

Securing Linux Server Using Honeypots and IP blocking

Nidhi Sharma

This can be achieved using these :

1. Deploying a Honeypot

- The honeypot is purely for research/logging attacks
- Host and VM, separated from production (testsite.mydomain.local on CentOS VM).
- SSH and vsftpd service deploy and expose for logging attacks.
- Ensure Isolation; honeypot not deployed on a sensitive or production network.
- Service ports (SSH on 2222) configured and made enticing for attackers.

2. Hardening SSH

- Change Default SSH port (from 22 to 2222).
- Configure users; Disable root logins.
- Password authentication presently enabled (to ease testing); plan for key-based authentication later.
- Update & Test SSH config; Validate known/unknown user access.

3. Automating IP blocking

- Enable EPEL repo; Install & configure Fail2ban
- Set Fail2ban jail for SSH. Verify it blocks after repeated failed attempts.
- Log Entries

4. Attack Simulation & Validation

- Use Kali or similar tools to run brute-force attacks against the honeypot.
- Document evidence of fail2ban bans, logged attacks, and honeypot alerts.

5. Log Analysis and Reporting

- Identify /var/log/secure location, command for failed attempt monitoring (grep "Failed password").
- Initial manual log parsing.
- Set up structured analytics (eg: Splunk, ELK, journald, etc.) and document incident samples.

6. Moving Forward: Future Enhancements

Problem Statement

Objective

To deploy a honeypot and configure defensive mechanisms like SSH hardening and automated IP blocking to detect, analyze, and mitigate SSH brute-force attacks on public-facing Linux servers

Problem Statement and Motivation

Real-time scenario:

SecureDefense, a cybersecurity firm managing Linux servers with public IPs, faces frequent SSH brute-force attacks that exploit weak credentials. To combat these threats, SecureDefense deploys honeypots to mimic real servers, capturing attack data to analyze patterns and attacker behaviors.

They enhance security by automating IP blocking using fail2ban, which monitors failed SSH login attempts and blocks attackers after repeated failures. Additionally, they harden SSH configurations by using non-standard ports and limiting login attempts to reduce attack surfaces.

This strategy not only protects client servers but also provides actionable insights to improve intrusion detection systems. By demonstrating proactive cybersecurity measures, SecureDefense strengthens its reputation as a trusted provider, attracting new enterprise clients.

Industry Relevance

The following tools used in this project serve specific purposes within the industry:

1. Honeypots (vsftpd, smbd, httpd, mysql): Decoy systems that mimic real servers to attract attackers, gather data on attack patterns, and improve intrusion detection systems
2. Fail2ban: Automates IP blocking after failed login attempts, effectively reducing brute-force attacks and maintaining server security
3. TCP Wrappers: Controls access by blocking or allowing specific IPs, adding an extra security layer to Linux servers
4. Firewalld: Manages dynamic firewall rules, restricting access to sensitive ports and mitigating brute-force attacks
5. Splunk: Analyzes server logs to identify attack trends and improve threat detection and response
6. OpenSSH: Secures remote access with custom configurations like nonstandard ports and login attempt limits to reduce vulnerabilities

Table of Contents

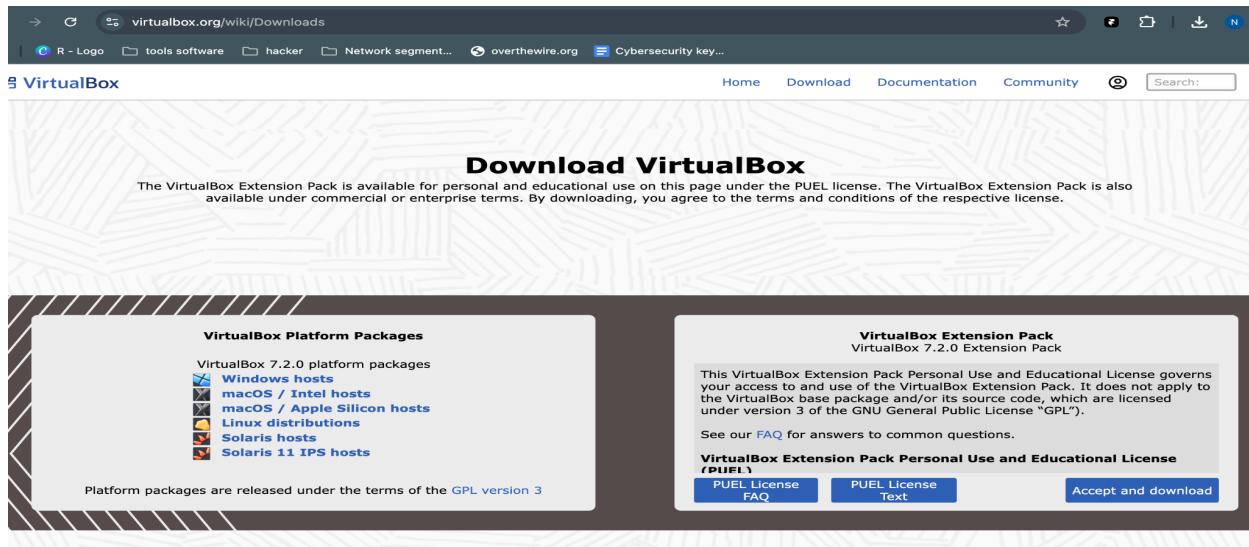
Problem Statement.....	3
Objective.....	3
Problem Statement and Motivation.....	3
Real-time scenario:.....	3
Industry Relevance.....	3
Table of Contents.....	5
Initial Setup of Virtual Machines in VirtualBox.....	7
Honeypot Deployment.....	10
Step 1: Simulate Real Server Environment - Key Installations.....	10
Step 2: Add Script to the httpd config File.....	11
Step 3: Setup and Host the Website on the Server.....	12
Harden SSH Configuration to Secure the Server.....	15
Step 1: Install OpenSSH server.....	15
Step 2: Edit SSH Configuration.....	15
Step 3: Set SELinux to allow new port (Port 2222).....	16
Step 4: Restart SSH to apply changes.....	16
Step 5: Configure Firewall to Allow New Port.....	17
Step 6: Test SSH Hardening.....	18
Create Credible Users on CentOS.....	18
Edit SSH Config to Allow Only These Users.....	18
Restart SSH Service.....	19
Test Allowed Users Can Login.....	20
Test Unknown User.....	22
Check Logs for The Failed Authentication.....	22
Automated IP Blocking.....	24
Step 1: Setup Fail2Ban IPS to Block Repeated Failures.....	24
Enable epel-release and then install fail2ban.....	24
Create / edit jail.local.....	24
Start and enable the fail2ban.....	25
Step 2: Install Firewalld.....	26
Step 3: Configure Fail2Ban to Use Firewallcmd for Active Banning..	27
Testing: Manual Attack.....	31
Step 1: Setup.....	31
Step 2: Attempt Brute force Attack.....	33
Penetration Testing Through Proxchains and Tor.....	35

Step 1: Configure Tor on Kali Linux.....	35
Step 2: Configure Proxychains to Use Tor.....	36
Test Proxychains with Tor.....	37
Step 3: Prepare Tools and Wordlists.....	37
Extract Wordlists.....	37
Prepare Username List.....	38
Step 4: Attack.....	38
Construct Your Hydra Command.....	38
Step 5: Analyse Logs Manually.....	39
Automating Log Analysis and Reporting.....	42
Step 1: Define Your Log Files and Data Points.....	42
Step 2: Write Basic Bash Script for Daily Report.....	42
Step 3: Make Script Executable.....	43
Step 4: Run and Test the Script.....	43
Step 5: Let's Automate the Script Using Cron For Daily or Weekly reporting.....	44
How to Schedule the Script with Cron.....	44
Add a Cron Schedule Line.....	44
Verification.....	45
Integrating all logs with SIEM Tools for Better Log Analysis.....	46
Step 1: Install Splunk from Official Website.....	46
Step 2: Start and enable Splunk service.....	47
Step 3: Configure Data Inputs for Critical Logs.....	48
What Happens After This?.....	51
Future Enhancements: Moving Forward.....	53
Deploy Advanced Honeypots:.....	53
Strengthen Authentication:.....	53
Centralize Log Management:.....	53
Automate Threat Intelligence Integration:.....	53
Streamline Incident Response:.....	54
Continuous Security Testing and Defense Tuning:.....	54
References.....	55

Initial Setup of Virtual Machines in VirtualBox

Note: This setup is required only for local machines you can do complete activity in the virtual lab without this setup (especially if CentOS linux is already in the lab)

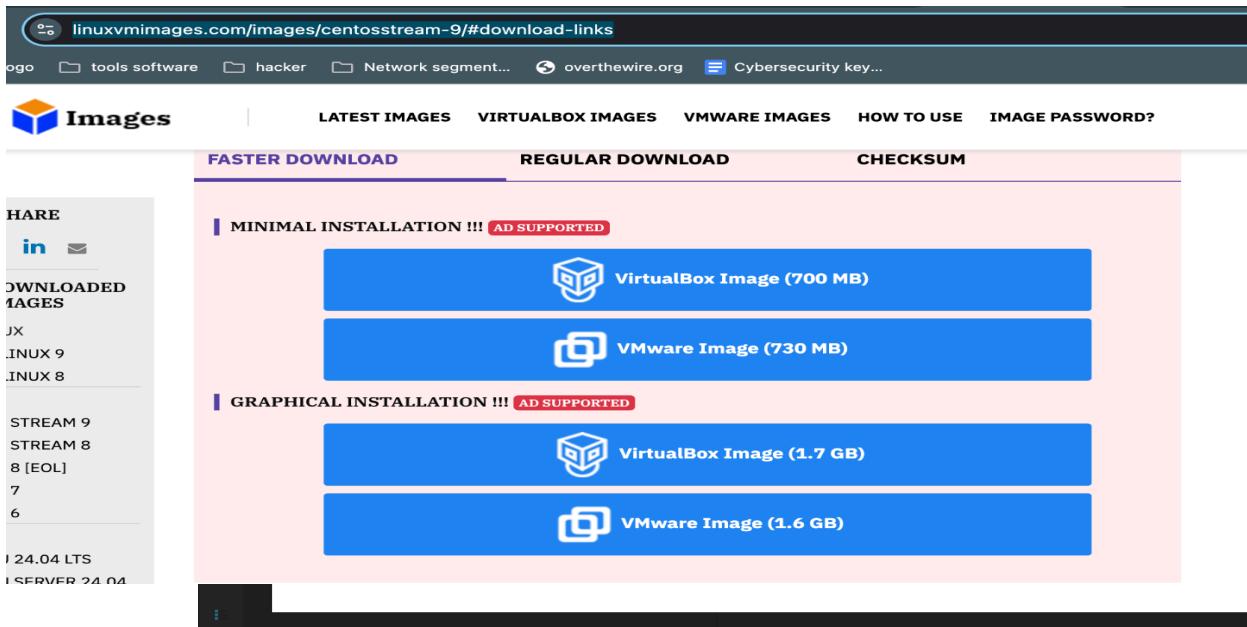
- Install VirtualBox from [Oracle Virtual Box](#) downloads page.



Select the package depending on your machine configuration.

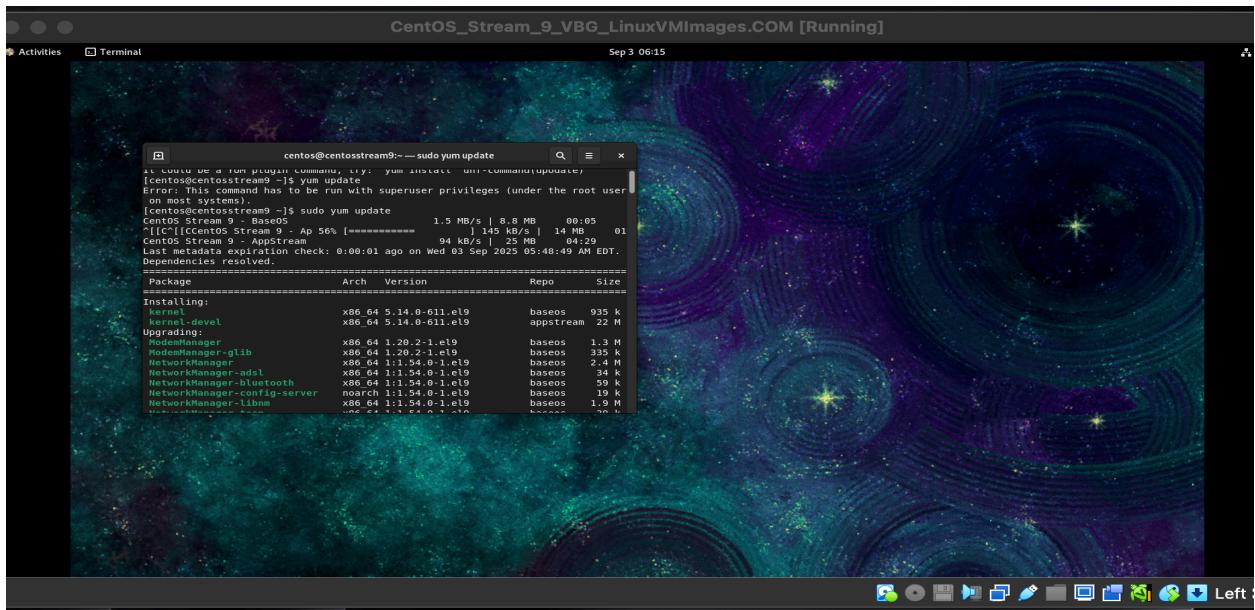
- Install CentOS linux VM
 - Download centos Linux VM from [this page](#). Opt for the Graphical Installation VirtualBox image (If you have VMware then use VMware Image)
 - Uncompress the downloaded file.
 - Open the VirtualBox and click on the + (Open/Add) sign.

