Use Fail2ban to scans log files and bans IPs that show the malicious signs

Aim

To use Fail2Ban to scan log files and ban IPs that show malicious signs.

Introduction

Fail2Ban is a versatile and powerful security tool designed to protect Linux servers from malicious activities by monitoring log files for suspicious patterns and automatically blocking IP addresses exhibiting such behavior. This open-source intrusion prevention framework operates by scanning log files, typically those associated with system services like SSH, Apache, or Nginx, and identifies patterns indicative of brute-force attacks, login failures, or other malicious activities. Once a potential threat is detected, Fail2Ban takes proactive measures by dynamically updating firewall rules to block the offending IP addresses, thereby preventing unauthorized access and enhancing the overall security posture of the system.

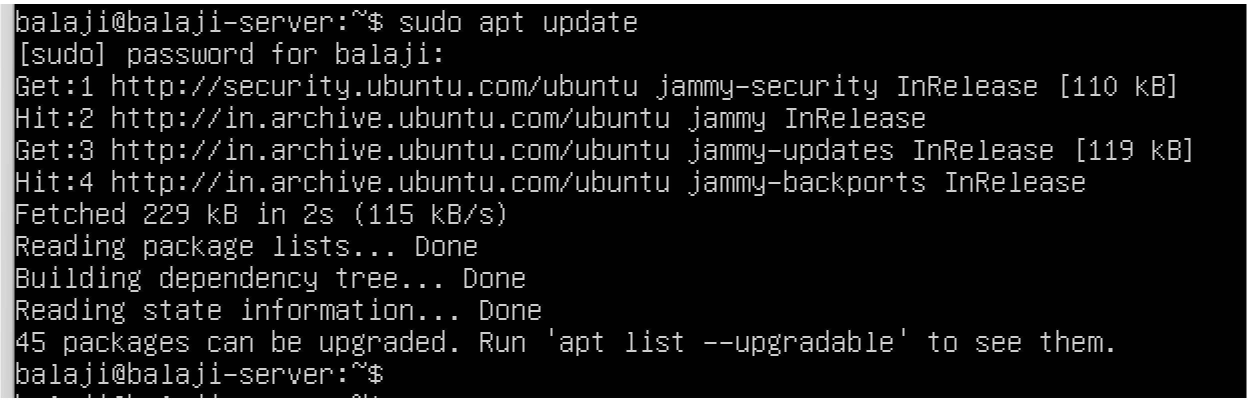
Fail2Ban is highly configurable, allowing users to define custom filters and adjust parameters based on their specific security requirements. This flexibility makes it suitable for a wide range of applications and environments, from individual servers to large-scale enterprise networks. Additionally, Fail2Ban's ability to work seamlessly with different log file formats and protocols contributes to its adaptability.

Procedure

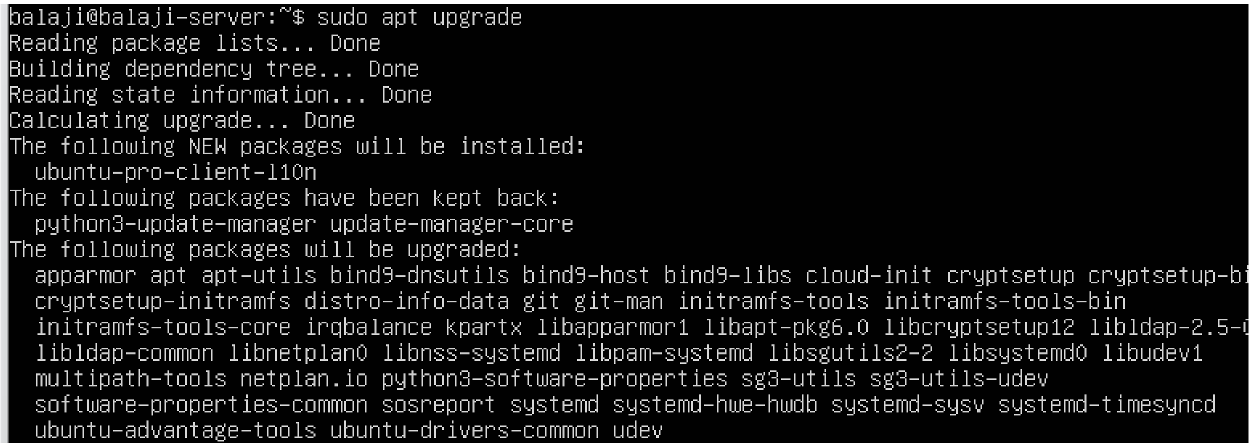
1. Open the Ubuntu Server virtual machine and update and upgrade the system using the following commands:

