Day 43

Exploitation Analyst

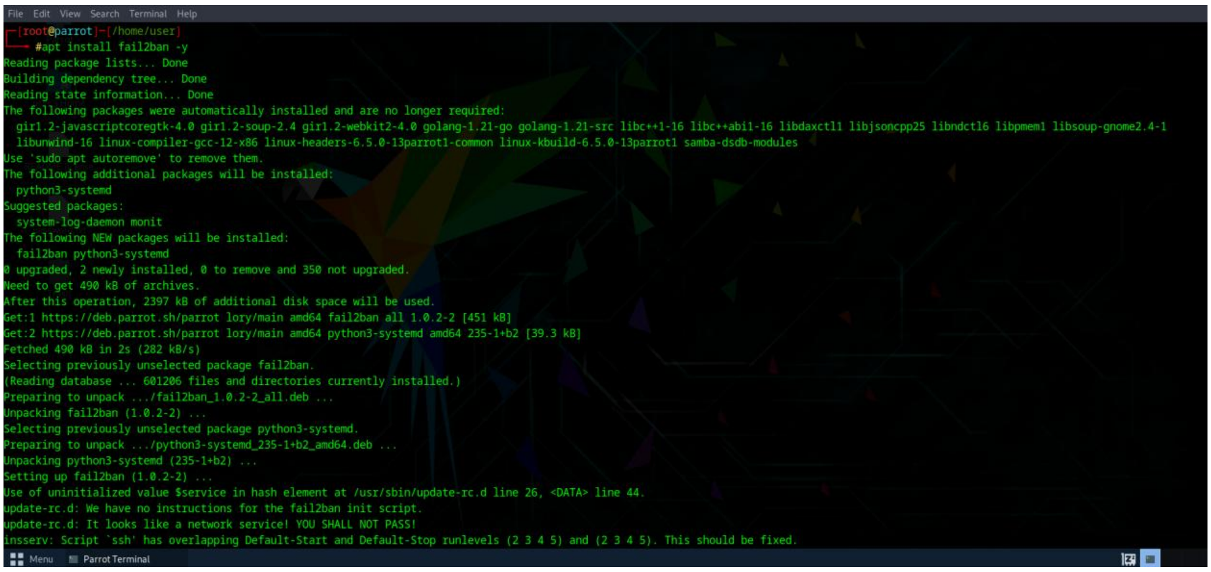
Firewalls and TCP wrappers:

Fail2ban:

What is Fail2ban?

Fail2ban is a security tool that helps protect Linux servers from brute-force attacks. It works by monitoring log files (like SSH, Apache, Nginx, etc.) for repeated failed login attempts or suspicious activity. When it detects too many failures from the same IP, it automatically bans that IP by adding a temporary firewall rule (using iptables or firewalld). This reduces the risk of attackers guessing passwords or overwhelming services.

Installing it: apt install fail2ban -y

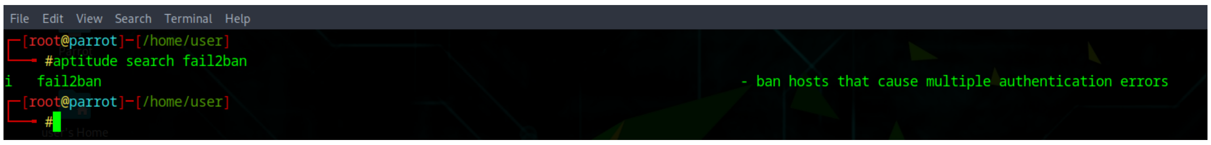


```
┌─[root@parrot]─[/home/user]
└──╼ #apt install fail2ban -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-webkit2-4.0 golang-1.21-go golang-1.21-src libc++1-16 libc++abi1-16 libdaxctl1 libjsoncpp25 libndctl6 libpmem1 libsoup-gnome2.4-1
  libunwind-16 linux-compiler-gcc-12-x86 linux-headers-6.5.0-13parrot1-common linux-kbuild-6.5.0-13parrot1 samba-dsdb-modules
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-systemd
Suggested packages:
  system-log-daemon monit
The following NEW packages will be installed:
  fail2ban python3-systemd
0 upgraded, 2 newly installed, 0 to remove and 350 not upgraded.
Need to get 490 kB of archives.
After this operation, 2397 kB of additional disk space will be used.
Get:1 https://deb.parrot.sh/parrot lory/main amd64 fail2ban all 1.0.2-2 [451 kB]
Get:2 https://deb.parrot.sh/parrot lory/main amd64 python3-systemd amd64 235-1+b2 [39.3 kB]
Fetched 490 kB in 2s (282 kB/s)
Selecting previously unselected package fail2ban.
(Reading database ... 601206 files and directories currently installed.)
Preparing to unpack .../fail2ban_1.0.2-2_all.deb ...
Unpacking fail2ban (1.0.2-2) ...
Selecting previously unselected package python3-systemd.
Preparing to unpack .../python3-systemd_235-1+b2_amd64.deb ...
Unpacking python3-systemd (235-1+b2) ...
Setting up fail2ban (1.0.2-2) ...
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
update-rc.d: We have no instructions for the fail2ban init script.
update-rc.d: It looks like a network service! YOU SHALL NOT PASS!
insserv: Script `ssh' has overlapping Default-Start and Default-Stop runlevels (2 3 4 5) and (2 3 4 5). This should be fixed.
```

