Course End Project 1

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

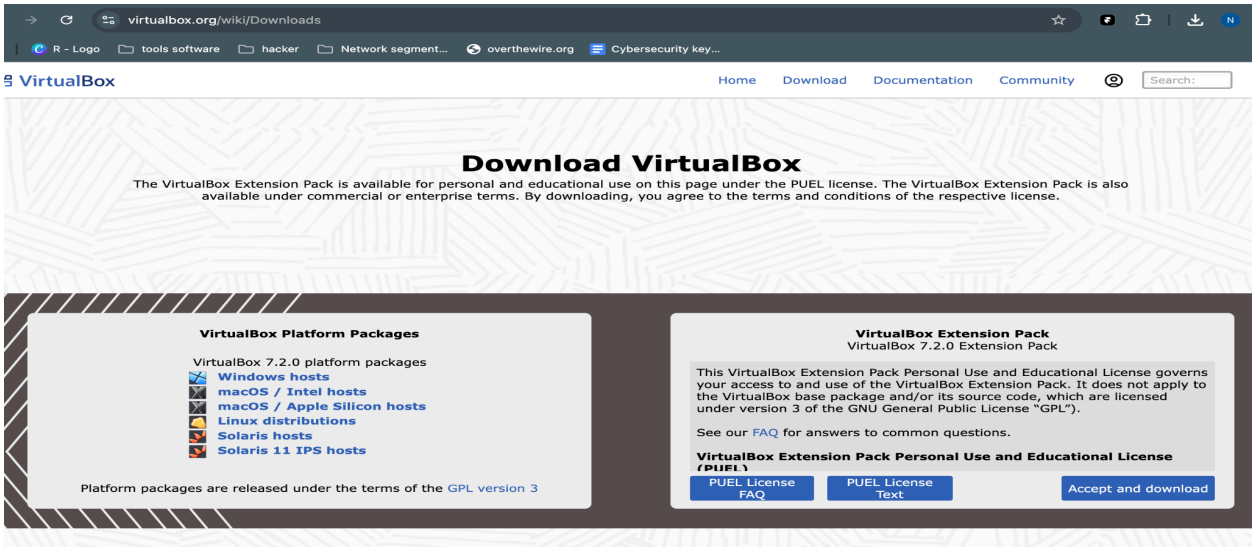
6. OpenSSH: Secures remote access with custom configurations like nonstandard ports and login attempt limits to reduce vulnerabilities

Table of Contents

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.

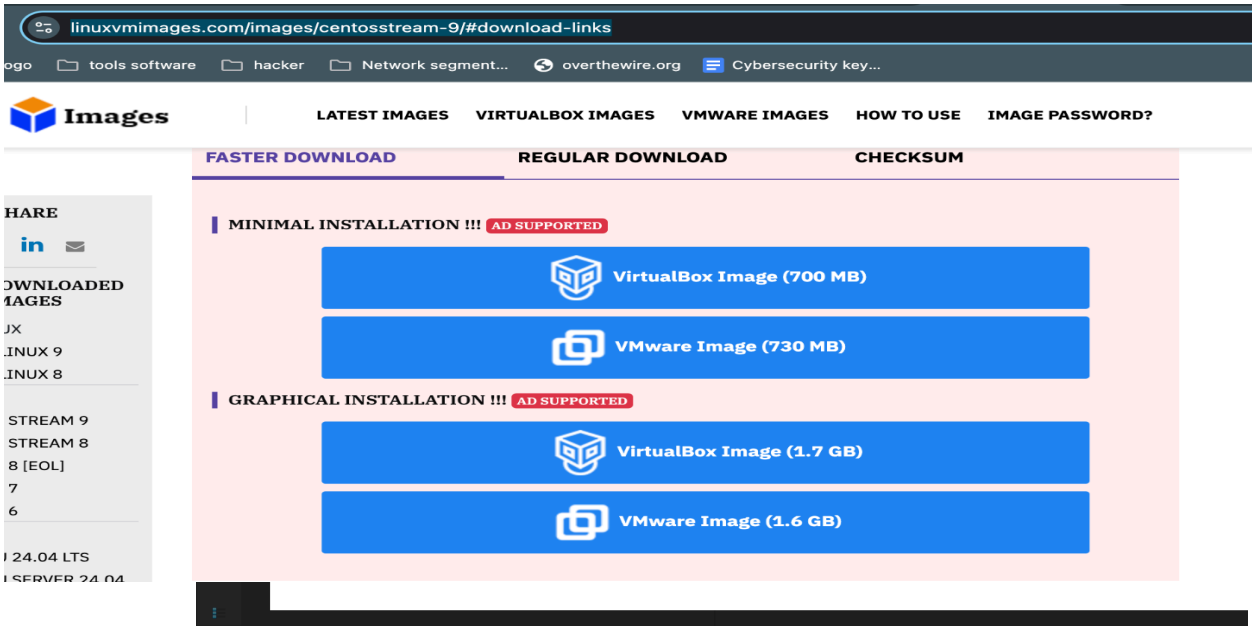


Select the package depending on your machine configuration.