sudo apt update   -> Refresh the package list.

sudo apt upgrade -> Installs the latest available updates on an Ubuntu Server.

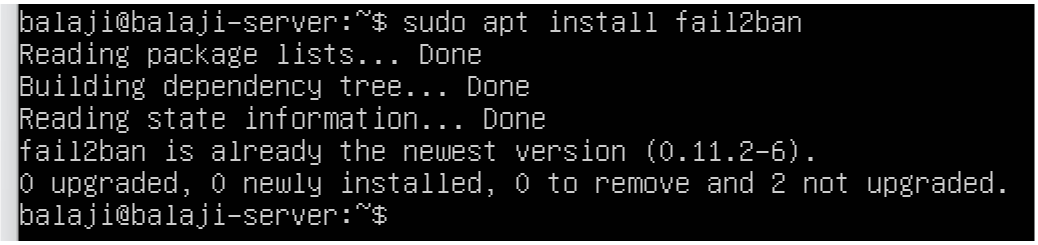


```
balaji@balaji-server:~$ sudo apt update
[sudo] password for balaji:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 229 kB in 2s (115 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
45 packages can be upgraded. Run 'apt list --upgradable' to see them.
balaji@balaji-server:~$
```



```
balaji@balaji-server:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  ubuntu-pro-client-l10n
The following packages have been kept back:
  python3-update-manager update-manager-core
The following packages will be upgraded:
  apparmor apt apt-utils bind9-dnsutils bind9-host bind9-libs cloud-init cryptsetup cryptsetup-bi
  cryptsetup-initramfs distro-info-data git git-man initramfs-tools initramfs-tools-bin
  initramfs-tools-core irqbalance kpartx libapparmor1 libapt-pkg6.0 libcryptsetup12 libldap-2.5-
  libldap-common libnetplan0 libnss-systemd libpam-systemd libsgutils2-2 libsystemd0 libudev1
  multipath-tools netplan.io python3-software-properties sg3-utils sg3-utils-udev
  software-properties-common sosreport systemd systemd-hwe-hwdb systemd-sysv systemd-timesyncd
  ubuntu-advantage-tools ubuntu-drivers-common udev
```

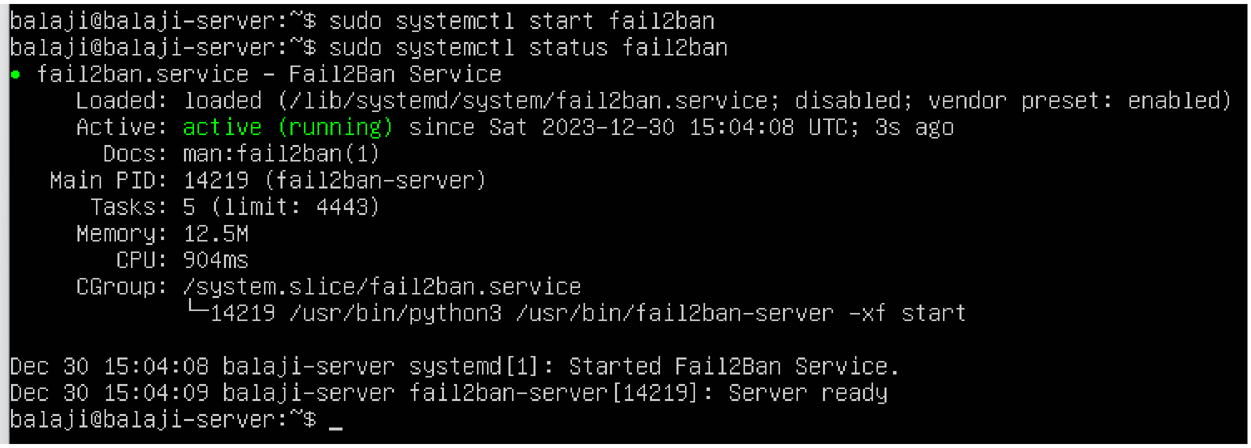
2. Install Fail2ban with root privileges using the following command: sudo apt install fail2ban. This command installs the Fail2ban software on your system.



```
balaji@balaji-server:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fail2ban is already the newest version (0.11.2-6).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
balaji@balaji-server:~$
```