Exploring Fail2ban:

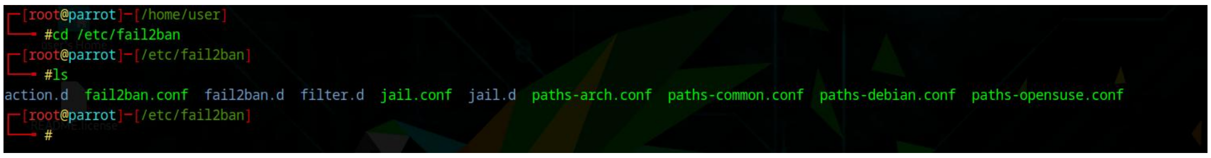
Steps:

Search for the fail2ban using: aptitude search fail2ban



```
┌─[root@parrot]─[/home/user]
└──╼ #aptitude search fail2ban
i   fail2ban                                                        - ban hosts that cause multiple authentication errors
┌─[root@parrot]─[/home/user]
└──╼ #
```

To use it, change the directory: use cd /etc/fail2ban



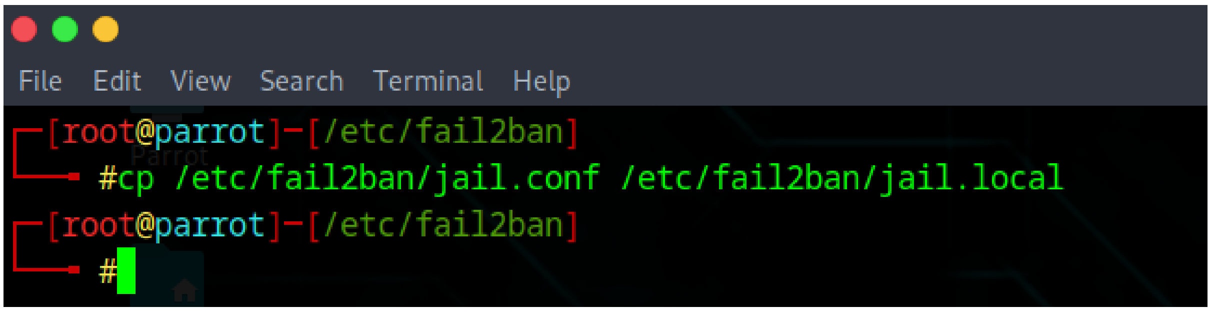
```
┌─[root@parrot]─[/home/user]
└──╼ #cd /etc/fail2ban
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #ls
action.d  fail2ban.conf  fail2ban.d  filter.d  jail.conf  jail.d  paths-arch.conf  paths-common.conf  paths-debian.conf  paths-opensuse.conf
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #
```

Here’s what those files/folders mean in /etc/fail2ban:

- **fail2ban.conf** → Main configuration (logging, backend).
- **jail.conf** → Default “jails” (services to protect). Never edit directly; copy to jail.local or use jail.d/.
- **jail.d/** → Place your custom jail configs here.
- **filter.d/** → Contains regex filters Fail2ban uses to detect malicious log entries.
- **action.d/** → Defines what to do when an IP is banned (usually add an iptables rule).
- **paths-*.conf** → OS-specific log file paths (so Fail2ban knows where to look).