- Then you can browse to the folder containing the file



CentOS_Stream_9_VBG_LinuxVMImages.COM.vbox (if you are using virtual box)

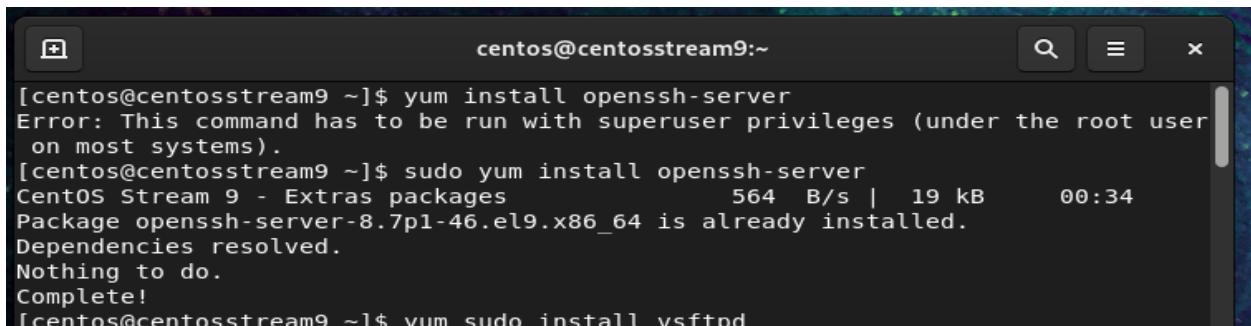
- Once Added → select the VM then click on Start (the green right arrow)
- Enter password (centos → common for all)
- Next click on activities → terminal → type command `sudo yum update`
- Whenever you are asked "do you want to install packages" type `y` and press enter
- Wait till all packages are installed



Honeypot Deployment

Step 1: Simulate Real Server Environment - Key Installations

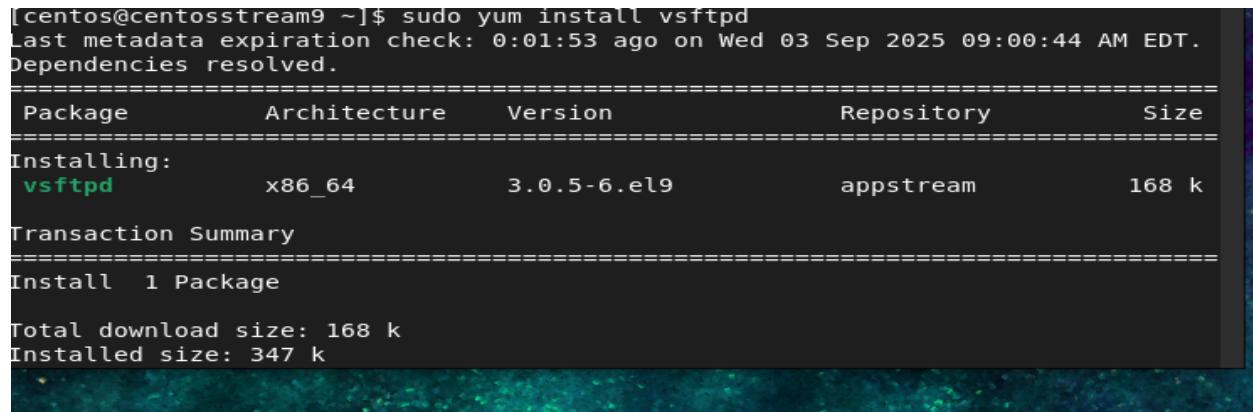
1. Install OpenSSH server Use `sudo yum install openssh-server`



A screenshot of a terminal window titled "centos@centosstream9:~". The terminal shows the following command and its output:

```
[centos@centosstream9 ~]$ yum install openssh-server
Error: This command has to be run with superuser privileges (under the root user
on most systems).
[centos@centosstream9 ~]$ sudo yum install openssh-server
CentOS Stream 9 - Extras packages      564 B/s | 19 kB     00:34
Package openssh-server-8.7p1-46.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[centos@centosstream9 ~]$ yum sudo install vsftpd
```

command on the terminal



A screenshot of a terminal window titled "centos@centosstream9:~". The terminal shows the following command and its output:

```
[centos@centosstream9 ~]$ sudo yum install vsftpd
Last metadata expiration check: 0:01:53 ago on Wed 03 Sep 2025 09:00:44 AM EDT.
Dependencies resolved.
=====
 Package           Architecture   Version        Repository      Size
 =====
 Installing:
  vsftpd          x86_64        3.0.5-6.el9    appstream    168 k
 Transaction Summary
 =====
 Install 1 Package

 Total download size: 168 k
 Installed size: 347 k
```

2. Install vsftpd. Use `sudo yum install vsftpd` command on the terminal

- Create the directory /reports using `mkdir reports` command on the terminal
- We install an Apache HTTP server to mimic the real environment. We use the command `sudo yum install httpd`

Step 2: Add Script to the httpd config File

- This is the Script:

```
[centos@centosstream9 ~]$ cd /etc/httpd/conf.d  
[centos@centosstream9 conf.d]$
```

- Navigate to the config file `cd /etc/httpd/conf.d`
 - Open vi editor using command `vi testsite.mydomain.local.conf`
I had to use sudo since editing the file required administrative permission. Alternatively to view the contents of the file you can use `cat` command and to edit the file you can use `nano`.

```
[centos@centosstream9 conf.d]$ sudo vi testsite.mydomain.local.conf
```

- Press `i` to enter insert mode and paste the script in first step

```

CentOS_Stream_9_VBG_LinuxVMImages.COM [Running]
[centos@centosstream9 conf.d]$ sudo nano testsite.mydomain.local.conf
[centos@centosstream9 conf.d]$ cat testsite.mydomain.local.conf
#domain=testsite.mydomain.local
<Directory "/var/www/testsite.mydomain.local">
    Options +Indexes
    AllowOverride all
    Require all granted
</Directory>
<VirtualHost *:80>
    DocumentRoot "/var/www/testsite.mydomain.local"
    ServerName "testsite.mydomain.local"
    ServerAdmin "webmaster@mydomain.local.in"
    ErrorLog "logs/testsite.mydomain.local_error_log"
    CustomLog "logs/testsite.mydomain.local_access_log" Combined
</VirtualHost>

[centos@centosstream9 conf.d]$ sudo systemctl start httpd
[centos@centosstream9 conf.d]$ █

```

Step 3: Setup and Host the Website on the Server

- Navigate to the directory using the command `cd /var/www`
- Create a new directory inside www using the command (might need to use sudo to elevate permission to create directory at this level): `mkdir testsite.mydomain.local`
- Change the directory using the command: `cd testsite.mydomain.local`
- Now create file and insert text in `index.html` using command (might need to use sudo to elevate permission to create directory at this level): `vi index.html`

```

centos@centosstream9:/var/www/testsite.mydomain.local
[centos@centosstream9 ~]$ cd /etc/httpd/conf.d
[centos@centosstream9 conf.d]$ vi testsite.mydomain.local.conf
[centos@centosstream9 conf.d]$ sudo vi testsite.mydomain.local.conf
[centos@centosstream9 conf.d]$ cd /var/www
[centos@centosstream9 www]$ mkdir testsite.mydomain.local
mkdir: cannot create directory 'testsite.mydomain.local': Permission denied
[centos@centosstream9 www]$ sudo mkdir testsite.mydomain.local
[centos@centosstream9 www]$ cd testsite.mydomain.local
[centos@centosstream9 testsite.mydomain.local]$ vi index.html
[centos@centosstream9 testsite.mydomain.local]$ sudo vi index.html
[centos@centosstream9 testsite.mydomain.local]$ █

```

- Use command `sudo ln -s /reports/` to create symbolic link ([more info](#))

```
Complete!
[centos@centosstream9 testsite.mydomain.local]$ ls
index.html reports
[centos@centosstream9 testsite.mydomain.local]$ sudo systemctl start httpd
[centos@centosstream9 testsite.mydomain.local]$ █
```

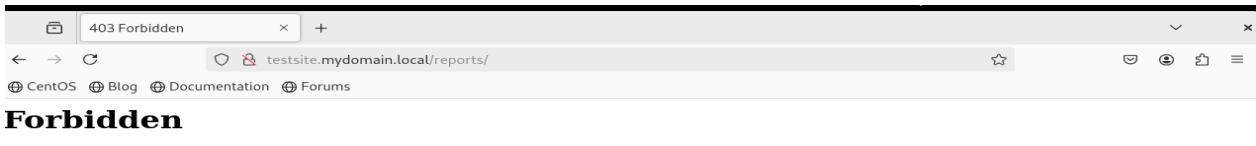
- Start the Apache HTTP server using command `systemctl start httpd`
(more [info](#))

```
index.html reports
[centos@centosstream9 testsite.mydomain.local]$ sudo systemctl start httpd
[centos@centosstream9 testsite.mydomain.local]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/
httpd.service.
[centos@centosstream9 testsite.mydomain.local]$ █
```

- Next start enable the httpd with command `sudo systemctl enable httpd`

```
index.html reports
[centos@centosstream9 testsite.mydomain.local]$ sudo systemctl start httpd
[centos@centosstream9 testsite.mydomain.local]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/
httpd.service.
[centos@centosstream9 testsite.mydomain.local]$ firefox http://testsite.mydomain.local$ █
```

- Open website using the command `firefox testsite.mydomain.local`
- Navigate to the <http://testsite.mydomain.local/reports>



[troubleshooting]. We get “Forbidden”

- Now run the command to allow the reports to be accessed by browser

```
chcon unconfined_u:object_r:httpd_sys_content_t:s0 /reports/
```

```
[centos@centosstream9 testsite.mydomain.local]$ sudo chcon unconfined_u:object_r:httpd_sys_content_t:s0 /reports/
[centos@centosstream9 testsite.mydomain.local]$ █
```

- Now start firefox again and open /reports

Name	Last modified	Size	Description
Parent Directory	-	-	

```
[centos@centosstream9 www]$ cd testsite.mydomain.local/
[centos@centosstream9 testsite.mydomain.local]$ touch /reports/101.log
touch: cannot touch '/reports/101.log': Permission denied
[centos@centosstream9 testsite.mydomain.local]$ sudo touch /reports/101.log
[centos@centosstream9 testsite.mydomain.local]$ █
```

- Now create 101.log file and access it using browser

Name	Last modified	Size	Description
Parent Directory	-	-	
101.log	2025-09-04 02:56	0	

Harden SSH Configuration to Secure the Server

Lets start hardening SSH on our CentOS Server to secure it against brute force attacks and unauthorised SSH access.

Step 1: Install OpenSSH server

This we had installed already

Step 2: Edit SSH Configuration

- Check current configuration by opening the file

```
sudo cat /etc/ssh/sshd_config
```

Make the following changes for better security:

- Change default SSH port (e.g., from 22 to 2222)
 Port 2222
- Disable root login via SSH:
 PermitRootLogin no
- Disable password authentication (require key-based auth):
 PasswordAuthentication no
- Allow only specific users (optional):
 AllowUsers yourusername
- Limit login attempts:
 MaxAuthTries 3
- Disable empty passwords:
 PermitEmptyPasswords no

```

# GNU nano 5.6.1
/etc/
# /etc/ssh/sshd_config.d/, which will be automatically included
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUM
#
Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:
#LoginGraceTime 2m
#PermitRootLogin no
#StrictModes yes
MaxAuthTries 3
#MaxSessions 10

GNU nano 5.6.1
/etc/ssh/sshd_config
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:
#LoginGraceTime 2m
#PermitRootLogin no
#StrictModes yes
MaxAuthTries 3
#MaxSessions 10

^G Help      ^O Write Out   ^W Where Is   ^K Cut
^X Exit      ^R Read File   ^L Replace   ^U Paste

```

Step 3: Set SELinux to allow new port (Port 2222)

- Check current SELinux SSH port labels:

```
sudo semanage port -l
```

SELinux Port Type	Proto	Port Number
afs3_callback_port_t	tcp	7001
afs3_callback_port_t	udp	7001
afs_bos_port_t	udp	7007
afs_fs_port_t	tcp	2040
afs_fs_port_t	udp	7000, 7005
afs_ka_port_t	udp	7004
afs_pt_port_t	tcp	7002
afs_pt_port_t	udp	7002
afs_vl_port_t	udp	7003
agentx_port_t	tcp	705

- Add a new port to SELinux:

```
sudo semanage port -a -t ssh_port_t -p tcp 2222
```

We checked whether the port is added by using `sudo semanage port -l`

Step 4: Restart SSH to apply changes

```
sudo systemctl restart sshd
sudo systemctl status sshd
```

soundd_port_t	tcp	8000, 9433, 16001
spamd_port_t	tcp	783, 10026, 10027
speech_port_t	tcp	8036
squid_port_t	tcp	3128, 3401, 4827
squid_port_t	udp	3401, 4827
ssdp_port_t	tcp	1900
ssdp_port_t	udp	1900
ssh_port_t	tcp	2222, 22
statsd_port_t	udp	8125
svn_port_t	tcp	3690
svn_port_t	udp	3690
svrloc_port_t	tcp	427

The status should be active(running)

```
centos@centosstream9:~$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; v
   Active: active (running) since Thu 2025-09-04 05:02:09 EDT; 9s
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 23654 (sshd)
      Tasks: 1 (limit: 10516)
     Memory: 2.1M (peak: 2.5M)
        CPU: 14ms
       CGroup: /system.slice/sshd.service
               └─23654 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-10>

Sep 04 05:02:09 centosstream9.linuxvmimages.local systemd[1]: Start>
```

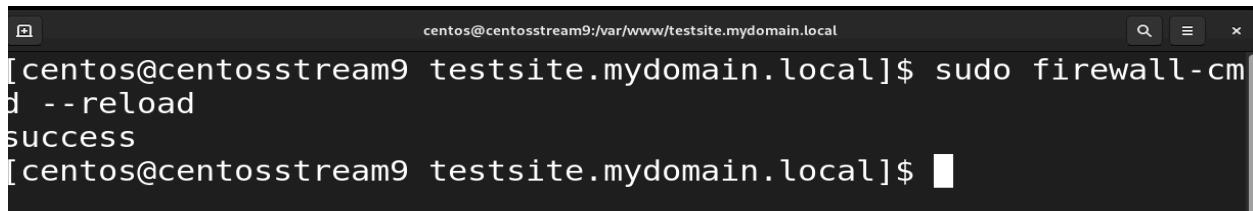
Step 5: Configure Firewall to Allow New Port

- Open new port via firewall:

```
sudo firewall-cmd --permanent --zone=public --add-port=2222/tcp
```

```
[centos@centosstream9 testsite.mydomain.local]$ sudo firewall-cm
d --permanent --zone=public --add-port=2222/tcp
success
[centos@centosstream9 testsite.mydomain.local]$
```

```
sudo firewall-cmd --reload
```



```
centos@centosstream9:~$ sudo firewall-cmd --reload
success
centos@centosstream9:~$
```

- Remove old port (optional - but best for good security): If you wish you can close this or keep it open for your Honeypot to be attractive

```
sudo firewall-cmd --permanent --zone=public --remove-port=22/tcp
```

```
sudo firewall-cmd --reload
```

Step 6: Test SSH Hardening

Create Credible Users on CentOS

- To test the SSH Hardening lets first create some users - some who will be granted access in the config file and some can be denied access.

```
sudo adduser user1
sudo passwd user1 # Set password when prompted
```

```
sudo adduser user2
sudo passwd user2
```

Edit SSH Config to Allow Only These Users

- Open ssh configuration file

```
[centos@centosstream9 ~]$ sudo adduser user1
[centos@centosstream9 ~]$ sudo passwd user1
Changing password for user user1.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
[centos@centosstream9 ~]$ sudo passwd user1
Changing password for user user1.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[centos@centosstream9 ~]$ sudo adduser user2
[centos@centosstream9 ~]$ sudo passwd user2
Changing password for user user2.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[centos@centosstream9 ~]$ █
```

```
sudo nano /etc/ssh/sshd_config
```