3. Start Fail2ban and check its status with the following commands:

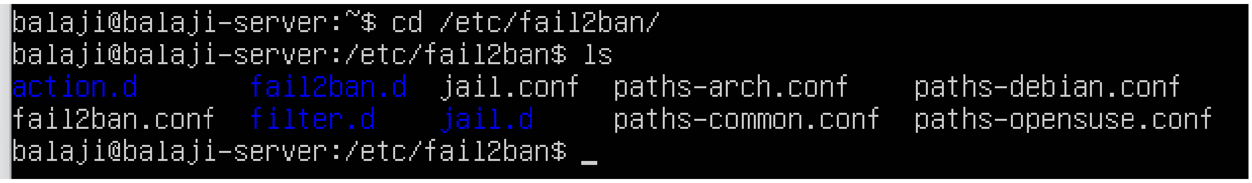
sudo systemctl start fail2ban
sudo systemctl status fail2ban



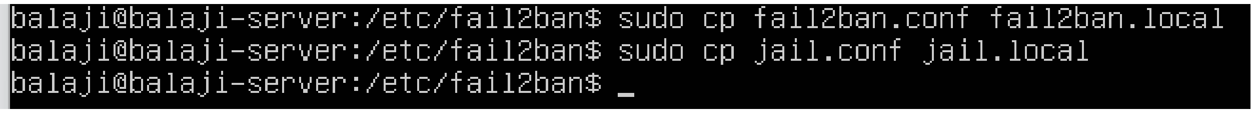
```
balaji@balaji-server:~$ sudo systemctl start fail2ban
balaji@balaji-server:~$ sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; disabled; vendor preset: enabled)
     Active: active (running) since Sat 2023-12-30 15:04:08 UTC; 3s ago
       Docs: man:fail2ban(1)
   Main PID: 14219 (fail2ban-server)
      Tasks: 5 (limit: 4443)
     Memory: 12.5M
        CPU: 904ms
     CGroup: /system.slice/fail2ban.service
             └─14219 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Dec 30 15:04:08 balaji-server systemd[1]: Started Fail2Ban Service.
Dec 30 15:04:09 balaji-server fail2ban-server[14219]: Server ready
balaji@balaji-server:~$ _
```

4. Fail2Ban's configuration files are typically found in the /etc/fail2ban/ directory. The main configuration file is jail.conf. Create local copies of Fail2Ban's configuration files with the following commands:

sudo cp fail2ban.conf fail2ban.local
sudo cp jail.conf jail.local

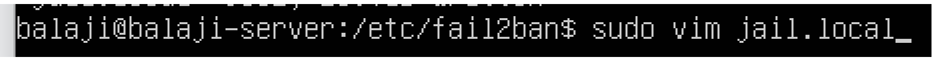


```
balaji@balaji-server:~$ cd /etc/fail2ban/
balaji@balaji-server:/etc/fail2ban$ ls
action.d       fail2ban.d  jail.conf  paths-arch.conf    paths-debian.conf
fail2ban.conf  filter.d    jail.d     paths-common.conf  paths-opensuse.conf
balaji@balaji-server:/etc/fail2ban$ _
```

