NUST CYBER SECURITY TEAM 2019:

Server Hardening

1. Fail2ban

The fail2ban application monitors server log files for intrusion attempts and other suspicious activity. After a predefined number of failures from a host, fail2ban blocks its IP address automatically for a specific duration.

With fail2ban, you can help secure your server against unauthorized access attempts. It is particularly effective in reducing the risk from scripted attacks and botnets.

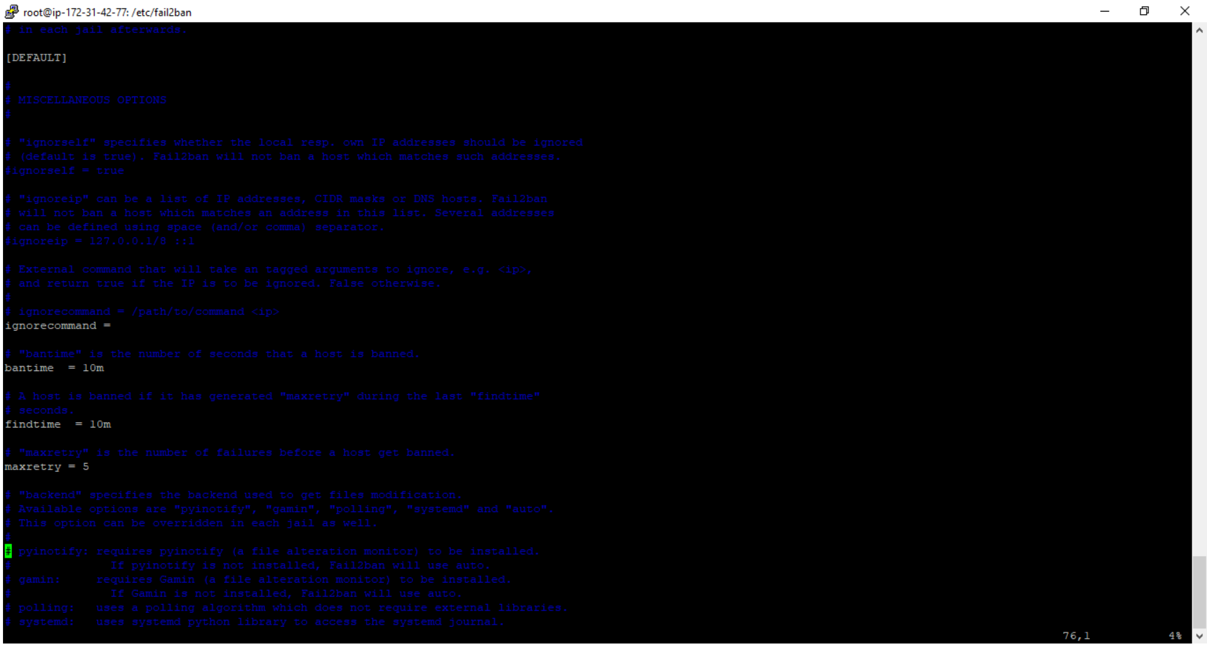
- To install the fail2ban package for your Linux distribution:
  #apt-get install fail2ban
  #yum install fail2ban

- Configuring fail2ban:
  #cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

The *jail.conf* file contains a basic configuration that you can use as a starting point, but it may be overwritten during updates. Fail2ban uses the separate *jail.local* file to actually read your configuration settings.

> Open the *jail.local* file in your preferred text editor.

> Locate the [**DEFAULT**] section, which contains the following global options:



root@ip-172-31-42-77: /etc/fail2ban

```
# in each jail afterwards.

[DEFAULT]

#
# MISCELLANEOUS OPTIONS
#

# "ignorself" specifies whether the local resp. own IP addresses should be ignored
# (default is true). Fail2ban will not ban a host which matches such addresses.
#ignorself = true

# "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban
# will not ban a host which matches an address in this list. Several addresses
# can be defined using space (and/or comma) separator.
#ignoreip = 127.0.0.1/8 ::1

# External command that will take an tagged arguments to ignore, e.g. <ip>,
# and return true if the IP is to be ignored. False otherwise.
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "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 = 5

# "backend" specifies the backend used to get files modification.
# Available options are "pyinotify", "gamin", "polling", "systemd" and "auto".
# This option can be overridden in each jail as well.
#
# pyinotify: requires pyinotify (a file alteration monitor) to be installed.
#              If pyinotify is not installed, Fail2ban will use auto.
# gamin:     requires Gamin (a file alteration monitor) to be installed.
#              If Gamin is not installed, Fail2ban will use auto.
# polling:   uses a polling algorithm which does not require external libraries.
# systemd:   uses systemd python library to access the systemd journal.
                                                              76,1          4%
```

ignoreip: This option enables you to specify IP addresses or hostnames that fail2ban will ignore. For example, you could add your home or office IP address so fail2ban does not prevent you from accessing your own server. To specify multiple addresses, separate them with a space.

