# Brute Force Attack Analysis – Write-Up

## 1. Overview

This project involved analyzing SSH authentication logs from a CentOS system that had been targeted by a brute-force attack. The objective was to extract attempted usernames, identify invalid user attempts, cross-reference them with system accounts, and provide security recommendations to prevent future attacks.

---

## 2. Log Access and Extraction

The provided log archive was downloaded in Kali Linux and extracted. The directory contained multiple `secure` log files from CentOS, which hold all SSH authentication events.

---

## 3. Username Extraction (Valid Login Attempts)

The following command was used to extract all attempted login usernames from failed SSH authentication events:

```
grep "Failed password for" secure* | grep -v "invalid user" | awk '{print
$9}' > usernames.log
```

**Result:**
The file `usernames.log` showed numerous brute-force attempts against existing system accounts, primarily targeting the `root`user.

This indicates a high-risk automated attack attempting to gain privileged access.

---

## 4. Invalid User Extraction

To extract all usernames that do **not** exist on the system (typical brute-force dictionary attempts), the following command was used:

```
grep "Invalid user" secure* | awk '{print $8}' | sort -u > abc.txt
```

**Result:**
`abc.txt` contained a large list of invalid usernames such as:

```
tom
ubuntu
uniadmin
zookeeper
```

```
root1
uftp
...
```

This pattern confirms an automated botnet attack using a large username dictionary.

---

# 5. Cross-Reference With System Accounts

A shell script was created to compare all extracted usernames against the system's `/etc/passwd` file:

```
#!/bin/bash

while read user; do
    if grep -q "^$user:" /etc/passwd; then
        echo "EXISTS: $user"
    else
        echo "NOT FOUND: $user"
    fi
done < abc.txt
```

**Result:**
All usernames listed in `abc.txt` returned **NOT FOUND**, meaning **no legitimate internal accounts were targeted** besides `root`.

This confirms the attack was purely external and automated.

---

# 6. Attack Interpretation

Based on the logs:

- The attacker performed a **high-volume SSH brute-force attack**.
- Hundreds of login attempts were directed at `root`, indicating a privilege-escalation motive.
- Thousands of invalid usernames were attempted, matching known global SSH botnet patterns.
- No internal or legitimate user accounts were impacted.

This activity represents a **high-severity external brute-force attack**.

---

# 7. Recommended Security Enhancements

These controls would prevent or significantly reduce similar attacks:

1. **Disable root SSH login**
   `PermitRootLogin no`
2. **Disable password authentication**
   `PasswordAuthentication no`
   (Use SSH keys only)
3. **Enable Fail2Ban**
   Automatically blocks IPs after repeated failures.
4. **Firewall restrictions**
   Allow SSH only from trusted IPs.
5. **Change default SSH port**
   Helps reduce botnet scanning.
6. **Enable MFA for SSH**
   Adds an additional security layer.

---

## 8. Continuous Monitoring Recommendations

- Implement log monitoring tools (Logwatch, Splunk, ELK).
- Enable alerts for repeated authentication failures.
- Monitor SSH access patterns and reputation-check suspicious IPs.
- Perform regular security configuration audits.