

# Proof Of Concept

## Linux Security- Exploitation & Hardening

### Task 2: Remote Access & SSH Hardening

#### 1. Executive Summary

This PoC demonstrates the risks associated with insecure SSH configurations, such as allowing root login and password-based authentication. It includes enabling SSH, performing a brute-force attack, and then hardening the SSH configuration to prevent unauthorized access.

#### 2. Objectives

- **Setup:** Enable SSH on a Linux machine, allow root login, and enable password authentication.
- **Exploit:** Perform a brute-force attack on SSH using tools like hydra or medusa.
- **Mitigation:** Disable root login, enable key-based authentication, and configure fail2ban to prevent brute-force attacks.

#### 3. Setup

##### 3.1. Enable SSH and Configure Insecure Settings

###### 1. Start and Enable SSH Service:

```
(llamafart@jaivanti)-[~]  
$ sudo systemctl start ssh  
[sudo] password for llamafart:  
  
(llamafart@jaivanti)-[~]  
$ sudo systemctl enable ssh  
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
```

## 2. Edit SSH Configuration to allow root login and password authentication:

Open the SSH configuration file:

```
(llamafart@jaivanti)-[~]  
$ sudo nano /etc/ssh/sshd_config  
  
(llamafart@jaivanti)-[~]  
$ sudo systemctl restart ssh
```

Modify the following lines:

```
#LoginGraceTime 2m  
PermitRootLogin yes  
#StrictModes yes  
#MaxAuthTries 6  
#MaxSessions 10  
  
#PubkeyAuthentication yes  
  
# Expect .ssh/authorized_keys2 to be disregarded by default in future.  
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2  
  
#AuthorizedPrincipalsFile none  
  
#AuthorizedKeysCommand none  
#AuthorizedKeysCommandUser nobody  
  
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts  
#HostbasedAuthentication no  
# Change to yes if you don't trust ~/.ssh/known_hosts for  
# HostbasedAuthentication  
#IgnoreUserKnownHosts no  
# Don't read the user's ~/.rhosts and ~/.shosts files  
#IgnoreRhosts yes  
  
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication yes  
#PermitEmptyPasswords no
```

## 4. Exploitation

### 4.1. Perform a Brute-Force Attack Using Hydra

#### 1. Create a Wordlist & Run Hydra to Brute-Force SSH:

```
(llamafart@jaivanti)-[~]  
$ nano wordlist.txt  
  
(llamafart@jaivanti)-[~]  
$ hydra -L wordlist.txt -P wordlist.txt ssh://localhost  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-11 22:09:33  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tas  
[DATA] max 9 tasks per 1 server, overall 9 tasks, 9 login tries (l:3/p:3), ~1 try per task  
[DATA] attacking ssh://localhost:22/  
[22][ssh] host: localhost login: llamafart password: 121005  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-11 22:09:37
```

## 5. Mitigation

### 5.1. Disable Root Login and Password Authentication

#### 1. Edit SSH Configuration:

Open the SSH configuration file and modify the following lines:

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

Save and exit the file, then restart the SSH service:

```
(llamafart@jaivanti)-[~]
$ sudo nano /etc/ssh/sshd_config
[sudo] password for llamafart:

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

## 5.2. Enable Key-Based Authentication

### 1. Generate SSH Keys (on the client machine):

```
(llamafart@jaivanti)-[~]
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/llamafart/.ssh/id_rsa): urmom
Enter passphrase for "urmom" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in urmom
Your public key has been saved in urmom.pub
The key fingerprint is:
SHA256:16TJOsVCJWmH5yiz2Hs7SkLMJq3tIfv4CAFZD7KiNOQ llamafart@jaivanti
The key's randomart image is:
+---[RSA 4096]-----+
|..o      .o.      |
|o= o      +oo      |
|=E .  ..= .      |
|= .+  o..o.=      |
|o . *o +S B .      |
| . *. o =         |
|. o + ..o         |
|. * +. o.         |
| +o+ .o.o         |
+---[SHA256]-----+
```

### 2. Copy the Public Key to the Server:

```
(llamafart@jaivanti)-[~]
$ ssh-copy-id llamafart@localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ssh-add -L
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:u0Nq5iThujM2cIsQhcsiS2rGQ3kpuNxYRcxSBHSbZPwQ.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system.
(if you think this is a mistake, you may want to use -f option)
```

## 5.3. Configure Fail2Ban to Prevent Brute-Force Attacks

### 1. Install Fail2Ban & Configure Fail2Ban for SSH:

Create a local configuration file:

```
(llamafart@jaivanti)-[~]
$ sudo apt install fail2ban -y
fail2ban is already the newest version (1.1.0-7).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1048
(llamafart@jaivanti)-[~]
$ sudo nano /etc/fail2ban/jail.local
```

Add the following lines:

```
GNU nano 8.2 /etc/fail2ban/jail.local *
[sshd]
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 3600
enabled = true
maxretry = 3
bantime = 600
```

Save and exit the file, then restart Fail2Ban:

```
(llamafart@jaivanti)-[~]
$ sudo systemctl restart fail2ban
```

## 2. Verify Fail2Ban Status:

```
(llamafart@jaivanti)-[~]
$ ssh llamafart@localhost
Linux jaivanti 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.11.2-1kali1 (2024-10-15) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar 11 12:50:45 2025 from 10.12.8.147
(llamafart@jaivanti)-[~]
$ ssh root@localhost
root@localhost: Permission denied (publickey).
```

## 6. Conclusion:

This PoC successfully demonstrated the risks of insecure SSH configurations and the effectiveness of hardening measures. By disabling root login, enforcing key-based authentication, and preventing brute-force attacks, the SSH service was secured