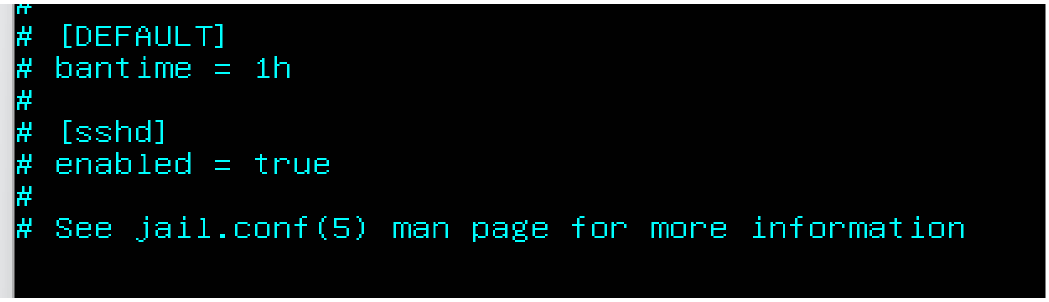


```
balaji@balaji-server:/etc/fail2ban$ sudo cp fail2ban.conf fail2ban.local
balaji@balaji-server:/etc/fail2ban$ sudo cp jail.conf jail.local
balaji@balaji-server:/etc/fail2ban$ _
```

5. Open the jail.local file and check the status of SSH by using the following command:
sudo vim jail.local

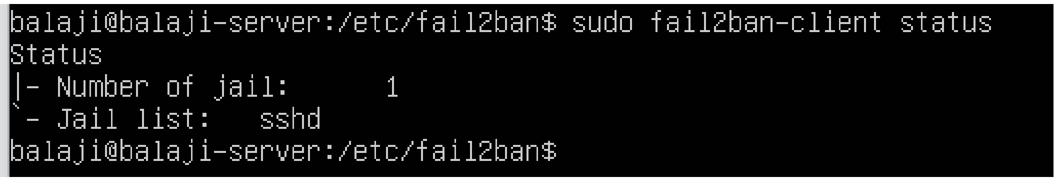


```
balaji@balaji-server:/etc/fail2ban$ sudo vim jail.local_
```



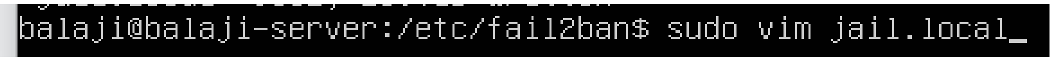
```
#
# [DEFAULT]
# bantime = 1h
#
# [sshd]
# enabled = true
#
# See jail.conf(5) man page for more information
```

6. The command ‘sudo fail2ban-client status’ is used to check the status of Fail2ban. It provides information about the currently banned IP addresses and the jails that are active. When you run this command, it will display the current status of Fail2ban on your system.

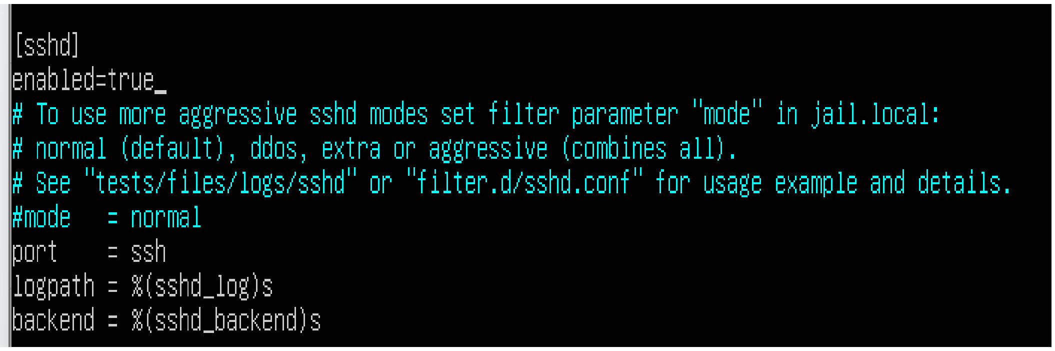


```
balaji@balaji-server:/etc/fail2ban$ sudo fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:   sshd
balaji@balaji-server:/etc/fail2ban$
```

