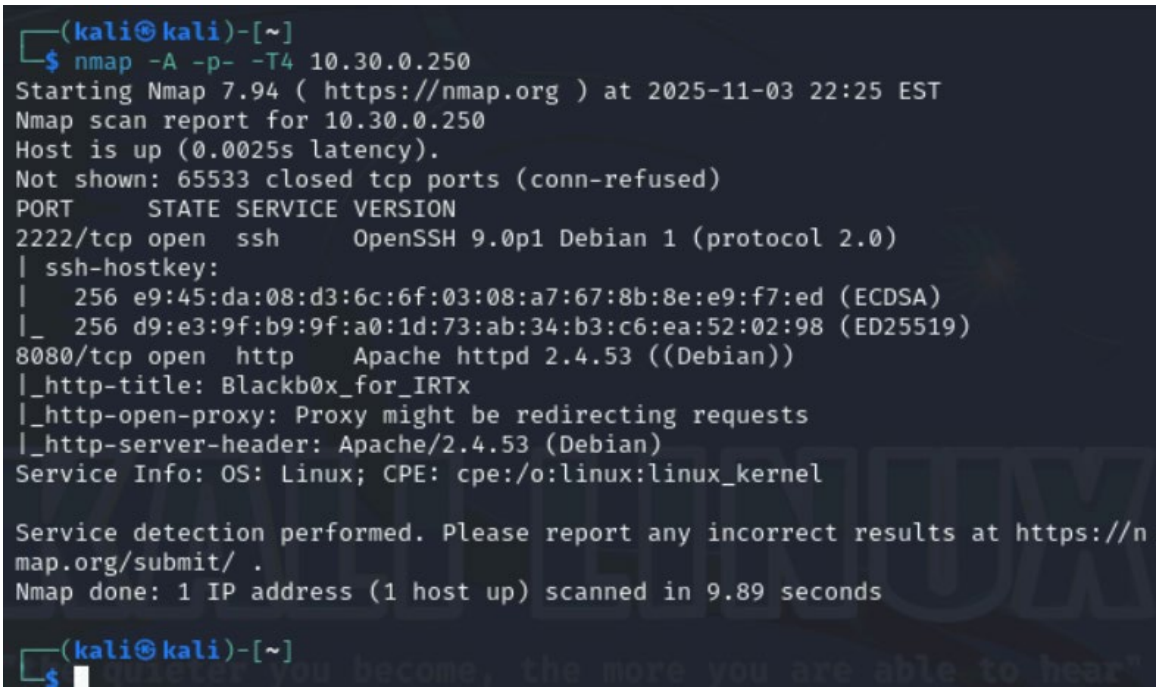
Target: 10.30.0.250

Objective: The objective of this activity is to identify open ports and live hosts on the target system or network, and to enumerate services running on those ports. These findings inform vulnerability assessment, service patching, and firewall configuration improvements. Nmap facilitates network discovery, service and version detection, and broader security assessment, and is commonly used in penetration testing to locate potential entry points for unauthorised access.

Type of Attack: Nmap Port Scanning

Attack steps and screenshots:

Run the following command
nmap -A -p- -T4 10.30.0.250



4.3 Web Content Scanning

Target: 10.30.0.250:8080

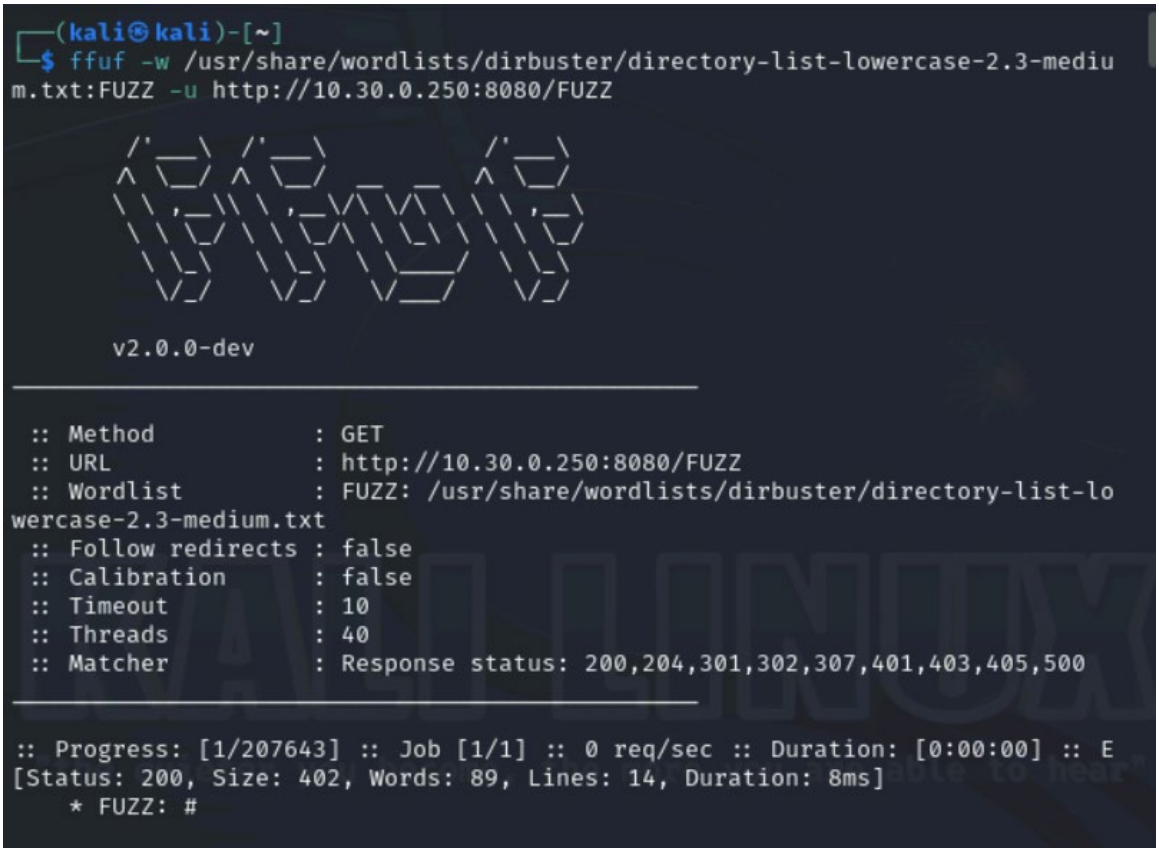
Objective: The objective of this activity is to discover hidden web content, directories, files, and other potential vulnerabilities in the target web application. These findings support assessment of the application's security posture, reveal misconfigurations, and identify possible points of exploitation.

Type of Attack: ffuf web fuzzing and directory brute-force

Attack steps and screenshots:

Run the following command

ffuf -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt:FUZZ -u http://10.30.0.250:8080/FUZZ



4.4 Spidering

Target: 10.30.0.250

Objective: The objective of this activity is to map the structure of the web application, uncover hidden or unlinked content, and identify potential vulnerabilities. This process provides a clearer understanding of the application's attack surface, exposes possible security weaknesses, and supports improvements to its overall security posture.