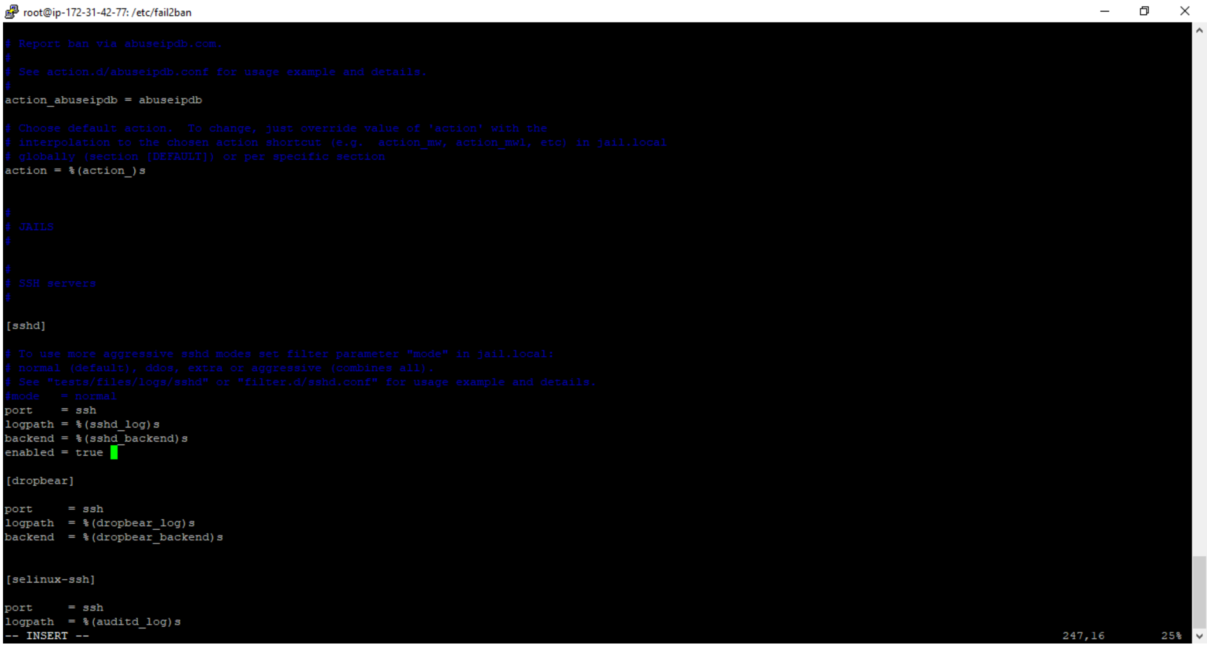
bantime: This option defines in seconds how long an IP address or host is banned. The default is 600 seconds (10 minutes).

maxretry: This option defines the number of failures a host is allowed before it is banned.

findtime: This option is used together with the **maxretry** option. If a host exceeds the **maxretry** setting within the time period specified by the **findtime** option, it is banned for the length of time specified by the **bantime** option.

With fail2ban's global options configured, you are now ready to enable and disable jails for the specific protocols and services you want to protect. By default, fail2ban monitors SSH login attempts (you can search for the [**ssh-iptables**] section in the *jail.local* file to view the specific settings for the SSH jail).

The *jail.local* file includes default jail settings for several protocols. Often, all you need to do to enable a jail is change its **enabled = false** line to **enabled = true** (add the enable line and restart fail2ban. You can also define custom jails and filters for additional flexibility



```
# Report ban via abuseipdb.com.
#
# See action.d/abuseipdb.conf for usage example and details.
#
action_abuseipdb = abuseipdb

# Choose default action.  To change, just override value of 'action' with the
# interpolation to the chosen action shortcut (e.g.  action_mw, action_mwl, etc) in jail.local
# globally (section [DEFAULT]) or per specific section
action = %(action_)s


#
# JAILS
#

#
# SSH servers
#

[sshd]

# 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
enabled = true

[dropbear]

port     = ssh
logpath  = %(dropbear_log)s
backend  = %(dropbear_backend)s


[selinux-ssh]

port     = ssh
logpath  = %(auditd_log)s
-- INSERT --                                                          247,16        25%
```

- Save your changes to the jail.local file.
- Restart the fail2ban service and load the new configuration

  #service fail2ban restart

  To display a list of IP addresses currently banned by fail2ban, type the following command:

  #iptables -S