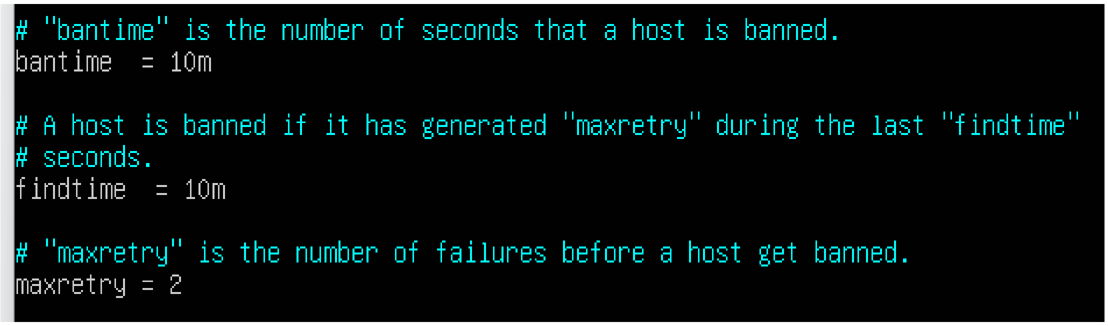
7. Open the 'jail.local' file and set 'SSH' to 'true,' then change 'maxretry' to '2.' The concept is that after two unsuccessful logins within 10 minutes ('findtime'), the respective IP address will be banned for a duration ('bantime') of 10 minutes.



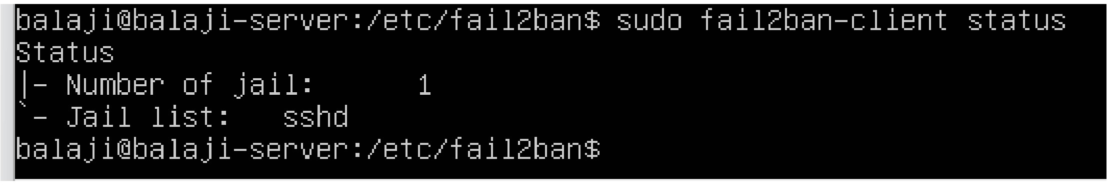
```
balaji@balaji-server:/etc/fail2ban$ sudo vim jail.local
```



```
[sshd]
enabled=true
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

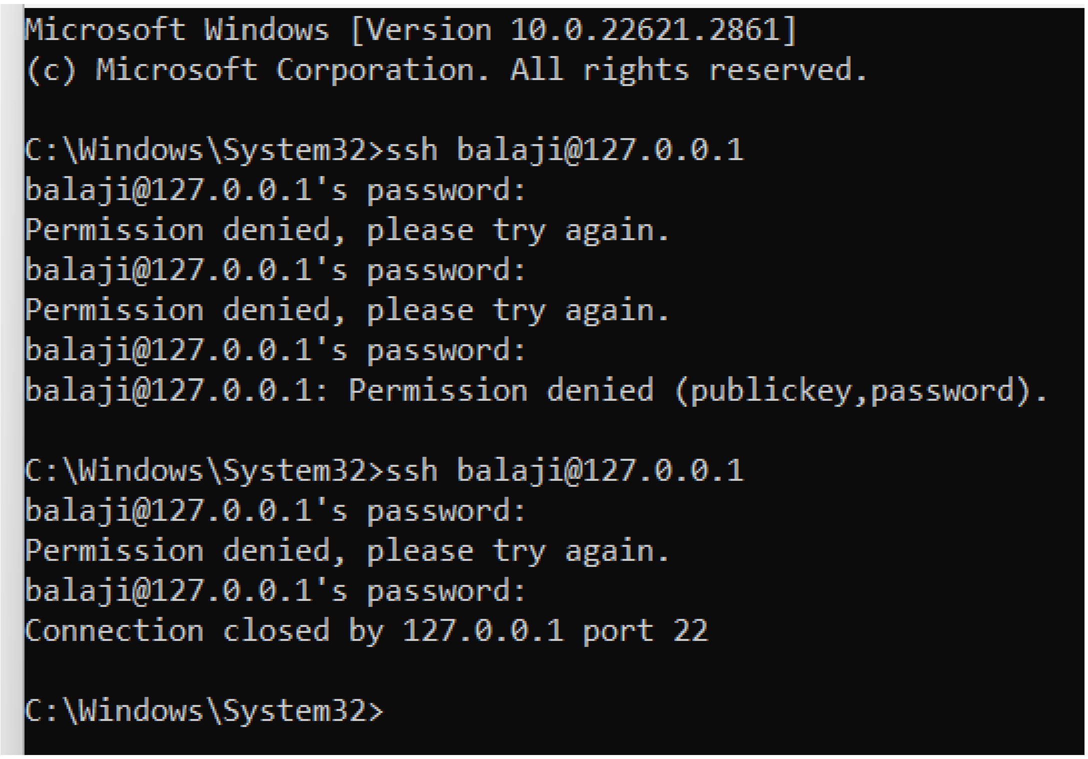