Type of Attack: Spidering with Zap

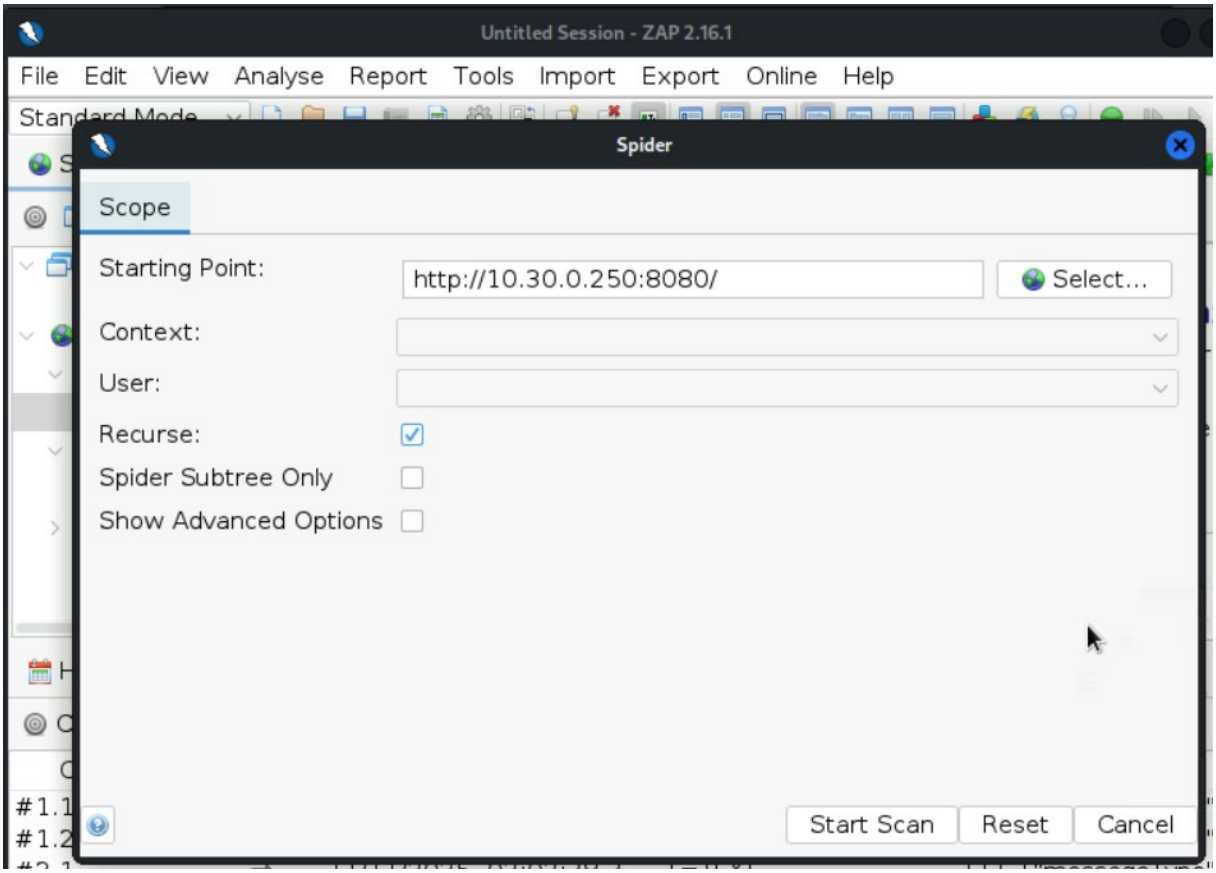
Attack steps and screenshots:

1) Start zap and open a browser

2) Browse to the web application http://10.30.0.250:8080

3) Right-click on the first successful GET request then select Attack > Spider

4) Check the Scope tab and click on [Start Scan]



