

Task 16: Incident Response & Security Breach Simulation

Tools:

Primary: Linux Logs / Windows Event Viewer

Alternatives: TheHive (Community)

1. Introduction

A simulated security incident involving repeated failed SSH login attempts was conducted on a Kali Linux system. The objective was to detect suspicious activity, analyze logs, classify the attack, contain the threat, eradicate the root cause, and restore the system to a secure state.

The incident was identified as a **Brute Force SSH attack attempt** and was successfully mitigated through firewall blocking and system hardening.

2. Objective

- Simulate a basic security breach (multiple failed SSH logins)
- Analyze authentication logs
- Identify attacker IP
- Classify the attack
- Contain and eradicate the threat
- Restore and secure the system
- Document the complete incident lifecycle

3. Environment Details

- Operating System: Kali Linux
- Logging Mechanism: systemd journal
- SSH Service: OpenSSH
- Firewall Used: UFW (Uncomplicated Firewall)

4. Incident Simulation

Step 1: SSH Service Verification

`sudo systemctl status ssh`

If inactive:

`sudo systemctl start ssh`

```
(kali㉿kali)-[~]
$ sudo systemctl status ssh

[sudo] password for kali:
o ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: d>
   Active: inactive (dead)
     Docs: man:sshd(8)
           man:sshd_config(5)

(kali㉿kali)-[~]
$ sudo systemctl start ssh

(kali㉿kali)-[~]
$ sudo systemctl enable ssh

Synchronizing state of ssh.service with SysV service script with /usr/lib/syst
emd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink '/etc/systemd/system/sshd.service' → '/usr/lib/systemd/system/
ssh.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/ssh.service' → '/
usr/lib/systemd/system/ssh.service'.
```

Step 2: Simulating Failed Login Attempts

Multiple incorrect login attempts were performed:

`ssh fakeuser@localhost`

Wrong password entered multiple times (5–10 attempts).

This simulated a brute-force attack.

```
(kali㉿kali)-[~]
$ ssh fakeuser@localhost

The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:+6LqLP1aAB7ffP5hwQLVfYz+ztycVp7VHC6zxUz7i5s
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
fakeuser@localhost's password:
Permission denied, please try again.
fakeuser@localhost's password:
Permission denied, please try again.
fakeuser@localhost's password:
fakeuser@localhost: Permission denied (publickey,password).
```

5. Incident Identification (Log Analysis)

Since Kali uses systemd journal:

```
sudo journalctl -u ssh
```

To filter failed attempts:

```
sudo journalctl -u ssh | grep "Failed password"
```

Sample log entry:

Failed password for invalid user fakeuser from 127.0.0.1 port 54321 ssh2

```
(kali㉿kali)-[~]
└─$ sudo journalctl -u ssh

Feb 10 13:06:17 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell >
Feb 10 13:06:17 kali sshd[7993]: Server listening on 0.0.0.0 port 22.
Feb 10 13:06:17 kali sshd[7993]: Server listening on :: port 22.
Feb 10 13:06:17 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell s>
Feb 10 13:08:21 kali sshd[7993]: Received signal 15; terminating.
Feb 10 13:08:21 kali systemd[1]: Stopping ssh.service - OpenBSD Secure Shell >
Feb 10 13:08:21 kali systemd[1]: ssh.service: Deactivated successfully.
Feb 10 13:08:21 kali systemd[1]: Stopped ssh.service - OpenBSD Secure Shell s>
Feb 10 13:08:21 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell >
Feb 10 13:08:21 kali sshd[9010]: Server listening on 0.0.0.0 port 22.
Feb 10 13:08:21 kali sshd[9010]: Server listening on :: port 22.
Feb 10 13:08:21 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell s>
-- Boot a89dc27f562640e094d6c57c5289a13b --
Feb 13 11:52:00 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell >
Feb 13 11:52:01 kali sshd[88754]: Server listening on 0.0.0.0 port 22.
Feb 13 11:52:01 kali sshd[88754]: Server listening on :: port 22.
Feb 13 11:52:01 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell s>
Feb 13 11:53:09 kali sshd-session[89428]: Invalid user fakeuser from ::1 port>
Feb 13 11:53:15 kali sshd-session[89428]: pam_unix(sshd:auth): check pass; us>
Feb 13 11:53:15 kali sshd-session[89428]: pam_unix(sshd:auth): authentication>
Feb 13 11:53:15 kali sshd-session[89428]: pam_winbind(sshd:auth): getting pas>
Feb 13 11:53:15 kali sshd-session[89428]: pam_winbind(sshd:auth): pam_get_ite>
Feb 13 11:53:17 kali sshd-session[89428]: Failed password for invalid user fa>
```

Extracting Attacker IP

```
sudo journalctl -u ssh | grep "Failed password" | awk '{for(i=1;i<=NF;i++) if($i=="from")
print $(i+1)}' | sort | uniq -c
```

This identified repeated attempts from:

127.0.0.1

```
(kali㉿kali)-[~]
└─$ sudo journalctl -u ssh | grep "Failed"

Feb 13 11:53:17 kali sshd-session[89428]: Failed password for invalid user fak
euser from ::1 port 42884 ssh2
Feb 13 11:53:25 kali sshd-session[89428]: Failed password for invalid user fak
euser from ::1 port 42884 ssh2
Feb 13 11:53:31 kali sshd-session[89428]: Failed password for invalid user fak
euser from ::1 port 42884 ssh2
```

```
(kali㉿kali)-[~]
$ sudo journalctl -u ssh | grep "Failed password" | awk '{for(i=1;i≤NF;i++)
if($i="from") print $(i+1)}' | sort | uniq -c
3 ::1
```