```
# "bantime" is the number of seconds that a host is banned.
bantime  = 10m

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 10m

# "maxretry" is the number of failures before a host get banned.
maxretry = 2
```



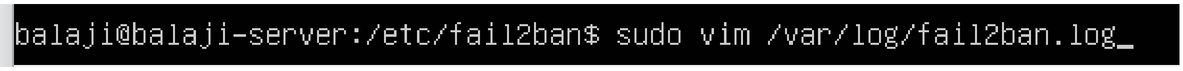
```
balaji@balaji-server:/etc/fail2ban$ sudo fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:   sshd
balaji@balaji-server:/etc/fail2ban$
```

8. Open the Windows Command Prompt as an Administrator and run the command: ‘ssh [username]@[hostname or IP address]’. Since we fixed the 'maxretry' to 2, the respective IP address will be banned after two unsuccessful attempts.

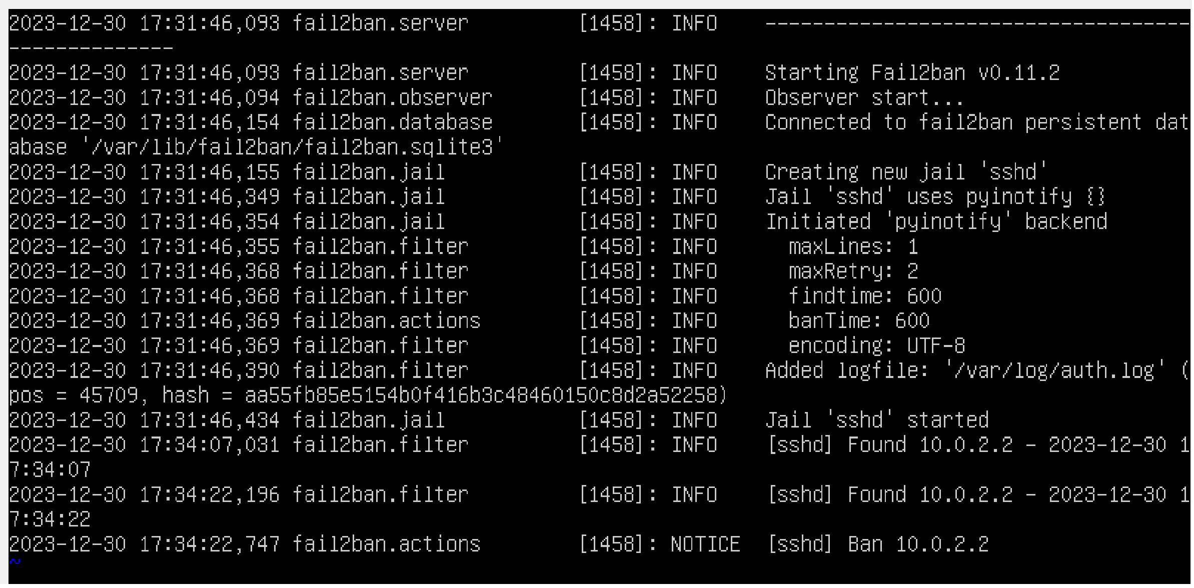