- Add the code

```
AllowUsers user1 user2
```

```
# and KbdInteractiveAuthentication
# WARNING: 'UsePAM no' is not supported
# problems.
#UsePAM no
AllowUsers user1 user2
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding yes
#PubkeyAuthentication yes
```

Restart SSH Service

```
sudo systemctl restart sshd
```

```
[centos@centosstream9 ~]$ sudo systemctl restart sshd  
[centos@centosstream9 ~]$
```

Test Allowed Users Can Login

- Now try to access using the command
`ssh user1@testsite.mydomain.local -p 2222`

```
[centos@centosstream9 ~]$ ssh user1@testsite.mydomain.local -p 2222
The authenticity of host '[testsite.mydomain.local]:2222 ([127.0.0.1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:CwCMdUrZadeJ8s3ihT4rws05cwAOSUJ0z8DTHPmcW8U.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[testsite.mydomain.local]:2222' (ED25519) to the list of known hosts.
+-----+
          LINUXVMIMAGES.COM
+-----+
          User Name: centos
          Password: centos (sudo su -)
user1@testsite.mydomain.local: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[centos@centosstream9 ~]$ █
```

- But, the above won't work since we have disabled password based authentication.
 - We can enable the PasswordAuthentication

```
#IgnoreRhosts yes  
  
# To disable tunneled clear text passwords, change to no here!  
#PasswordAuthentication no  
PermitEmptyPasswords no  
  
[ Wrote 131 lines ]  
  
^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo  
^X Exit      ^R Read File   ^\ Replace    ^U Paste      ^J Justify    ^- Go To Line M-E Redo
```

Comment out PasswordAuthentication no using #

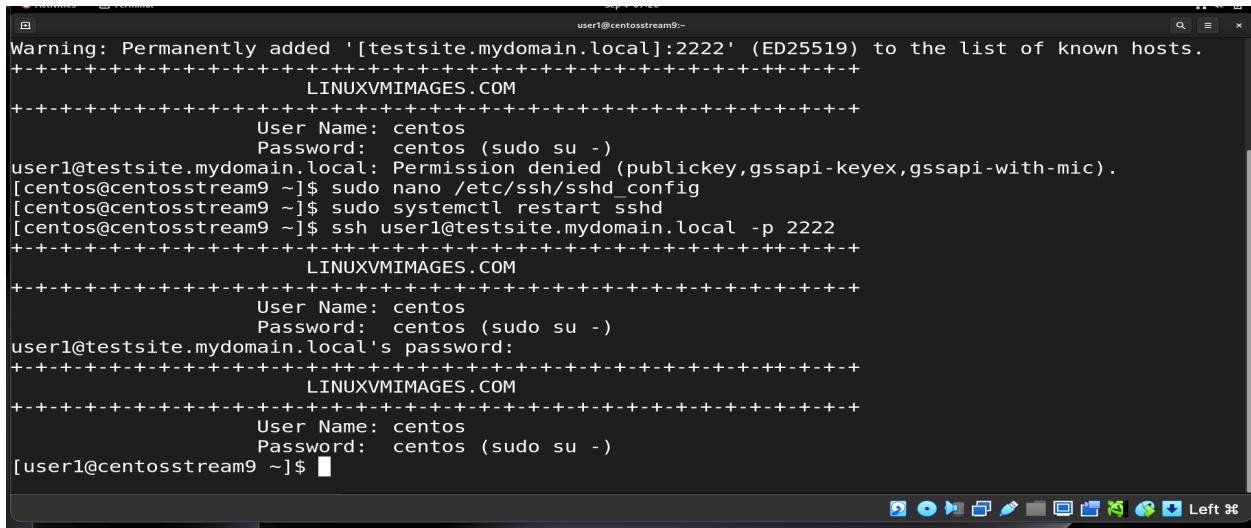
```
[centos@centosstream9 ~]$ sudo systemctl restart sshd  
[centos@centosstream9 ~]$ █
```

- Now restart the ssh service again

- Then try again with same command

```
ssh user1@testsite.mydomain.local -p 2222
```

It works.



The screenshot shows a terminal window titled 'user1@centosstream9:~'. The session starts with a warning message about host key fingerprinting. It then shows the user attempting to log in via SSH with the command 'ssh user1@testsite.mydomain.local -p 2222'. The server responds with a 'Permission denied' message, indicating that password authentication is disabled. The user then tries to log in again, this time providing a password ('centos (sudo su -)'). The server rejects the password attempt, stating 'user1@testsite.mydomain.local's password:'. Finally, the user attempts to log in again with the password, which is rejected again. The terminal window has a dark background and light-colored text. At the bottom, there is a standard Linux desktop taskbar with icons for file operations and system status.

```
Warning: Permanently added '[testsite.mydomain.local]:2222' (ED25519) to the list of known hosts.
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      LINUXVMIMAGES.COM
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      User Name: centos
      Password: centos (sudo su -)
user1@testsite.mydomain.local: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[centos@centosstream9 ~]$ sudo nano /etc/ssh/sshd_config
[centos@centosstream9 ~]$ sudo systemctl restart sshd
[centos@centosstream9 ~]$ ssh user1@testsite.mydomain.local -p 2222
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      LINUXVMIMAGES.COM
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      User Name: centos
      Password: centos (sudo su -)
user1@testsite.mydomain.local's password:
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      LINUXVMIMAGES.COM
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      User Name: centos
      Password: centos (sudo su -)
[centos@centosstream9 ~]$
```

If not then we would have to setup key based authentication which would be beyond the scope of this project

Enabling password-based authentication temporarily lets us test and manage the honeypot environment quickly without the complexity of SSH key setup.

Points to note:

- Password authentication is less secure than key-based authentication, so this is a temporary measure.
- Lets use strong, unique passwords for your users.
- Keep fail2ban and IP blocking scripts active to help mitigate brute-force attacks while password authentication is enabled.
- Plan to switch to key-based authentication later when we scope the honeypot project or move to a production-like environment for higher security.

- When ready, we can disable password login again (PasswordAuthentication no) and configure keys securely.
- This approach balances flexibility in development and testing with eventual best practices for hardening.

Test Unknown User

- Before adding new user
 - Exit from current logged in user
 - Use command `exit`
- Add a new user


```
sudo adduser userNotAllowed
sudo passwd userNotAllowed
```
- Next try to access `testsite.mydomain.local`

```
ssh unknownuser@your_server_ip -p 2222
```
- All above commands are in the screenshot below:

The screenshot shows a terminal window on a Linux desktop environment. The terminal output is as follows:

```
[sudo] password for user1:
user1 is not in the sudoers file. This incident will be reported.
[user1@centosstream9 ~]$ exit
logout
Connection to testsite.mydomain.local closed.
[centos@centosstream9 ~]$ sudo adduser userNotAllowed
[centos@centosstream9 ~]$ sudo passwd userNotAllowed
Changing password for user userNotAllowed.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[centos@centosstream9 ~]$ ssh userNotAllowed@testsite.mydomain.local -p 2222
+-----+
          LINUXVMIMAGES.COM
+-----+
          User Name: centos
          Password: centos (sudo su -)
userNotAllowed@testsite.mydomain.local's password:
Permission denied, please try again.
userNotAllowed@testsite.mydomain.local's password: █
```

The terminal window has a dark background with light-colored text. At the bottom, there is a standard Linux desktop taskbar with icons for various applications like a browser, file manager, and system tools.

The message says Permission Denied on the terminal.

Check Logs for The Failed Authentication

```
sudo cat /var/log/secure
```

```
or user root
Sep  4 07:52:14 centosstream9 sshd[4709]: User userNotAllowed from 127.0.0.1 not allowed because not listed in AllowUsers
Sep  4 07:52:19 centosstream9 unix_chkpwd[4713]: password check failed for user (userNotAllowed)
Sep  4 07:52:19 centosstream9 sshd[4709]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=127.0.0.1 user=userNotAllowed
Sep  4 07:52:21 centosstream9 sshd[4709]: Failed password for invalid user userNotAllowed from 127.0.0.1 port 51274 ssh2
Sep  4 08:05:42 centosstream9 gdm-password][4741]: gkr-pam: unlocked login keyring
Sep  4 08:06:12 centosstream9 sudo[4775]: centos : TTY=pts/0 ; PWD=/home/centos ; USER=root ; COMMAND=/bin/cat /var/log/secure
Sep  4 08:06:12 centosstream9 sudo[4775]: pam_unix(sudo:session): session opened for user root(uid=0) by centos(uid=1000)
[centos@centosstream9 ~]$ █
```

Automated IP Blocking

Step 1: Setup Fail2Ban IPS to Block Repeated Failures

Enable epel-release and then install fail2ban

```
sudo yum install epel-release
```

```
[centos@centosstream9 testsite.mydomain.local]$ sudo yum install epel-release
Last metadata expiration check: 0:41:50 ago on Thu 04 Sep 2025 05:10:22 AM EDT.
Dependencies resolved.
=====
 Package           Arch      Version       Repository      Size
 =====
 Installing:
 epel-release     noarch    9-7.el9      extras-common   19 k
 Installing weak dependencies:
 epel-next-release noarch    9-7.el9      extras-common   8.1 k
 Transaction Summary
 =====
 Install 2 Packages

Total download size: 27 k
```

```
sudo yum install fail2ban
```

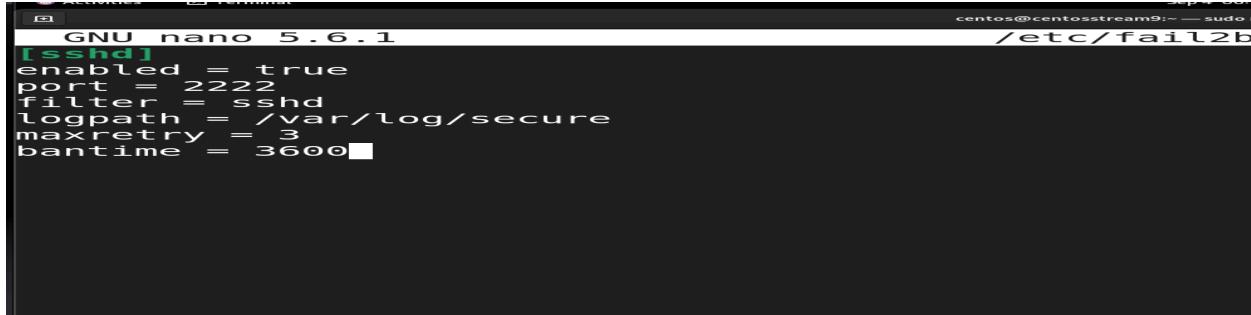
Create / edit jail.local

```
sudo nano /etc/fail2ban/jail.local
```

```
Complete!
[centos@centosstream9 ~]$ sudo systemctl start fail2ban
[centos@centosstream9 ~]$ sudo systemctl enable fail2ban
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /usr/lib/systemd/system/fail2ban.service.
[centos@centosstream9 ~]$ █
```

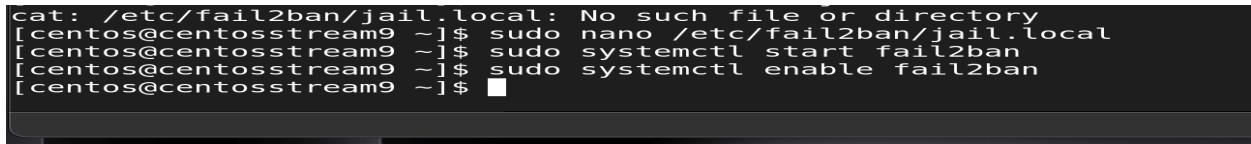
Add the code like below in the editor:

```
[sshd]
enabled = true
port = 2222
filter = sshd
logpath = /var/log/secure
maxretry = 3
bantime = 3600
```



The screenshot shows a terminal window titled "GNU nano 5.6.1" with the file path "/etc/fail2ban/filter.d/sshd.conf". The content of the file is identical to the one shown above, defining a filter for the sshd service.

Save the file using `^o` and then enter. Exit using `^x`

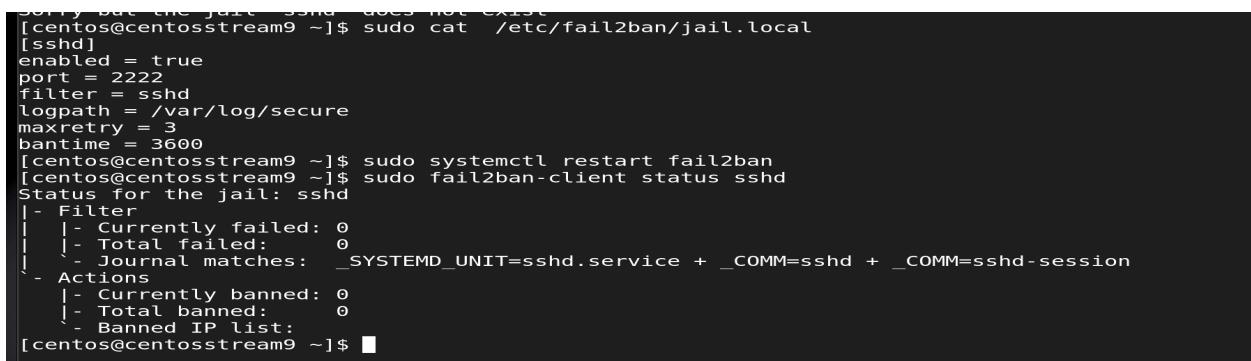


The screenshot shows a terminal window with the following commands being run:

```
cat: /etc/fail2ban/jail.local: No such file or directory
[centos@centosstream9 ~]$ sudo nano /etc/fail2ban/jail.local
[centos@centosstream9 ~]$ sudo systemctl start fail2ban
[centos@centosstream9 ~]$ sudo systemctl enable fail2ban
[centos@centosstream9 ~]$ █
```

Start and enable the fail2ban

Verify jail status
`sudo fail2ban-client status sshd`



The screenshot shows a terminal window with the following command being run:

```
sudo fail2ban-client status sshd
[centos@centosstream9 ~]$ sudo cat /etc/fail2ban/jail.local
[sshd]
enabled = true
port = 2222
filter = sshd
logpath = /var/log/secure
maxretry = 3
bantime = 3600
[centos@centosstream9 ~]$ sudo systemctl restart fail2ban
[centos@centosstream9 ~]$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|   |- Currently failed: 0
|   |- Total failed: 0
|   |- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd + _COMM=sshd-session
- Actions
  |- Currently banned: 0
  |- Total banned: 0
  |- Banned IP list:
[centos@centosstream9 ~]$ █
```

Step 2: Install Firewalld

Run command

```
sudo yum install firewalld -y
```

```
UNIT firewalld.service could not be found.  
[centos@centosstream9 ~]$ sudo yum install firewalld -y  
Last metadata expiration check: 4:16:53 ago on Fri 05 Sep 2025 06:52:28 AM EDT.  
Package firewalld-1.3.4-15.el9.noarch is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!
```