4.5 Brute force attack

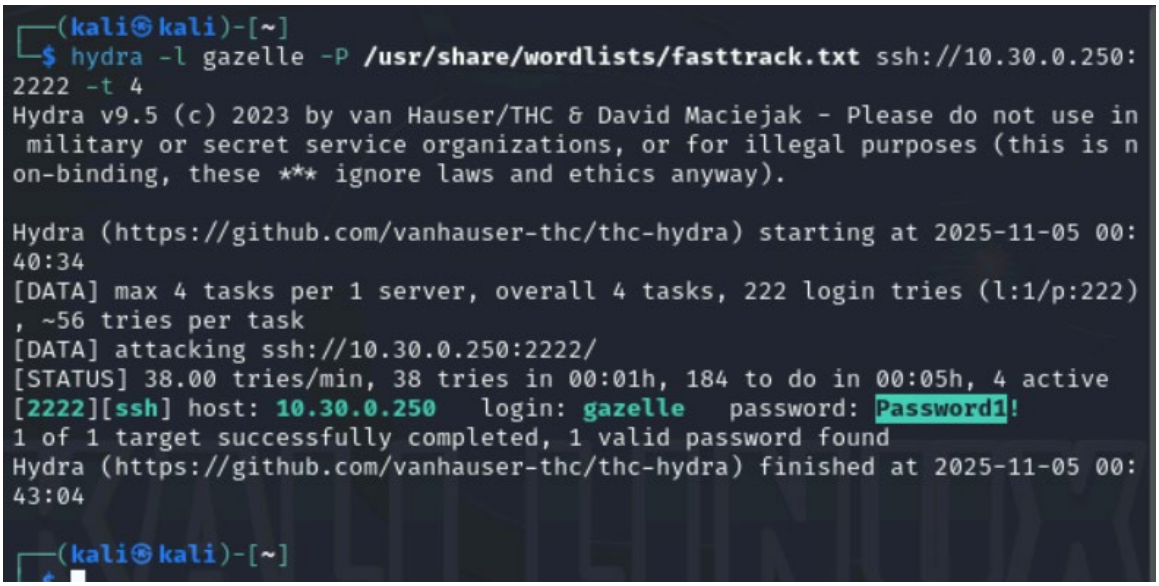
Target: 10.30.0.250:2222

Objective: The objective of this brute-force attack is to obtain unauthorised access by systematically testing a large dictionary of passwords until a valid credential is found.

Type of Attack: SSH password brute-forcing using Hydra

Attack steps and screenshots:

Run the following command
hydra -l gazelle -P /usr/share/wordlists/fasttrack.txt ssh://10.30.0.250:2222 -t 4



4.6 System Information Discovery

Target: 10.30.0.250

Objective: The LinPEAS script (linpeas.sh) was run to assess the Linux system's security posture and identify vulnerabilities or misconfigurations that might enable privilege escalation or compromise. The tool automates information gathering and analysis, revealing configuration details, permissions, installed software, network connections, and other useful indicators.

Type of Attack: Post-exploitation enumeration using linpeas.sh

Attack steps and screenshots:

Run linpeas.sh on the target machine

4.7 Privilege Escalation

Target: 10.30.0.250

Objective: The objective of privilege escalation is to obtain higher access rights on a system than those initially assigned to a user or process. Successful escalation gives an attacker increased control, access to sensitive data, the ability to run arbitrary commands, and the capacity to modify critical files, potentially compromising the entire system.

Type of Attack: Linux privilege escalation

Attack steps and screenshots:

Based on the output of linpeas.sh, the general attack steps for privilege escalation in Linux can include:

1. *Enumeration: Examine the linpeas.sh results to locate vulnerabilities, misconfigurations, or insecure permissions that could be leveraged for privilege escalation.*
2. *Exploiting SUID/SGID Permissions: Identify binaries with SUID or SGID bits set. These executables can allow a non-privileged user to run commands with the owner's or group's privileges — investigate each binary for known exploits or version-specific weaknesses.*
3. *Exploiting Weak File Permissions: Find files with overly permissive or incorrect permissions that a regular user could modify or overwrite (e.g., configuration files, scripts). Such files can often be manipulated to escalate privileges.*
4. *Exploiting Kernel Vulnerabilities: Check linpeas.sh output for kernel-level vulnerabilities. Research any identified issues and determine whether known kernel exploits apply to the target's kernel version.*
5. *Exploiting Services and Applications: Audit services and applications running with elevated privileges for misconfigurations, unpatched versions, or insecure defaults. These weaknesses may be exploitable to gain higher privileges.*
6. *Exploiting Weak User Permissions: Identify user accounts with inappropriate privileges, weak passwords, or misconfigurations. Addressing these account-level issues prevents straightforward escalation paths.*
7. *Exploiting Cron Jobs: Inspect scheduled cron jobs for tasks running with elevated privileges. Review the scripts and commands they invoke for misconfigurations, insecure file permissions, or exploitable logic that could be leveraged to escalate privileges.*
8. *Exploiting Environment Variables: Examine linpeas.sh output for environment variables that may contain sensitive values or be writable by non-privileged users. Determine whether these variables can be manipulated or contain secrets, and check for any known vulnerabilities or exploits related to those variables.*
9. *Exploiting Configuration Files: Locate configuration files that may store credentials (passwords, API keys, tokens) or sensitive settings. Verify file permissions and look for misconfigurations or exposed secrets that could be abused to obtain elevated privileges.*

It should be noted that exact privilege-escalation procedures and methods will differ based on system configuration, discovered vulnerabilities, and the Linux distribution and version involved.