```
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>ssh balaji@127.0.0.1
balaji@127.0.0.1's password:
Permission denied, please try again.
balaji@127.0.0.1's password:
Permission denied, please try again.
balaji@127.0.0.1's password:
balaji@127.0.0.1: Permission denied (publickey,password).

C:\Windows\System32>ssh balaji@127.0.0.1
balaji@127.0.0.1's password:
Permission denied, please try again.
balaji@127.0.0.1's password:
Connection closed by 127.0.0.1 port 22

C:\Windows\System32>
```

9. Open the Fail2ban log file on the server using the command : ‘sudo vim /var/log/fail2ban.log’. This command allows you to view the Fail2ban log and check its status.



```
balaji@balaji-server:/etc/fail2ban$ sudo vim /var/log/fail2ban.log_
```



```
2023-12-30 17:31:46,093 fail2ban.server         [1458]: INFO    ------------------------------------
--------------
2023-12-30 17:31:46,093 fail2ban.server         [1458]: INFO    Starting Fail2ban v0.11.2
2023-12-30 17:31:46,094 fail2ban.observer       [1458]: INFO    Observer start...
2023-12-30 17:31:46,154 fail2ban.database       [1458]: INFO    Connected to fail2ban persistent dat
abase '/var/lib/fail2ban/fail2ban.sqlite3'
2023-12-30 17:31:46,155 fail2ban.jail           [1458]: INFO    Creating new jail 'sshd'
2023-12-30 17:31:46,349 fail2ban.jail           [1458]: INFO    Jail 'sshd' uses pyinotify {}
2023-12-30 17:31:46,354 fail2ban.jail           [1458]: INFO    Initiated 'pyinotify' backend
2023-12-30 17:31:46,355 fail2ban.filter         [1458]: INFO      maxLines: 1
2023-12-30 17:31:46,368 fail2ban.filter         [1458]: INFO      maxRetry: 2
2023-12-30 17:31:46,368 fail2ban.filter         [1458]: INFO      findtime: 600
2023-12-30 17:31:46,369 fail2ban.actions        [1458]: INFO      banTime: 600
2023-12-30 17:31:46,369 fail2ban.filter         [1458]: INFO      encoding: UTF-8
2023-12-30 17:31:46,390 fail2ban.filter         [1458]: INFO    Added logfile: '/var/log/auth.log' (
pos = 45709, hash = aa55fb85e5154b0f416b3c48460150c8d2a52258)
2023-12-30 17:31:46,434 fail2ban.jail           [1458]: INFO    Jail 'sshd' started
2023-12-30 17:34:07,031 fail2ban.filter         [1458]: INFO    [sshd] Found 10.0.2.2 - 2023-12-30 1
7:34:07
2023-12-30 17:34:22,196 fail2ban.filter         [1458]: INFO    [sshd] Found 10.0.2.2 - 2023-12-30 1
7:34:22
2023-12-30 17:34:22,747 fail2ban.actions        [1458]: NOTICE  [sshd] Ban 10.0.2.2
~
```

10. The corresponding IP address will be automatically unbanned after a duration of 10 minutes.

Result

In this exercise, we learned to use Fail2Ban for enhanced server security by scanning logs, detecting malicious signs, and blocking suspicious IPs. The exercise covered installation, customization, and monitoring, showcasing Fail2Ban's effectiveness in dynamically preventing unauthorized access and strengthening overall system security.