Enable the firewalld and start it

```
sudo systemctl enable firewalld  
sudo systemctl start firewalld
```

```
[centos@centosstream9 ~]$ sudo systemctl enable firewalld  
[centos@centosstream9 ~]$ sudo systemctl start firewalld  
[centos@centosstream9 ~]$
```

Check Status of firewalld

```
sudo systemctl status firewalld
```

```
[centos@centosstream9 ~]$ sudo systemctl start firewalld  
[centos@centosstream9 ~]$ sudo systemctl status firewalld  
● firewalld.service - firewalld - dynamic firewall daemon  
    Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: en  
    Active: active (running) since Fri 2025-09-05 11:06:07 EDT; 11min ago  
      Docs: man:firewalld(1)  
    Main PID: 830 (firewalld)  
       Tasks: 2 (limit: 10510)  
     Memory: 40.4M (peak: 65.7M)  
        CPU: 1.596s  
      CGroup: /system.slice/firewalld.service  
              └─830 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nrepid  
  
Sep 05 11:06:05 centosstream9.linuxvmimages.local systemd[1]: Starting firewalld ->  
Sep 05 11:06:07 centosstream9.linuxvmimages.local systemd[1]: Started firewalld ->  
lines 1-13/13 (END)
```

It must be active (running)

Step 3: Configure Fail2Ban to Use Firewallcmd for Active Banning

- Set `action = firewallcmd-ipset` in `/etc/fail2ban/jail.local`

```
[centos@centosstream9 ~]$ sudo nano /etc/fail2ban/jail.local  
[centos@centosstream9 ~]$ █
```

```
[centos@centosstream9 ~]$ sudo nano /etc/fail2ban/jail.local  
[centos@centosstream9 ~]$ cat /etc/fail2ban/jail.local  
[sshd]  
enabled = true  
port = 2222  
filter = sshd  
logpath = /var/log/secure  
maxretry = 3  
bantime = 3600  
action = firewallcmd-ipset  
[centos@centosstream9 ~]$ █
```

- Restart Fail2ban to activate new settings

```
action = firewallcmd-ipset  
[centos@centosstream9 ~]$ sudo systemctl restart firewalld  
[centos@centosstream9 ~]$ █
```

- Check status of fail2ban enabled jails and ssh jail

```
sudo fail2ban-client status
```

```
Sep 05 11:24:08 centosstream9.linuxvmimages.local systemd[1]: Started Firewall Service.  
[centos@centosstream9 ~]$ sudo fail2ban-client status  
Status  
|- Number of jail:      1  
|- Jail list:    sshd  
[centos@centosstream9 ~]$ █
```

```
[centos@centosstream9 ~]$ sudo fail2ban-client status
Sep 05 11:24:07 centosstream9.linuxvmimages.local systemd[1]: Starting firewalld ->
Sep 05 11:24:08 centosstream9.linuxvmimages.local systemd[1]: Started firewalld ->
[centos@centosstream9 ~]$ sudo fail2ban-client status
Status
|- Number of jail:    1
|- Jail list:    sshd
[centos@centosstream9 ~]$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed:    0
| |- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd + _COMM=sshd-session
`- Actions
  |- Currently banned: 0
  |- Total banned:    0
  |- Banned IP list:
[centos@centosstream9 ~]$
```

```
sudo fail2ban-client status sshd
```

- Try wrong password (more than 3 times)

```
ssh -p 2222 user1@testsite.mydomain.local
```

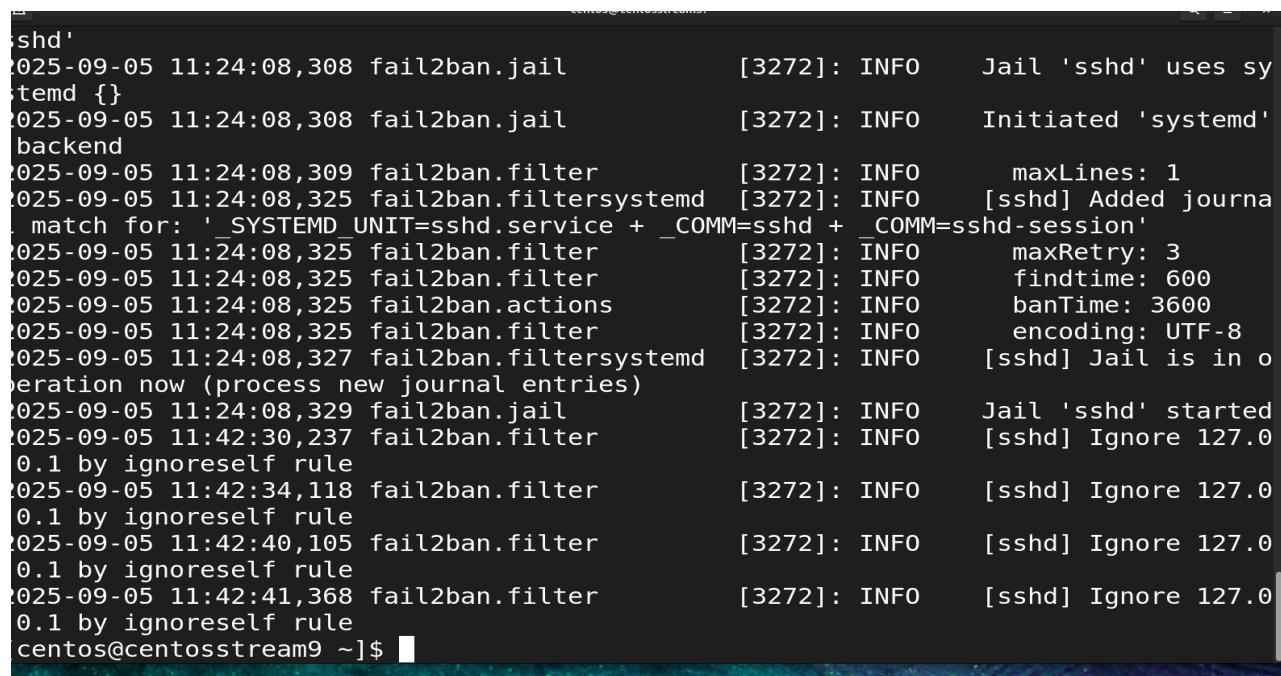
```
- Banned IP list:
[centos@centosstream9 ~]$ ssh -p 2222 user1@testsite.mydomain.local
+-----+
          LINUXVMIMAGES.COM
+-----+
          User Name: centos
          Password: centos (sudo su -)
user1@testsite.mydomain.local's password:
Permission denied, please try again.
user1@testsite.mydomain.local's password:
Permission denied, please try again.
user1@testsite.mydomain.local's password:
Received disconnect from 127.0.0.1 port 2222:2: Too many authentication failures
Disconnected from 127.0.0.1 port 2222
[centos@centosstream9 ~]$
```

- Confirm Fail2Ban jail status again and it has blocked ip

```
sudo tail -n 50 /var/log/fail2ban.log
```

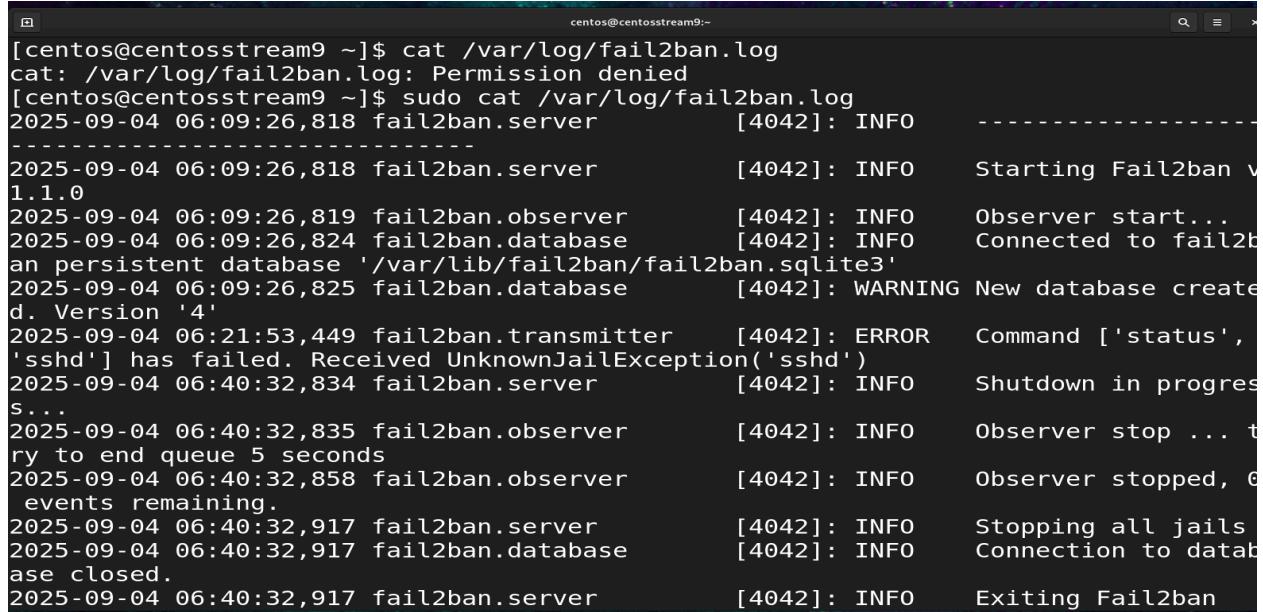
- We can also use sudo firewall-cmd --list-all

```
[centos@centosstream9 ~]$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
    services: cockpit dhcpcv6-client ssh
    ports: 2222/tcp
    protocols:
      forward: yes
      masquerade: no
    forward-ports:
    source-ports:
    icmp-blocks:
    rich rules:
[centos@centosstream9 ~]$ sudo firewall-cmd --list-ips
usage: 'firewall-cmd --help' for usage information or see firewall-cmd(1) man page
firewall-cmd: error: unrecognized arguments: --list-ips
[centos@centosstream9 ~]$ sudo firewall-cmd --list-all | grep banned
[centos@centosstream9 ~]$ sudo tail -n 50 /var/log/fail2ban.log
```



```
sshd'
2025-09-05 11:24:08,308 fail2ban.jail [3272]: INFO Jail 'sshd' uses sy
temd {}
2025-09-05 11:24:08,308 fail2ban.jail [3272]: INFO Initiated 'systemd'
backend
2025-09-05 11:24:08,309 fail2ban.filter [3272]: INFO maxLines: 1
2025-09-05 11:24:08,325 fail2ban.filtersystemd [3272]: INFO [sshd] Added journa
l match for: '_SYSTEMD_UNIT=sshd.service + _COMM=sshd + _COMM=sshd-session'
2025-09-05 11:24:08,325 fail2ban.filter [3272]: INFO maxRetry: 3
2025-09-05 11:24:08,325 fail2ban.filter [3272]: INFO findtime: 600
2025-09-05 11:24:08,325 fail2ban.actions [3272]: INFO banTime: 3600
2025-09-05 11:24:08,325 fail2ban.filter [3272]: INFO encoding: UTF-8
2025-09-05 11:24:08,327 fail2ban.filtersystemd [3272]: INFO [sshd] Jail is in o
peration now (process new journal entries)
2025-09-05 11:24:08,329 fail2ban.jail [3272]: INFO Jail 'sshd' started
2025-09-05 11:42:30,237 fail2ban.filter [3272]: INFO [sshd] Ignore 127.0
.1 by ignoreself rule
2025-09-05 11:42:34,118 fail2ban.filter [3272]: INFO [sshd] Ignore 127.0
.1 by ignoreself rule
2025-09-05 11:42:40,105 fail2ban.filter [3272]: INFO [sshd] Ignore 127.0
.1 by ignoreself rule
2025-09-05 11:42:41,368 fail2ban.filter [3272]: INFO [sshd] Ignore 127.0
.1 by ignoreself rule
centos@centosstream9 ~]$ █
```

- We can also check fail2ban logs using
`sudo cat /var/log/fail2ban.log`
- Log Analysis: Access is denied. “ignore self rule” means the server is protected from accidentally banning itself, and Fail2ban is operating correctly. We can now confidently simulate brute force or failed logins from an external IP to see Fail2ban add that IP to the ban list.



```

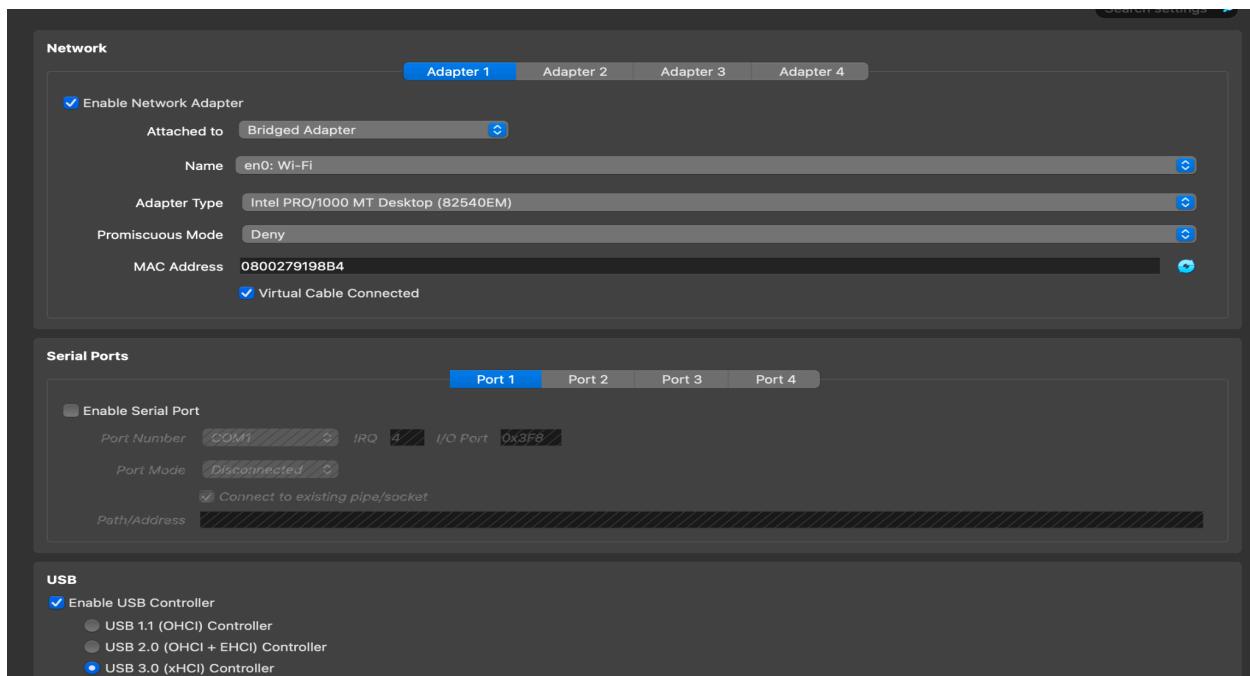
centos@centosstream9:~$ cat /var/log/fail2ban.log
cat: /var/log/fail2ban.log: Permission denied
[centos@centosstream9 ~]$ sudo cat /var/log/fail2ban.log
2025-09-04 06:09:26,818 fail2ban.server [4042]: INFO -----
-----
2025-09-04 06:09:26,818 fail2ban.server [4042]: INFO Starting Fail2ban v
1.1.0
2025-09-04 06:09:26,819 fail2ban.observer [4042]: INFO Observer start...
2025-09-04 06:09:26,824 fail2ban.database [4042]: INFO Connected to fail2b
an persistent database '/var/lib/fail2ban/fail2ban.sqlite3'
2025-09-04 06:09:26,825 fail2ban.database [4042]: WARNING New database create
d. Version '4'
2025-09-04 06:21:53,449 fail2ban.transmitter [4042]: ERROR Command ['status',
'sshd'] has failed. Received UnknownJailException('sshd')
2025-09-04 06:40:32,834 fail2ban.server [4042]: INFO Shutdown in progres
s...
2025-09-04 06:40:32,835 fail2ban.observer [4042]: INFO Observer stop ... t
ry to end queue 5 seconds
2025-09-04 06:40:32,858 fail2ban.observer [4042]: INFO Observer stopped, 0
events remaining.
2025-09-04 06:40:32,917 fail2ban.server [4042]: INFO Stopping all jails
2025-09-04 06:40:32,917 fail2ban.database [4042]: INFO Connection to databa
se closed.
2025-09-04 06:40:32,917 fail2ban.server [4042]: INFO Exiting Fail2ban