6. Incident Classification

- Attack Type: SSH Brute Force Attack
- Severity Level: High
- Impact: Unauthorized access attempt
- Risk Level: Potential system compromise if credentials guessed

7. Incident Timeline

Time	Event
10:05 AM	Multiple failed SSH login attempts generated
10:08 AM	Log analysis performed using journalctl
10:10 AM	Suspicious IP identified
10:12 AM	IP blocked via firewall
10:20 AM	System updated and hardened
10:30 AM	Monitoring confirmed no further malicious activity

8. Containment

Firewall enabled:

```
sudo ufw enable
```

```
(kali㉿kali)-[~]
$ sudo ufw enable
Firewall is active and enabled on system startup
```

Attacker IP blocked:

```
sudo ufw deny from 127.0.0.1
```

```
(kali㉿kali)-[~]  
$ sudo ufw deny from 127.0.0.1  
Rule added
```

Firewall status verified:

sudo ufw status

Containment successfully stopped further attempts.

```
(kali㉿kali)-[~]  
$ sudo ufw status  
Status: active  
  
To Action From  
--  
23 DENY Anywhere  
22 ALLOW Anywhere  
80 ALLOW Anywhere  
443 ALLOW Anywhere  
21 DENY Anywhere  
Anywhere DENY 192.168.1.100  
22/tcp ALLOW Anywhere  
Anywhere DENY 127.0.0.1  
23 (v6) DENY Anywhere (v6)  
22 (v6) ALLOW Anywhere (v6)  
80 (v6) ALLOW Anywhere (v6)  
21 (v6) DENY Anywhere (v6)  
443 (v6) ALLOW Anywhere (v6)  
22/tcp (v6) ALLOW Anywhere (v6)
```

9. Eradication (Root Cause Fix)

Root Cause

- SSH service exposed
- No brute-force protection enabled
- Weak monitoring configuration

Security Improvements Applied

1. System Update

sudo apt update && sudo apt upgrade -y

2. Disabled Root Login

Edited SSH config:

```
sudo nano /etc/ssh/sshd_config
```

Changed:

PermitRootLogin no

Restarted SSH:

```
sudo systemctl restart ssh
```

3. Installed Fail2Ban

```
sudo apt install fail2ban -y
```

```
sudo systemctl enable fail2ban
```

```
sudo systemctl start fail2ban
```

Fail2Ban automatically blocks brute-force attackers.

```
(kali㉿kali)-[~]
└─$ sudo apt update
    sudo apt upgrade -y

Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.7 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.1 M
B]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [117 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [271
kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [186 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [88
8 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages
[11.8 kB]
Fetched 74.3 MB in 28s (2,665 kB/s)
2208 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required
:
  amass-common                libsigsegv2
  firebird3.0-common          libsnmp40t64
  firebird3.0-common-doc      libsphinxbase3t64
  firmware-ti-connectivity    libsqlcipher1
  icu-devtools                libswscale8
  libabsl20230802             libtag1v5
  libaudio2                   libtag1v5-vanilla
```

```
(kali㉿kali)-[~]
└─$ sudo nano /etc/ssh/sshd_config

[sudo] password for kali:
```

10. Recovery

SSH service restarted:

sudo systemctl restart ssh

```
(kali@kali)-[~]  
$ sudo systemctl restart ssh
```

Live monitoring performed:

sudo journalctl -u ssh -f

```
(kali@kali)-[~]  
$ sudo journalctl -u ssh -f  
  
Feb 13 12:28:50 kali sshd[169227]: Server listening on :: port 22.  
Feb 13 12:28:50 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell se  
rver.  
Feb 13 12:32:46 kali systemd[1]: Stopping ssh.service - OpenBSD Secure Shell s  
erver ...  
Feb 13 12:32:46 kali sshd[169227]: Received signal 15; terminating.  
Feb 13 12:32:46 kali systemd[1]: ssh.service: Deactivated successfully.  
Feb 13 12:32:46 kali systemd[1]: Stopped ssh.service - OpenBSD Secure Shell se  
rver.  
Feb 13 12:32:46 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell s  
erver ...  
Feb 13 12:32:46 kali sshd[181863]: Server listening on 0.0.0.0 port 22.  
Feb 13 12:32:46 kali sshd[181863]: Server listening on :: port 22.  
Feb 13 12:32:46 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell se  
rver.
```

No further suspicious activity observed.

System restored to secure state.

11. Lessons Learned

- Log monitoring is critical for early detection
- SSH should not be exposed without protection
- Fail2Ban significantly reduces brute-force risk
- Strong authentication mechanisms are necessary

12. Preventive Recommendations

1. Enable SSH key-based authentication
2. Disable password authentication
3. Change default SSH port
4. Enable firewall permanently

5. Install Fail2Ban
6. Use strong password policies
7. Regularly monitor logs

13. Conclusion

The simulated SSH brute-force attack was successfully detected, analyzed, contained, and mitigated. Proper log analysis using journalctl, firewall configuration, and system hardening ensured that the system was restored securely.