Never edit jail.conf directly. Instead:

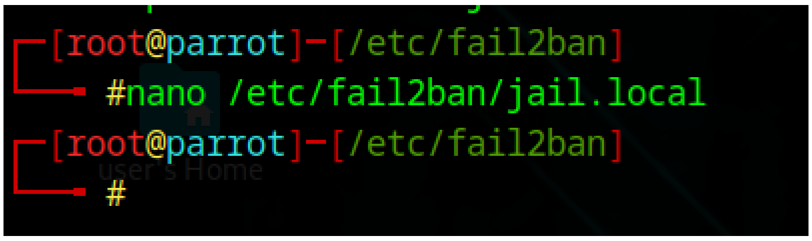
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local



```
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #
```

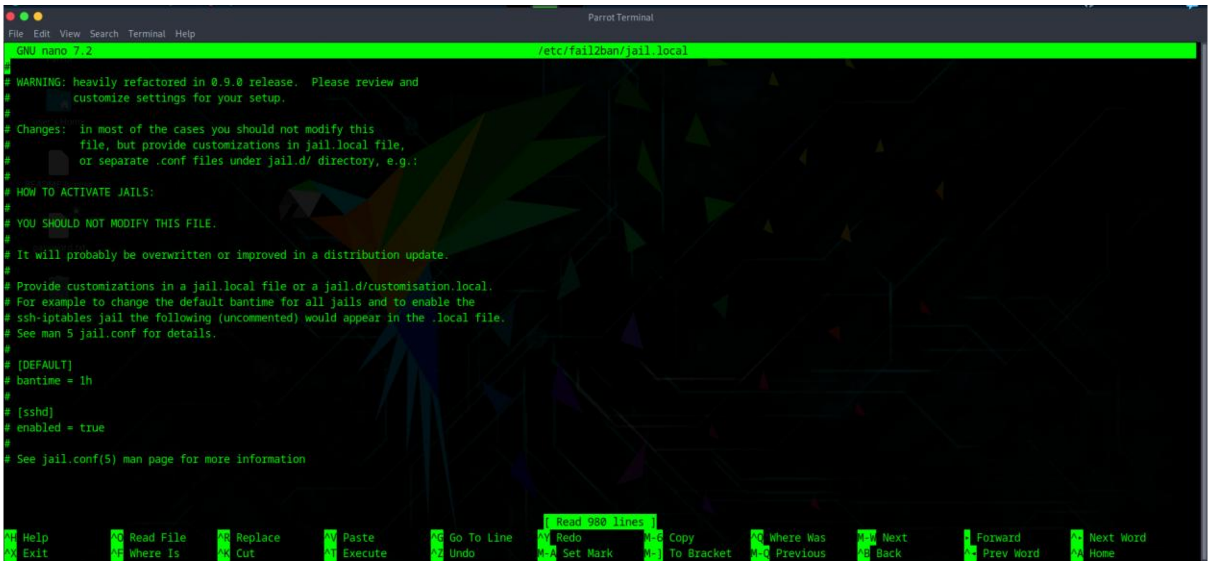
Use command to edit the jail.local file:

nano /etc/fail2ban/jail.local



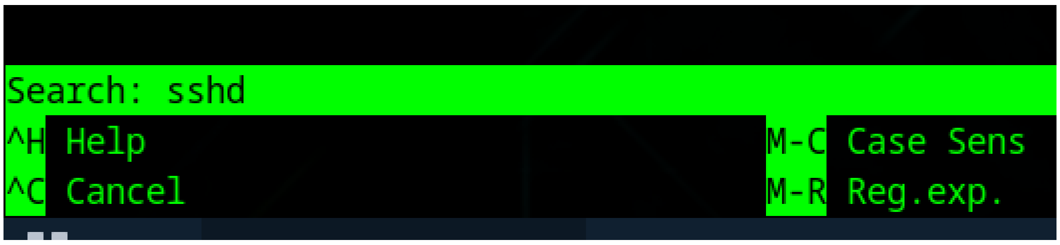
```
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #nano /etc/fail2ban/jail.local
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #
```

Following window will appear:



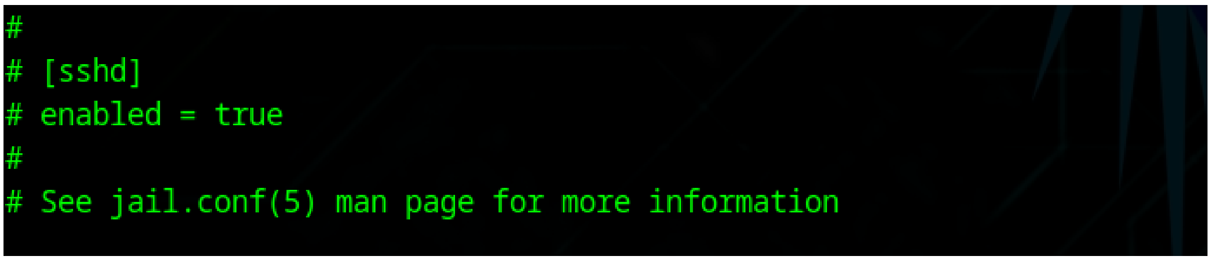
```
  GNU nano 7.2                                  /etc/fail2ban/jail.local
#
# WARNING: heavily refactored in 0.9.0 release.  Please review and
#          customize settings for your setup.
#
# Changes:  in most of the cases you should not modify this
#           file, but provide customizations in jail.local file,
#           or separate .conf files under jail.d/ directory, e.g.:
#
# HOW TO ACTIVATE JAILS:
#
# YOU SHOULD NOT MODIFY THIS FILE.
#
# It will probably be overwritten or improved in a distribution update.
#
# Provide customizations in a jail.local file or a jail.d/customisation.local.
# For example to change the default bantime for all jails and to enable the
# ssh-iptables jail the following (uncommented) would appear in the .local file.
# See man 5 jail.conf for details.
#
# [DEFAULT]
# bantime = 1h
#
# [sshd]
# enabled = true
#
# See jail.conf(5) man page for more information



                                        [ Read 980 lines ]
^H Help        ^O Read File   ^R Replace     ^V Paste       ^G Go To Line  ^Y Redo        M-6 Copy       ^Q Where Was   M-W Next       ^F Forward     ^→ Next Word
^X Exit        ^F Where Is    ^K Cut         ^T Execute     ^Z Undo        M-A Set Mark   M-] To Bracket M-Q Previous   ^B Back        ^← Prev Word   ^A Home
```

Look for the ‘sshd’ section:



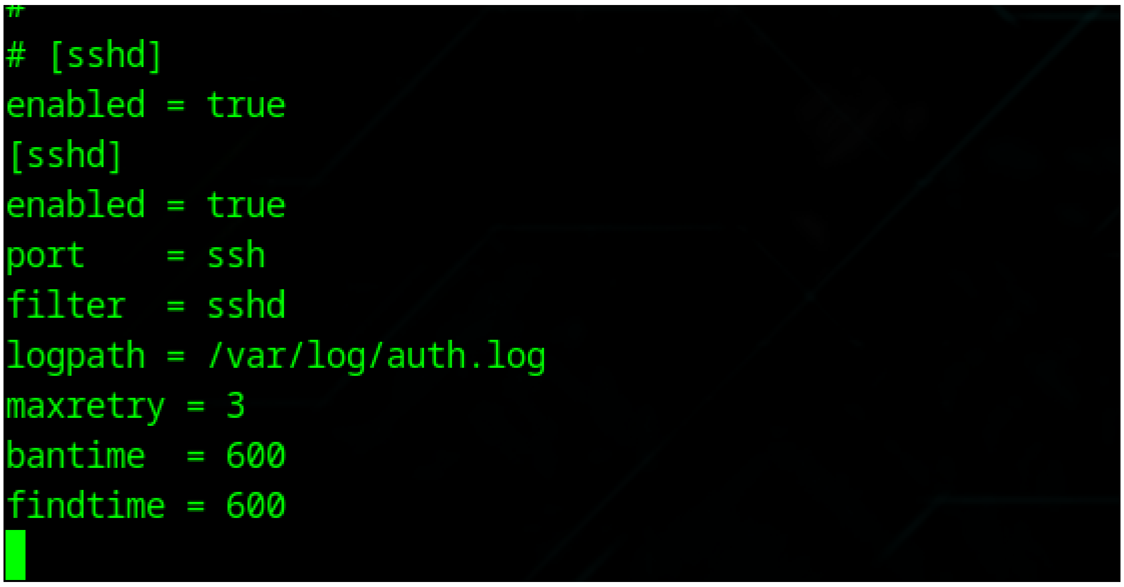
```
Search: sshd
^H Help                                M-C Case Sens
^C Cancel                              M-R Reg.exp.
```

Following section will appear:



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

Edit that section like this:



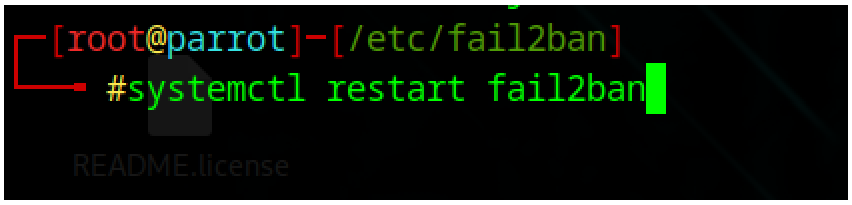
```
#
# [sshd]
enabled = true
[sshd]
enabled = true
port     = ssh
filter   = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime   = 600
findtime = 600
```

Meaning:

- **enabled** → turn on SSH jail
- **maxretry = 3** → after 3 failed attempts, ban
- **bantime = 600** → ban IP for 10 minutes
- **findtime = 600** → failed attempts counted within 10 min

Save the changes, and then restart the fail2ban using the following command:

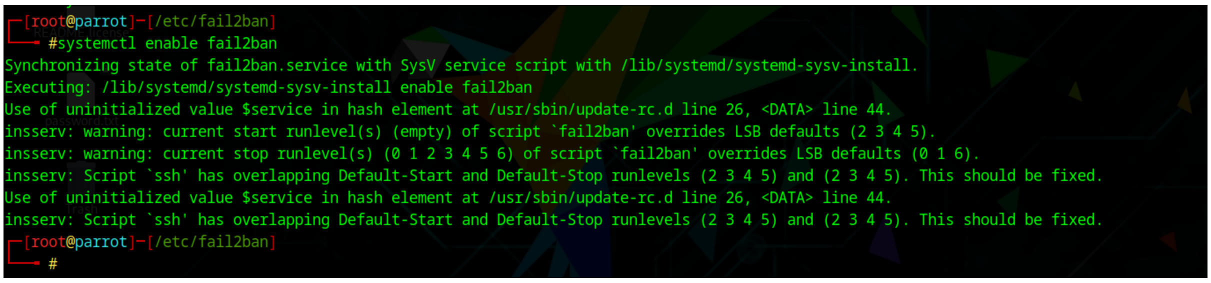
sudo systemctl restart fail2ban



```
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #systemctl restart fail2ban
```

Enable this on boot using the following command:

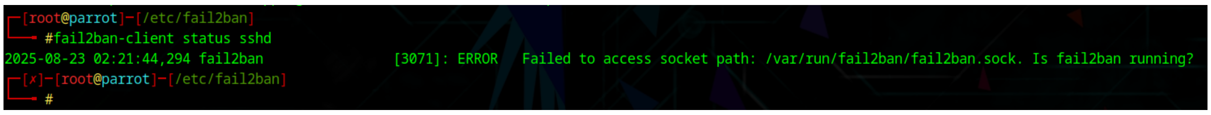
sudo systemctl enable fail2ban



```
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
insserv: warning: current start runlevel(s) (empty) of script `fail2ban' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `fail2ban' overrides LSB defaults (0 1 6).
insserv: Script `ssh' has overlapping Default-Start and Default-Stop runlevels (2 3 4 5) and (2 3 4 5). This should be fixed.
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
insserv: Script `ssh' has overlapping Default-Start and Default-Stop runlevels (2 3 4 5) and (2 3 4 5). This should be fixed.
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #
```

Verify it's working by checking status of SSH jail:

fail2ban-client status sshd



```
┌─[root@parrot]─[/etc/fail2ban]
└──╼ #fail2ban-client status sshd
2025-08-23 02:21:44,294 fail2ban                [3071]: ERROR   Failed to access socket path: /var/run/fail2ban/fail2ban.sock. Is fail2ban running?
┌─[✗]─[root@parrot]─[/etc/fail2ban]
└──╼ #
```

What happens in background?

Fail2ban works in the background by constantly monitoring log files, such as /var/log/auth.log for SSH, and looking for repeated failed login attempts using regex filters. When the number of failures exceeds the configured threshold within a set time window, Fail2ban automatically adds a temporary firewall rule (via iptables) to block the attacker's IP, either dropping their traffic silently or rejecting it. The ban lasts for the specified duration, after which Fail2ban removes the rule, restoring normal access. This way, it dynamically protects services by combining log monitoring with automatic firewall updates, without affecting legitimate users.

--The End--