```

Testing: Manual Attack

Step 1: Setup

- Switch off the VM in virtual box



- Configure Network in VM settings to Bridged Adapter, check Virtual Cable and Promiscuous mode must be Denied
- Next, find out IP of the VM now both in which we have honeypot (centOS) and the other which will act as attacker
- To know Window OS device IP → We open command prompt → ipconfig → press enter → we get IP which will be in banned IPs list
- To know centOS (Honeypot) device ID → open terminal → ip addr → press enter → we get device IP

```
[centos@centosstream9 ~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:53:e7:0d brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.184/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
        valid_lft 6653sec preferred_lft 6653sec
    inet6 fe80::a00:27ff:fe53:e70d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[centos@centosstream9 ~]$
```

- Ping the cent os or honeypot ip from attacker

```
Command Prompt
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vboxuser>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::eaf4:3fac:6a5c:a1c3%8
IPv4 Address. . . . . : 192.168.0.28
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1

C:\Users\vboxuser>ping 192.168.0.184

Pinging 192.168.0.184 with 32 bytes of data:
Reply from 192.168.0.184: bytes=32 time=1ms TTL=64
Reply from 192.168.0.184: bytes=32 time<1ms TTL=64
Reply from 192.168.0.184: bytes=32 time<1ms TTL=64
Reply from 192.168.0.184: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.184:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\vboxuser>
```

- Ping from centOS / honeypot to windows to confirm bilateral communication

```
[centos@centosstream9 ~]$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data
.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=9.24 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=9.05 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=9.27 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=2.18 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=64 time=2.12 ms
```

Step 2: Attempt Brute force Attack

- Now since we have devices ready and can communicate with each other. We now launch our attack from windows OS
- Type command `ssh user1@192.168.0.184 -p 2222`
- Type random password multiple times until it says "Too many authentication failures"

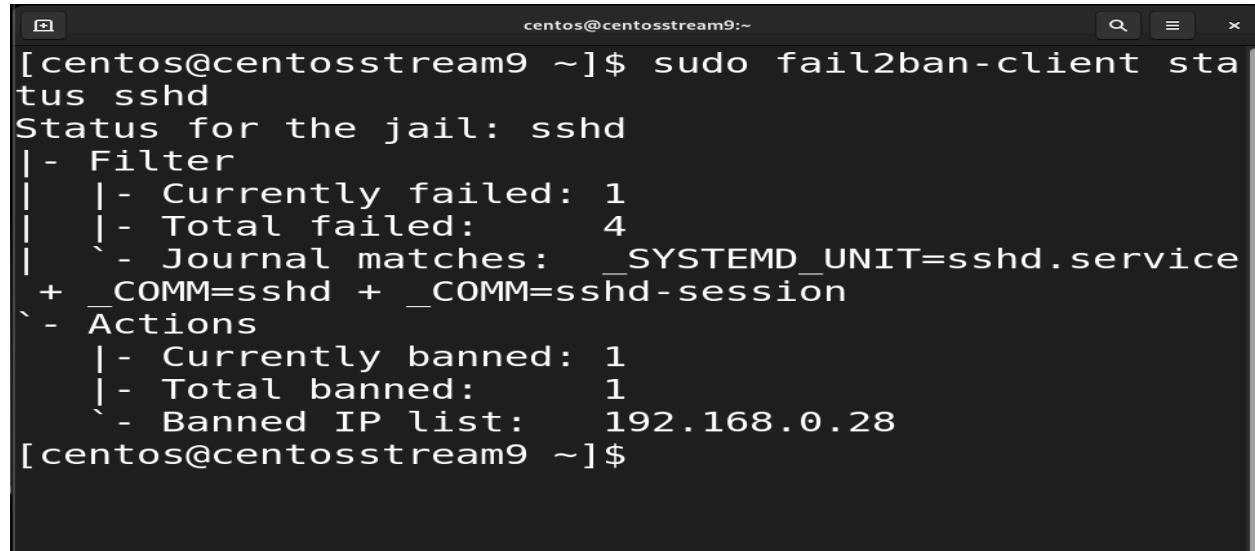
```
C:\Users\vboxuser>ssh user1@192.168.0.184 -p 2222
Reply from 192.168.0.184: bytes=32 time=1ms TTL=64
Reply from 192.168.0.184: bytes=32 time<1ms TTL=64
Reply from 192.168.0.184: bytes=32 time<1ms TTL=64
Reply from 192.168.0.184: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.184:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\vboxuser>ssh user1@192.168.0.184 -p 2222
The authenticity of host '[192.168.0.184]:2222 ([192.168.0.184]:2222)' can't be established.
ED25519 key fingerprint is SHA256:CwCMdUrZadeJ8s3ihT4rws05cwAOsUJ0z8DTHPmcW8U.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.0.184]:2222' (ED25519) to the list of known hosts.
+-----+
          LINUXVMIMAGES.COM
+-----+
          User Name: centos
          Password: centos (sudo su -)
user1@192.168.0.184's password:
Permission denied, please try again.
user1@192.168.0.184's password:
Permission denied, please try again.
user1@192.168.0.184's password:
Received disconnect from 192.168.0.184 port 2222: Too many authentication failures
Disconnected from 192.168.0.184 port 2222

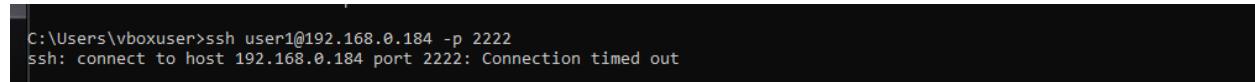
C:\Users\vboxuser>
```

- Next check the logs of fail2ban using command `sudo fail2ban-client status sshd`
- The logs must include the IP of attacker in banned IP list



```
[centos@centosstream9 ~]$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| | - Currently failed: 1
| | - Total failed: 4
| | - Journal matches: _SYSTEMD_UNIT=sshd.service
+ - COMM=sshd + _COMM=sshd-session
`- Actions
  |- Currently banned: 1
  |- Total banned: 1
  - Banned IP list: 192.168.0.28
[centos@centosstream9 ~]$
```

- Also note next time, if attacker uses same IP then the connection gets timed out as shown in screenshot below



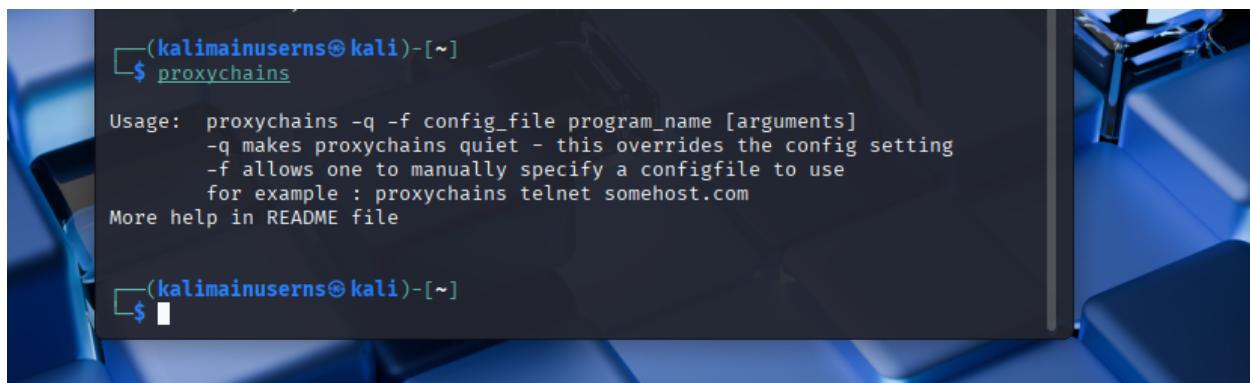
```
C:\Users\vboxuser>ssh user1@192.168.0.184 -p 2222
ssh: connect to host 192.168.0.184 port 2222: Connection timed out
```

Penetration Testing Through Proxychains and Tor

Setup Kali Linux in your VM ([guide](#))

Step 1: Configure Tor on Kali Linux

- Check if Proxychain is installed in Kali. To check just type `proxychain` in the terminal (if no we will install in next step)

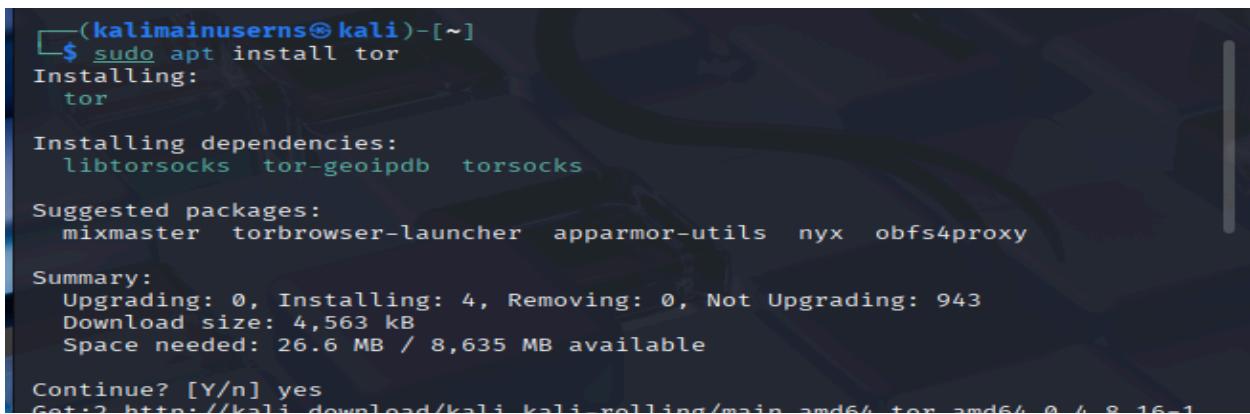


```
(kalimainusersns㉿kali)-[~]
$ proxychains

Usage: proxychains -q -f config_file program_name [arguments]
      -q makes proxychains quiet - this overrides the config setting
      -f allows one to manually specify a configfile to use
      for example : proxychains telnet somehost.com
More help in README file

(kalimainusersns㉿kali)-[~]
$
```

- Install and Start Tor Service
- To install Tor run the command `sudo apt install tor`



```
(kalimainusersns㉿kali)-[~]
$ sudo apt install tor
Installing:
  tor

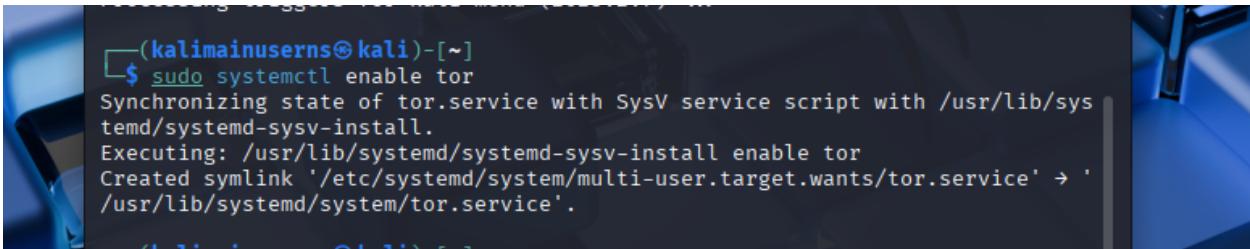
Installing dependencies:
  libtorsocks  tor-geoipdb  torsocks

Suggested packages:
  mixmaster  torbrowser-launcher  apparmor-utils  nyx  obfs4proxy

Summary:
  Upgrading: 0, Installing: 4, Removing: 0, Not Upgrading: 943
  Download size: 4,563 kB
  Space needed: 26.6 MB / 8,635 MB available

Continue? [Y/n] yes
Get:2 http://kali.download/kali kali-rolling/main amd64 tor amd64 0.4.8.16-1
```

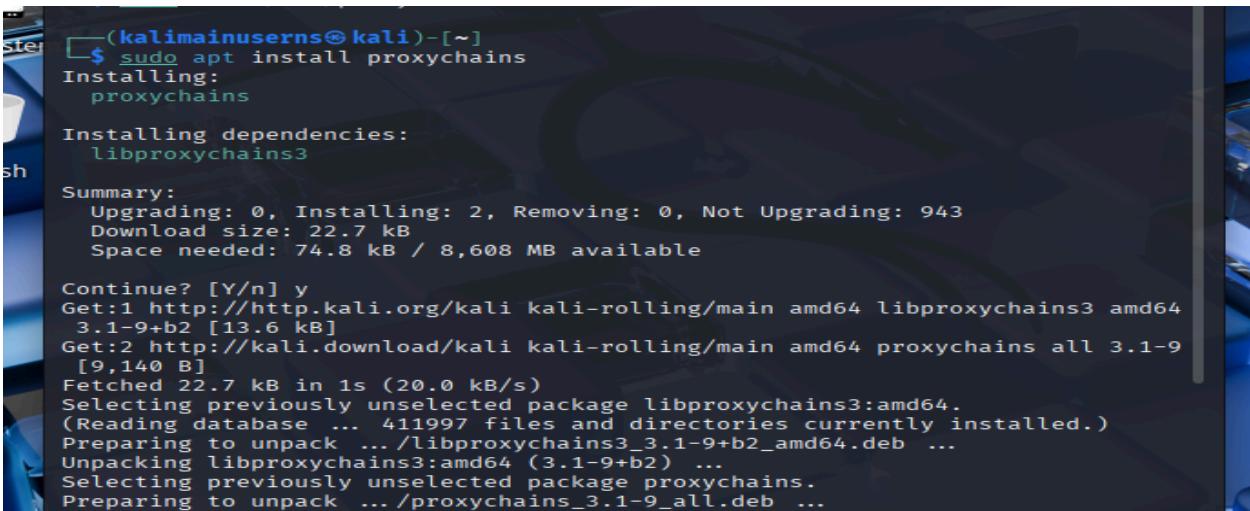
- Enable Tor to start on boot, `sudo systemctl enable tor`



```
(kalimainusersns㉿kali)-[~]
└─$ sudo systemctl enable tor
Synchronizing state of tor.service with SysV service script with /usr/lib/sys
temd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable tor
Created symlink '/etc/systemd/system/multi-user.target.wants/tor.service' → '/
/usr/lib/systemd/system/tor.service'.
```

Step 2: Configure Proxychains to Use Tor

- Install proxychains (if not already installed) using command `sudo apt install proxychains`



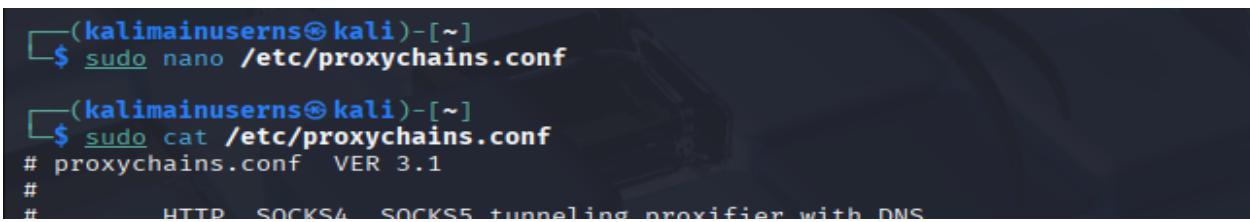
```
(kalimainusersns㉿kali)-[~]
└─$ sudo apt install proxychains
Installing:
  proxychains

Installing dependencies:
  libproxychains3

Summary:
  Upgrading: 0, Installing: 2, Removing: 0, Not Upgrading: 943
  Download size: 22.7 kB
  Space needed: 74.8 kB / 8,608 MB available

Continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-rolling/main amd64 libproxychains3 amd64
  3.1-9+b2 [13.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 proxychains all 3.1-9
  [9,140 B]
Fetched 22.7 kB in 1s (20.0 kB/s)
Selecting previously unselected package libproxychains3:amd64.
(Reading database ... 411997 files and directories currently installed.)
Preparing to unpack .../libproxychains3_3.1-9+b2_amd64.deb ...
Unpacking libproxychains3:amd64 (3.1-9+b2) ...
Selecting previously unselected package proxychains.
Preparing to unpack .../proxychains_3.1-9_all.deb ...
```

- Open the proxychains configuration file using command `sudo nano /etc/proxychains.conf`
- Scroll to the bottom, find the line that defines your proxy list, and ensure it includes the Tor Socks4 / socks5 proxy(if doesn't exist):
`socks4 127.0.0.1 9050 or socks5 127.0.0.1 9050`
- Save and exit (in nano: Ctrl+O, then Ctrl+X)



```
(kalimainusersns㉿kali)-[~]
└─$ sudo nano /etc/proxychains.conf

(kalimainusersns㉿kali)-[~]
└─$ sudo cat /etc/proxchains.conf
# proxychains.conf  VER 3.1
#
#           HTTP   SOCKS4   SOCKS5 tunneling proxifier with DNS
```

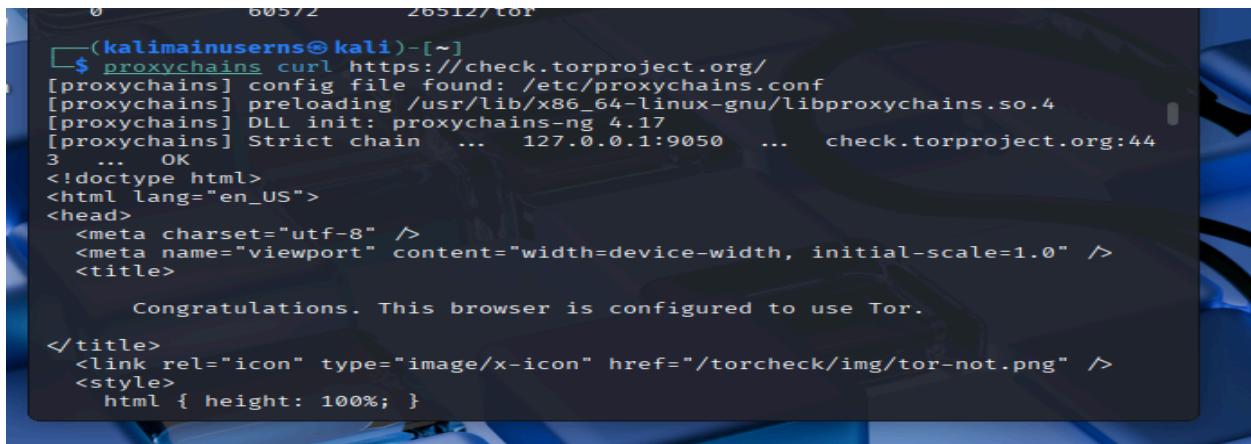
```
tcp_connect_time_out 8000

# ProxyList format
#     type host port [user pass]
#     (values separated by 'tab' or 'blank')
#
#
# Examples:
#
#     socks5 192.168.67.78    1080      lamer    secret
#     http   192.168.89.3     8080      justu    hidden
#     socks4 192.168.1.49     1080
#     http   192.168.39.93    8080
#
#
# proxy types: http, socks4, socks5
#           ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 9050
```

Test Proxychains with Tor

- Test if proxychains and Tor are working by running proxychains

```
sudo curl https://check.torproject.org/
```



```
0          60572      26512/tor
└─(kalimainusers㉿kali)-[~]
  $ proxychains curl https://check.torproject.org/
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain ... 127.0.0.1:9050 ... check.torproject.org:44
3 ... OK
<!doctype html>
<html lang="en_US">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>
    Congratulations. This browser is configured to use Tor.
  </title>
  <link rel="icon" type="image/x-icon" href="/torcheck/img/tor-not.png" />
  <style>
    html { height: 100%; }
```

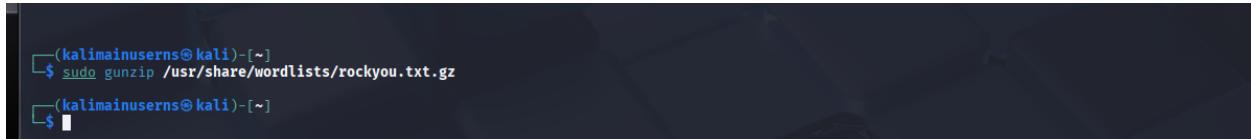
Step 3: Prepare Tools and Wordlists

Extract Wordlists

- Kali Linux includes a popular password dictionary (`rockyou.txt`) located at `/usr/share/wordlists/rockyou.txt.gz`.
- Extract it first:

```
sudo gunzip /usr/share/wordlists/rockyou.txt.gz
```

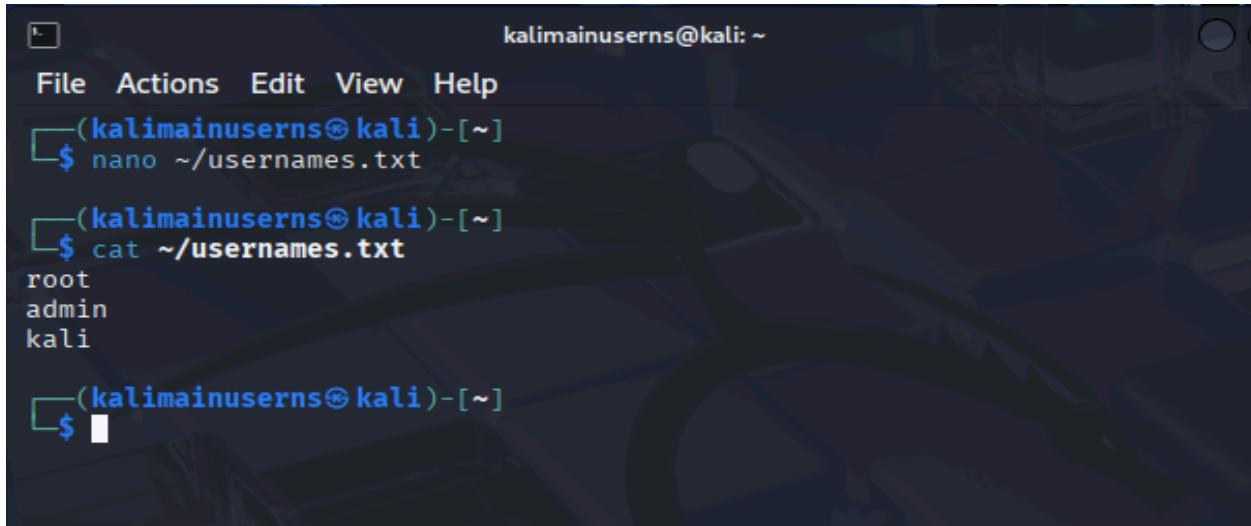
- This will provide /usr/share/wordlists/rockyou.txt for use.



```
(kalimainusersns㉿kali)-[~]
$ sudo gunzip /usr/share/wordlists/rockyou.txt.gz
(kalimainusersns㉿kali)-[~]
$
```

Prepare Username List

- Create a file (e.g., usernames.txt) with possible usernames.
Example: nano ~/usernames.txt



```
kalimainusersns㉿kali: ~
File Actions Edit View Help
(kalimainusersns㉿kali)-[~]
$ nano ~/usernames.txt
(kalimainusersns㉿kali)-[~]
$ cat ~/usernames.txt
root
admin
kali
(kalimainusersns㉿kali)-[~]
$
```

- Identify Your Target (IP: 192.168.0.184, Port: 2222)

Step 4: Attack

Construct Your Hydra Command

- The command structure for Hydra with proxychains for SSH is:
`hydra -L ~/usernames.txt -P /usr/share/wordlists/rockyou.txt
ssh://TARGET_IP -s PORT -t 4`
- `-L ~/usernames.txt` : username list file
- `-P /usr/share/wordlists/rockyou.txt` : password list file
- `ssh://TARGET_IP` : target server IP

- `-s PORT` : SSH port (default is 22, but use 2222 or your custom port if needed)
- `-t 4` : number of parallel tasks, can adjust for performance
- So our target IP is `192.168.0.184` and SSH port is 2222:
`hydra -L ~/usernames.txt -P /usr/share/wordlists/rockyou.txt ssh://192.168.0.184 -s 2222 -t 4`

```
centos@192.168.0.184's password:

└─(kalimainusersns㉿kali)-[~]
$ hydra -L ~/usernames.txt -P /usr/share/wordlists/rockyou.txt ssh://192.168.0.184 -s 2222 -t 4

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-08 02:29:25
[DATA] max 4 tasks per 1 server, overall 4 tasks, 43033197 login tries (l:3/p:14344399), ~10758300 tries per task
[DATA] attacking ssh://192.168.0.184:2222/
[STATUS] 12.00 tries/min, 12 tries in 00:01h, 43033185 to do in 59768:19h, 4 active
[ERROR] all children were disabled due to too many connection errors
0 of 1 target completed, 0 valid password found
[INFO] Writing restore file because 2 server scans could not be completed
[ERROR] 1 target was disabled because of too many errors
[ERROR] 1 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-08 02:31:33

└─(kalimainusersns㉿kali)-[~]
$ █
```

- Once hydra started attacking, you can check logs using `sudo tail -f /var/log/secure`
- Also check logs in fail2ban `sudo tail -f /var/log/fail2ban.log` and also `sudo journalctl -u fail2ban -f`

Step 5: Analyse Logs Manually

- The screenshots show that:

- Hydra successfully made SSH connection attempts and tried 12 logins before encountering too many errors and disabling tasks.
 - CentOS security logs (`/var/log/secure`, `/var/log/fail2ban.log`, `sudo tail -f /var/log/fail2ban.log`, `sudo journalctl -u fail2ban -f`) properly captured these brute-force attempts, at corresponding timestamps and multiple ports.
 - Confirmation
 - The brute-force attack is visible and traceable in the `/var/log/secure` log on CentOS.
 - Every incorrect password attempt is recorded with source IP, username, and port details.
 - The target system logs all failed attempts as expected for SSH brute-force attacks.
 - The Hydra task was stopped by connection errors, likely because of rapid-fire requests or temporary security measures (such as SSH rate limiting or protections).

```
[centos@centosstream9 ~]$ sudo tail -f /var/log/secure
Sep  8 02:29:49 centosstream9 unix_chkpwd[3302]: password check
failed for user (root)
Sep [centos@centosstream9 ~]$ sudo cat sudo tail /var/log/fail2ban.log
uthencat: sudo: No such file or directory
t=192 2025-09-04 06:09:26,818 fail2ban.server      [4042]: INFO  -----
utthen 2025-09-04 06:09:26,818 fail2ban.server      [4042]: INFO  Starting Fail2ban v1.1.0
t=192 2025-09-04 06:09:26,819 fail2ban.observer    [4042]: INFO  Observer start...
utthen 2025-09-04 06:09:26,824 fail2ban.database    [4042]: INFO  Connected to fail2ban persistent database '/var/lib/
t=192 fail2ban/fail2ban.sqlite3'
Sep  2025-09-04 06:09:26,825 fail2ban.database      [4042]: WARNING New database created. Version '4'
Sep  2025-09-04 06:21:53,449 fail2ban.transmitter   [4042]: ERROR Command ['status', 'sshd'] has failed. Received Unkn
failownJailException('sshd')
Sep  2025-09-04 06:40:32,834 fail2ban.server      [4042]: INFO  Shutdown in progress...
utthen 2025-09-04 06:40:32,835 fail2ban.observer    [4042]: INFO  Observer stop ... try to end queue 5 seconds
t=192 2025-09-04 06:40:32,858 fail2ban.observer    [4042]: INFO  Observer stopped, 0 events remaining.
t=192 2025-09-04 06:40:32,917 fail2ban.server      [4042]: INFO  Stopping all jails
Sep  2025-09-04 06:40:32,917 fail2ban.database    [4042]: INFO  Connection to database closed.
valid 2025-09-04 06:40:32,917 fail2ban.server      [4042]: INFO  -----
valid 2025-09-04 06:40:33,084 fail2ban.server      [4234]: INFO  -----
Sep  2025-09-04 06:40:33,084 fail2ban.server      [4234]: INFO  Starting Fail2ban v1.1.0
valid 2025-09-04 06:40:33,085 fail2ban.observer    [4234]: INFO  Observer start...
valid 2025-09-04 06:40:33,089 fail2ban.database    [4234]: INFO  Connected to fail2ban persistent database '/var/lib/
Sep  2025-09-04 06:40:33,089 fail2ban/fail2ban.sqlite3'
valid 2025-09-04 06:40:33,089 fail2ban.jail       [4234]: INFO  Creating new jail 'sshd'
valid 2025-09-04 06:40:33,101 fail2ban.jail       [4234]: INFO  Jail 'sshd' uses systemd {}
Sep  2025-09-04 06:40:33,101 fail2ban.jail       [4234]: INFO  Initiated 'systemd' backend
valid 2025-09-04 06:40:33,102 fail2ban.filter     [4234]: INFO  maxLines: 1
valid 2025-09-04 06:40:33,115 fail2ban.filtersystemd [4234]: INFO  [sshd] Added journal match for: '_SYSTEMD_UNIT=sshd'.
Sep  service + _COMM=sshd + COMM=sshd-session'      [4234]: INFO  -----
Sep  2025-09-04 06:40:33,115 fail2ban.filter     [4234]: INFO  maxRetry: 3
: ses 2025-09-04 06:40:33,115 fail2ban.filter     [4234]: INFO  findtime: 600
Sep  2025-09-04 06:40:33,115 fail2ban.actions    [4234]: INFO  banTime: 3600
PWD=/home/centos ; USER=root ; COMMAND=/bin/tail -f /var/log/sec
ure
Sep  8 02:30:26 centosstream9 sudo[3328]: pam_unix(sudo:session)
: session opened for user root(uid=0) by centos(uid=1000)
^C
```

- The test setup is functioning as expected for attack simulation and logging.

```

2025-09-08 02:04:28,440 fail2ban.filter [1183]: INFO [sshd] jail is in operation now (process new journals)
entries)
2025-09-08 02:04:28,454 fail2ban.jail      [1183]: INFO Jail 'sshd' started
2025-09-08 02:27:57,956 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:27:57
2025-09-08 02:29:49,323 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,528 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,850 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,851 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,854 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,884 fail2ban.actions  [1183]: NOTICE [sshd] Ban 192.168.0.209
2025-09-08 02:29:51,851 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:51
2025-09-08 02:29:51,853 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:51
2025-09-08 02:29:51,854 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:51
2025-09-08 02:29:51,854 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:51
[centos@centosstream9 ~]$ sudo tail -f /var/log/fail2ban.log
2025-09-08 02:29:49,323 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,528 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,850 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,851 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,854 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:49
2025-09-08 02:29:49,884 fail2ban.actions  [1183]: NOTICE [sshd] Ban 192.168.0.209
2025-09-08 02:29:51,851 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:51
2025-09-08 02:29:51,853 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:51
2025-09-08 02:29:51,854 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:51
2025-09-08 02:29:51,854 fail2ban.filter    [1183]: INFO [sshd] Found 192.168.0.209 - 2025-09-08 02:29:51
^C

```

Automating Log Analysis and Reporting

Step 1: Define Your Log Files and Data Points

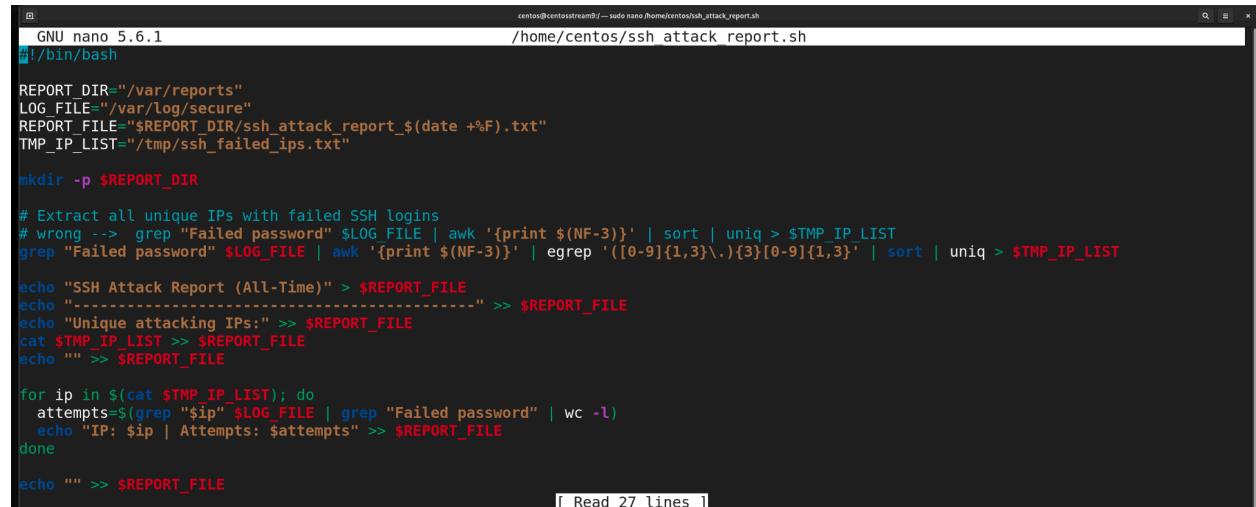
Primary log file: `/var/log/secure` (SSH login attempts)

Key metrics to extract:

- This script must return
- List unique attacking IPs.
- Count the number of failed attempts for each IP.
- Output a simple, readable report

Step 2: Write Basic Bash Script for Daily Report

Create a file called `ssh_attack_report.sh` in home directory with the script below (in the screenshot)



```
GNU nano 5.6.1 centos@centosstream8: ~ - sudo nano /home/centos/ssh_attack_report.sh
#!/bin/bash

/home/centos/ssh attack report.sh

REPORT_DIR="/var/reports"
LOG_FILE="/var/log/secure"
REPORT_FILE="$REPORT_DIR/ssh_attack_report_$(date +%F).txt"
TMP_IP_LIST="/tmp/ssh_failed_ips.txt"

mkdir -p $REPORT_DIR

# Extract all unique IPs with failed SSH logins
# wrong --> grep "Failed password" $LOG_FILE | awk '{print $(NF-3)}' | sort | uniq > $TMP_IP_LIST
grep "Failed password" $LOG_FILE | awk '{print $(NF-3)}' | egrep '([0-9]{1,3}\.){3}[0-9]{1,3}' | sort | uniq > $TMP_IP_LIST

echo "SSH Attack Report (All-Time)" > $REPORT_FILE
echo "-----" >> $REPORT_FILE
echo "Unique attacking IPs:" >> $REPORT_FILE
cat $TMP_IP_LIST >> $REPORT_FILE
echo "" >> $REPORT_FILE

for ip in $(cat $TMP_IP_LIST); do
    attempts=$(grep "$ip" $LOG_FILE | grep "Failed password" | wc -l)
    echo "IP: $ip | Attempts: $attempts" >> $REPORT_FILE
done

echo "" >> $REPORT_FILE
```

[Read 27 lines]

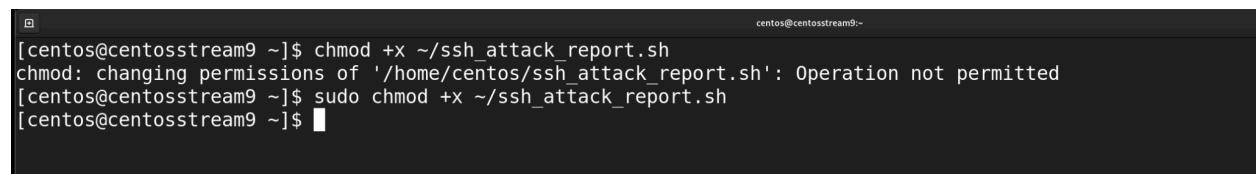
This above script:

- List unique attacking IPs.
- Count the number of failed attempts for each IP.
- Output a simple, readable report

Step 3: Make Script Executable

Give the file executable permission

```
sudo chmod +x ~/ssh_attack_report.sh
```



A terminal window showing the execution of the chmod command. The user runs 'chmod +x ~/ssh_attack_report.sh' but receives an error message: 'chmod: changing permissions of '/home/centos/ssh_attack_report.sh': Operation not permitted'. Then, the user runs 'sudo chmod +x ~/ssh_attack_report.sh' successfully, indicated by the lack of an error message.

```
[centos@centosstream9 ~]$ chmod +x ~/ssh_attack_report.sh
chmod: changing permissions of '/home/centos/ssh_attack_report.sh': Operation not permitted
[centos@centosstream9 ~]$ sudo chmod +x ~/ssh_attack_report.sh
[centos@centosstream9 ~]$ █
```

Step 4: Run and Test the Script

Run: `sudo ~/ssh_attack_report.sh`

Test: `cat /var/reports/ssh_attack_2025-09-08.txt`

Below screenshot shows the result of the commands above:

```
[centos@centosstream9 /]$ sudo ~/ssh_attack_report.sh
[centos@centosstream9 /]$ cat /var/reports/ssh_attack_report_2025-09-08.txt
SSH Attack Report (All-Time)
-----
Unique attacking IPs:
127.0.0.1
192.168.0.209
192.168.0.28

IP: 127.0.0.1 | Attempts: 7
IP: 192.168.0.209 | Attempts: 4
IP: 192.168.0.28 | Attempts: 3

Report generated at Mon Sep  8 06:48:59 AM EDT 2025
[centos@centosstream9 /]$ sudo nano ~/ssh_attack_report.sh
[centos@centosstream9 /]$ cat /var/reports/ssh_attack_report_2025-09-08.txt
SSH Attack Report (All-Time)
-----
Unique attacking IPs:
127.0.0.1
192.168.0.209
192.168.0.28

IP: 127.0.0.1 | Attempts: 7
IP: 192.168.0.209 | Attempts: 4
IP: 192.168.0.28 | Attempts: 3

Report generated at Mon Sep  8 06:48:59 AM EDT 2025
[centos@centosstream9 /]$ █
```

Step 5: Let's Automate the Script Using Cron For Daily or Weekly reporting

How to Schedule the Script with Cron

```
File Edit View Search Terminal Help
[centos@centosstream9 ~]$ sudo crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
```

Edit the Root User's Crontab and Launch crontab `sudo crontab -e`

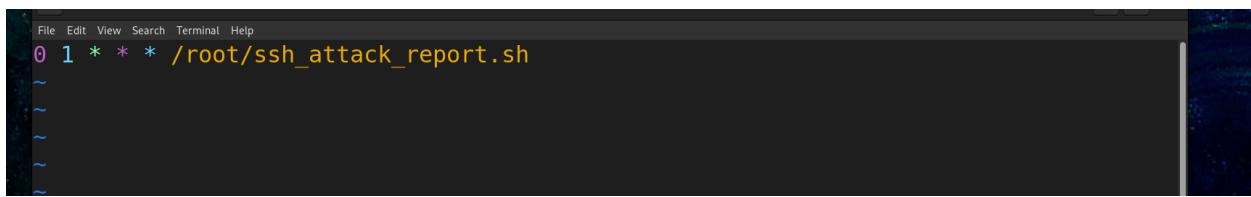
Add a Cron Schedule Line

For Daily Reporting (implemented this one) at 1:00 AM:

```
0 1 * * * /root/ssh_attack_report.sh
```

For weekly reporting at 1:00AM Every Monday:

```
0 1 * * 1 /root/ssh_attack_report.sh
```

A screenshot of a terminal window with a dark background. At the top, there is a menu bar with options: File, Edit, View, Search, Terminal, and Help. Below the menu, the command `0 1 * * 1 /root/ssh_attack_report.sh` is displayed in yellow text. There are four additional lines of the same command below it, each starting with a tilde (~).

```
File Edit View Search Terminal Help
0 1 * * 1 /root/ssh_attack_report.sh
~
~
~
~
```

Verification

After the next scheduled time, check `/var/reports/` for a new report file.

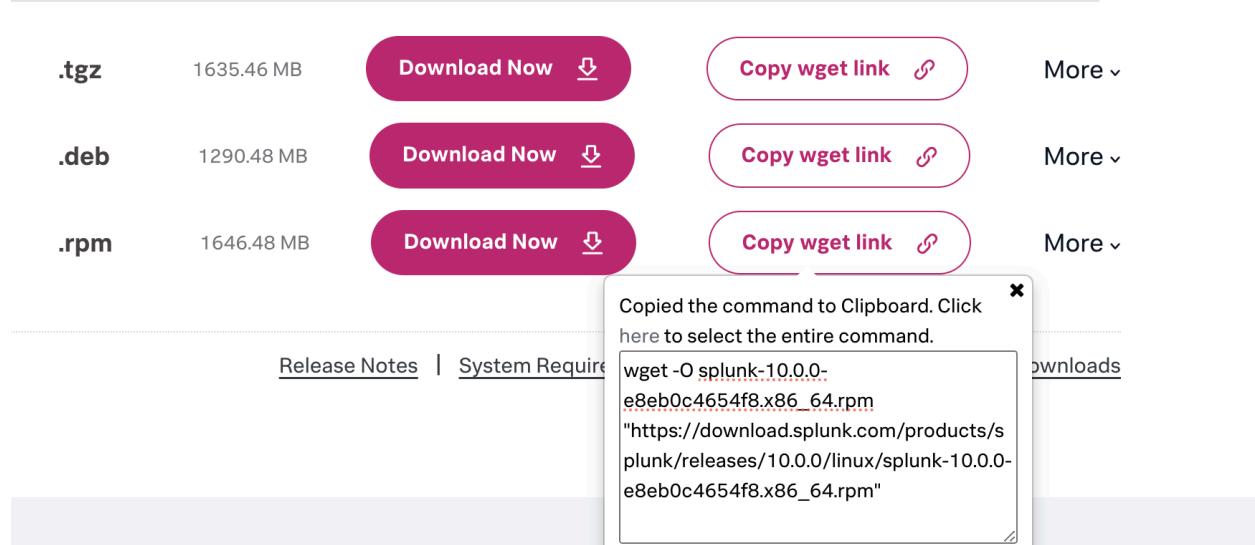
You can also check if the cron job ran by looking in cron logs:

```
sudo grep CRON /var/log/secure
```

Integrating all logs with SIEM Tools for Better Log Analysis

Step 1: Install Splunk from Official Website

- Select the package of splunk suitable for you from [Official Website](#)



- Find the link for the wget command like in the screenshot.

On CentOS Command Line use command:

```
wget -O splunk-10.0.0-e8eb0c4654f8.x86_64.rpm  
"https://download.splunk.com/products/splunk/releases/10.0.0/linux/splunk-10.0.0-e8eb0c4654f8.x86_64.rpm"
```

```
bash: wgetthispc: command not found...
[centos@centosstream9 ~]$ wget -O splunk-10.0.0-e8eb0c4654f8.x86_64.rpm "https://download.splunk.com/products/splunk/releases/10.0.0/linux/splunk-10.0.0-e8eb0c4654f8.x86_64.rpm"
--2025-09-08 10:48:23-- https://download.splunk.com/products/splunk/releases/10.0.0/linux/splunk-10.0.0-e8eb0c4654f8.x86_64.rpm
Resolving download.splunk.com (download.splunk.com) ... 18.66.57.35, 18.66.57.80, 18.66.57.87, ...
Connecting to download.splunk.com (download.splunk.com)|18.66.57.35|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1726454835 (1.6G) [binary/octet-stream]
Saving to: 'splunk-10.0.0-e8eb0c4654f8.x86_64.rpm'

splunk-10.0.0-e8eb0c 100%[=====] 1.61G 21.1MB/s in 82s

2025-09-08 10:49:47 (20.0 MB/s) - 'splunk-10.0.0-e8eb0c4654f8.x86_64.rpm' saved [1726454835/1726454835]
```

- Install splunk with root privileges
- ```
sudo rpm -ivh splunk_package_name.rpm
```

```
[centos@centosstream9 ~]$ ls
Desktop Music reports Templates
Documents Pictures splunk-10.0.0-e8eb0c4654f8.x86_64.rpm Videos
Downloads Public ssh_attack_report.sh

[centos@centosstream9 ~]$ sudo rpm -ivh splunk-10.0.0-e8eb0c4654f8.x86_64.rpm
warning: splunk-10.0.0-e8eb0c4654f8.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID b3cd4420: NOKEY
Verifying...
```

## Step 2: Start and enable Splunk service

```
sudo /opt/splunk/bin/splunk start --accept-license
```

If you need to restart after installation we do not need to place  
`--accept-license`

When prompted, set an admin username and password for the Splunk web interface.

```
[centos@centosstream9 ~]$ sudo /opt/splunk/bin/splunk start --accept-license
systemctl: /opt/splunk/lib/libcrypto.so.3: version `OPENSSL_3.4.0' not found (required by /usr/lib64/systemd/libsystemd-shared-252.so)

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: admin
Password must contain at least:
 * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
ERROR: The password does not match.
```

- Access Splunk's web interface by clicking on the link it is hosted on and enter username password when prompted

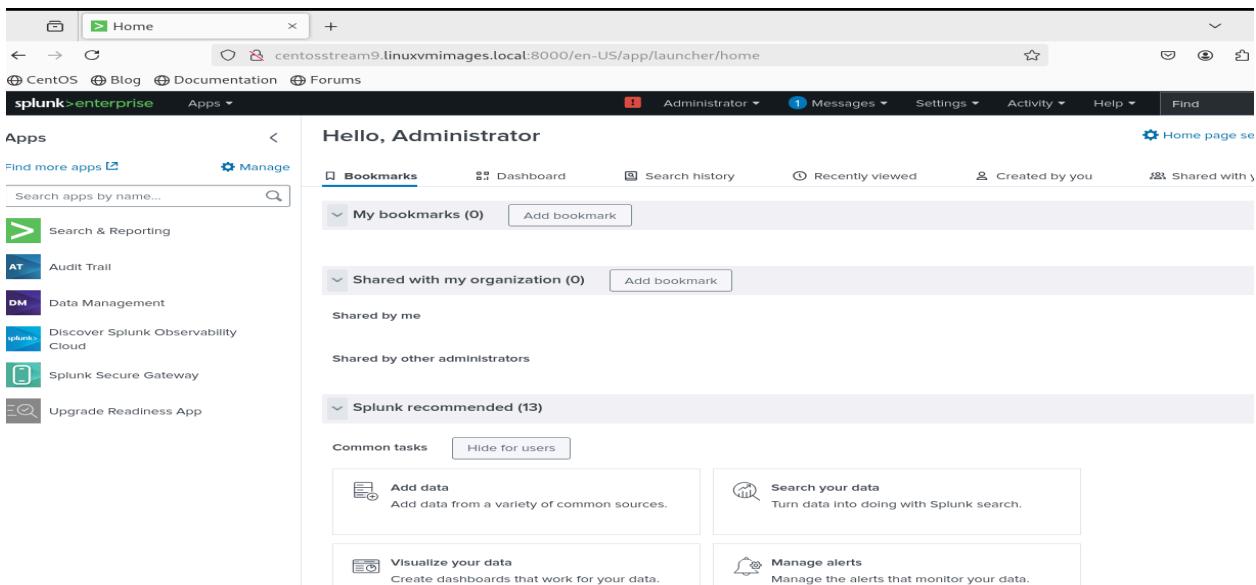
```
.....
Warning: ignoring -extensions option without -extfile
Certificate request self-signature ok
subject=CN = centosstream9.linuxvmimages.local, O = SplunkUser
pDone

Waiting for web server at http://127.0.0.1:8000 to be available.....
..... Done

If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://centosstream9.linuxvmimages.local:8000
```

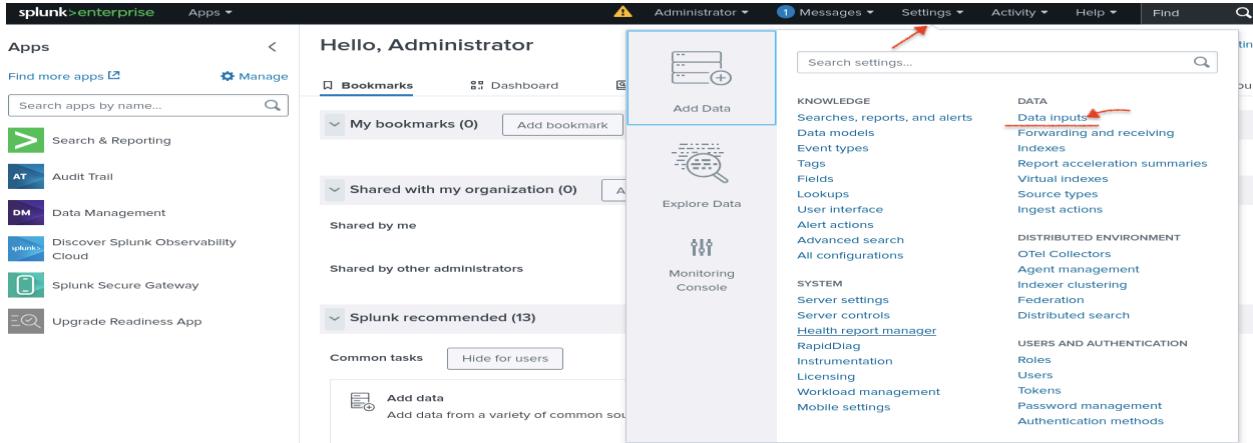
- Dashboard will open after login



## Step 3: Configure Data Inputs for Critical Logs

- From the Splunk Web UI, go to Settings > Data Inputs > Files & Directories.

- Add `/var/log/secure`, `/var/log/fail2ban.log`, and honeypot log files (if applicable).
- Since it is a less complex environment we leave index as default and source type be detected by splunk.



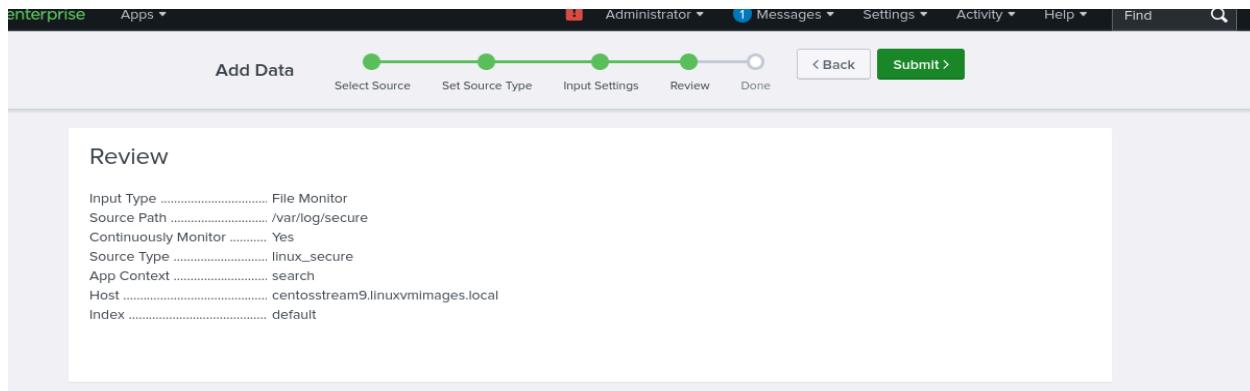
This screenshot shows the 'Local inputs' configuration page. It lists several input types with their descriptions and current counts. The columns are 'Type', 'Inputs', and 'Actions'. The 'Actions' column includes a '+ Add new' button for each row. Red arrows point to the 'Files & Directories' link and the '+ Add new' button in the 'Actions' column of the first row.

| Type                                                                                                                                           | Inputs | Actions   |
|------------------------------------------------------------------------------------------------------------------------------------------------|--------|-----------|
| <a href="#">Files &amp; Directories</a><br>Index a local file or monitor an entire directory.                                                  | 19     | + Add new |
| <a href="#">HTTP Event Collector</a><br>Receive data over HTTP or HTTPS.                                                                       | 0      | + Add new |
| <a href="#">TCP</a><br>Listen on a TCP port for incoming data, e.g. syslog.                                                                    | 0      | + Add new |
| <a href="#">UDP</a><br>Listen on a UDP port for incoming data, e.g. syslog.                                                                    | 0      | + Add new |
| <a href="#">Scripts</a><br>Run custom scripts to collect or generate more data.                                                                | 37     | + Add new |
| <a href="#">checkapp</a>                                                                                                                       | 0      | + Add new |
| <a href="#">Systemd Journald Input for Splunk</a><br>This is the input that gets data from journald (systemd's logging component) into Splunk. | 0      | + Add new |
| <a href="#">Log Input for the Splunk platform</a><br>This input collects data from logd on macOS and sends it to the Splunk platform.          | 0      | + Add new |

- Select source by browsing or typing the address directly and press next button

- Data is fetched and displayed you can click Next > button

- The data input settings must be kept by default until one knows exactly what they are doing
- Next click Review > button



- Then click Submit
- Repeat these steps (all in Step 3) for any more logs which you want to act as input to the Splunk SIEM.

## What Happens After This?

- Splunk will start indexing new entries appended to the monitored files.
- We can immediately begin searching via the Splunk Search Processing Language (SPL), e.g.,  
`index=main sourcetype=linux_secure "Failed password"`
- This will display the SSH failed login attempts collected by Splunk.

Splunk > enterprise Apps ▾

Administrator 1 Messages Settings Activity Help Find Search & Reporting

Search Analytics Datasets Reports Alerts Dashboards

New Search

index=main sourcetype=linux\_secure "Failed password"

5 events (9/7/25 11:00:00.000 PM to 9/8/25 11:57:11.000 PM) No Event Sampling ▾

Save As Create Table View Close

Events (5) Patterns Statistics Visualization

Timeline format ▾ Zoom Out Zoom to Selection Deselect 1 hour per column

Format Show: 20 Per Page View: List

| Time                  | Event                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9/8/25 6:16:27:000 AM | Sep 8 06:16:27 centosstream9 sudo[3318]: centos : TTY=pts/0 ; PWD=/ ; USER=root ; COMMAND=/bin/grep 'Failed password' /var/log/secure<br>host = centosstream9/linuxvmimages.local source = /var/log/secure sourcetype = linux_secure |
| 9/8/25 2:29:51:000 AM | Sep 8 02:29:51 centosstream9 sshd[3296]: Failed password for invalid user root from 192.168.0.209 port 44476 ssh2<br>host = centosstream9/linuxvmimages.local source = /var/log/secure sourcetype = linux_secure                     |
| 9/8/25 2:29:51:000 AM | Sep 8 02:29:51 centosstream9 sshd[3294]: Failed password for invalid user root from 192.168.0.209 port 44472 ssh2<br>host = centosstream9/linuxvmimages.local source = /var/log/secure sourcetype = linux_secure                     |
| 9/8/25 2:29:51:000 AM | Sep 8 02:29:51 centosstream9 sshd[3293]: Failed password for invalid user root from 192.168.0.209 port 44466 ssh2<br>host = centosstream9/linuxvmimages.local source = /var/log/secure sourcetype = linux_secure                     |
| 9/8/25 2:29:51:000 AM | Sep 8 02:29:51 centosstream9 sshd[3295]: Failed password for invalid user root from 192.168.0.209 port 44474 ssh2<br>host = centosstream9/linuxvmimages.local source = /var/log/secure sourcetype = linux_secure                     |

< Hide Fields All Fields

SELECTED FIELDS  
*a host 1*  
*a source 1*  
*a sourcetype 1*

INTERESTING FIELDS  
*a COMMAND 1*  
*# date\_hour 2*  
*# date\_mday 1*  
*# date\_minute 2*  
*# date\_month 1*  
*# date\_second 2*  
*# date\_wday 1*  
*# date\_year 1*  
*a date\_zone 1*

# **Future Enhancements: Moving Forward**

As attack trends evolve and enterprise security demands grow, several enhancements can further strengthen the solution presented in this project:

## **Deploy Advanced Honeypots:**

Integrate interactive honeypots such as Cowrie for SSH, or extend coverage to FTP, SMB, and web services, to capture detailed attack techniques and malware samples for deeper threat intelligence.

## **Strengthen Authentication:**

Transition from password-based SSH authentication to key-based access for all accounts. Introduce multi-factor authentication (MFA) for added protection against credential theft.

## **Centralize Log Management:**

Scale the Splunk or ELK-based log aggregation to monitor multiple servers, enabling integrated dashboards, real-time alerts, and organization-wide visibility of attack activities.

## **Automate Threat Intelligence Integration:**

Enrich log data with external threat intelligence feeds, providing geolocation, reputation scoring, and indicators of compromise (IoCs) to prioritize response efforts effectively.

## **Streamline Incident Response:**

Automate the generation and distribution of security incident reports. Set up scheduled email notifications to alert administrators of high-risk events or automated blocks.

## **Continuous Security Testing and Defense Tuning:**

Regularly simulate brute-force and reconnaissance attacks using updated tools to validate detection accuracy and blocking effectiveness. Adjust fail2ban, firewall, and honeypot policies in response to new adversarial tactics.

These enhancements will promote a more proactive, scalable, and resilient defense model, ready to adapt to future cybersecurity challenges in real-world environments.

# References

Splunk Documentation. (2024). Splunk Docs. Retrieved from <https://help.splunk.com/en>

Fail2ban Project. (n.d.). Fail2ban Documentation. Retrieved from [https://www.fail2ban.org/wiki/index.php/Main\\_Page](https://www.fail2ban.org/wiki/index.php/Main_Page)

OpenSSH Project. (2025). OpenSSH Official Website. Retrieved from <https://www.openssh.com>

Firewalld Project. (2025). Firewalld Documentation. Retrieved from <https://firewalld.org/documentation/>

Red Hat, Inc. (2023). TCP Wrappers Configuration Files. Retrieved from [https://docs.redhat.com/en/documentation/red\\_hat\\_enterprise\\_linux/6/html/security\\_guide/sect-security\\_guide-tcp\\_wrappers\\_and\\_xinetd-tcp\\_wrappers\\_configuration\\_files](https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/6/html/security_guide/sect-security_guide-tcp_wrappers_and_xinetd-tcp_wrappers_configuration_files)

Cowrie Honeypot Project. (n.d.). Cowrie Documentation. Retrieved from <https://cowrie.readthedocs.io/en/latest/>