- 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)

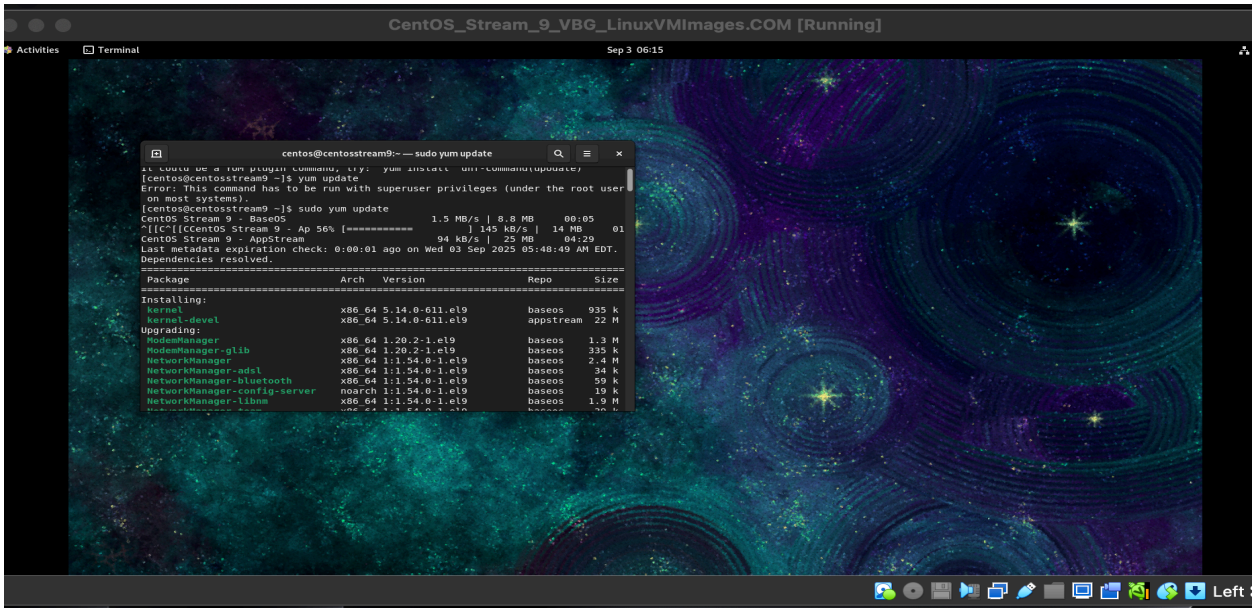
  - Uncompress the downloaded file.
  - Open the VirtualBox and click on the + (Open/Add) sign.

- Then you can browse to the folder containing the file



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

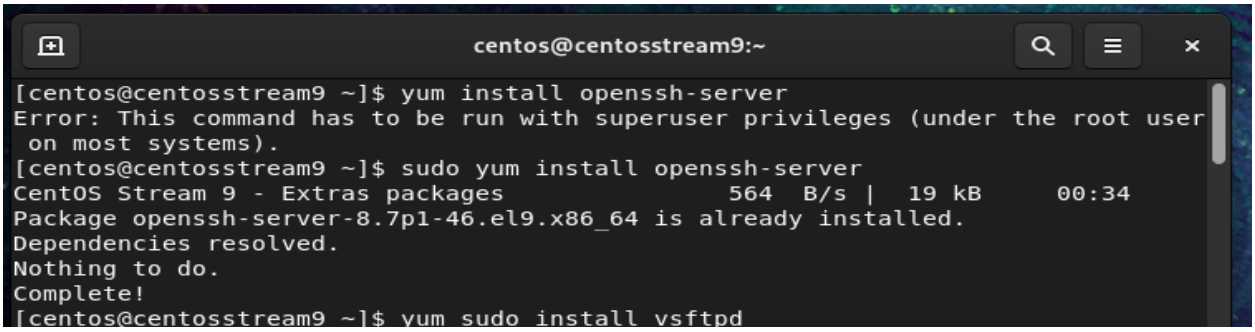
- Once Added → select the VM then click on Start (the green right arrow)
- Enter password (centos → common for al)
- Next click on activities → terminal → type command `sudo yum update`
- Whenever you are asked “do you want to install packages” type `y` and press enter
- Wait till all packages are installed


CentOS_Stream_9_VBG_LinuxVMImages.COM [Running]
Activities
Terminal
Sep 3 06:15
centos@centosstream9:~ — sudo yum update
[centos@centosstream9 ~]$ yum update
Error: This command has to be run with superuser privileges (under the root user on most systems).
[centos@centosstream9 ~]$ sudo yum update
CentOS Stream 9 - BaseOS 1.5 MB/s | 8.8 MB 00:05
CentOS Stream 9 - AppStream 94 kB/s | 25 MB 04:29
Last metadata expiration check: 0:00:01 ago on Wed 03 Sep 2025 05:48:49 AM EDT.
Dependencies resolved.
Package Arch Version Repo Size
Installing:
kernel x86_64 5.14.0-611.el9 baseos 935 k
kernel-devel x86_64 5.14.0-611.el9 appstream 22 M
Upgrading:
ModemManager x86_64 1.20.2-1.el9 baseos 1.3 M
ModemManager-glib x86_64 1.20.2-1.el9 baseos 335 k
NetworkManager x86_64 1:1.54.0-1.el9 baseos 2.4 M
NetworkManager-adsl x86_64 1:1.54.0-1.el9 baseos 34 k
NetworkManager-bluetooth x86_64 1:1.54.0-1.el9 baseos 59 k
NetworkManager-config-server noarch 1:1.54.0-1.el9 baseos 19 k
NetworkManager-libnm x86_64 1:1.54.0-1.el9 baseos 1.9 M

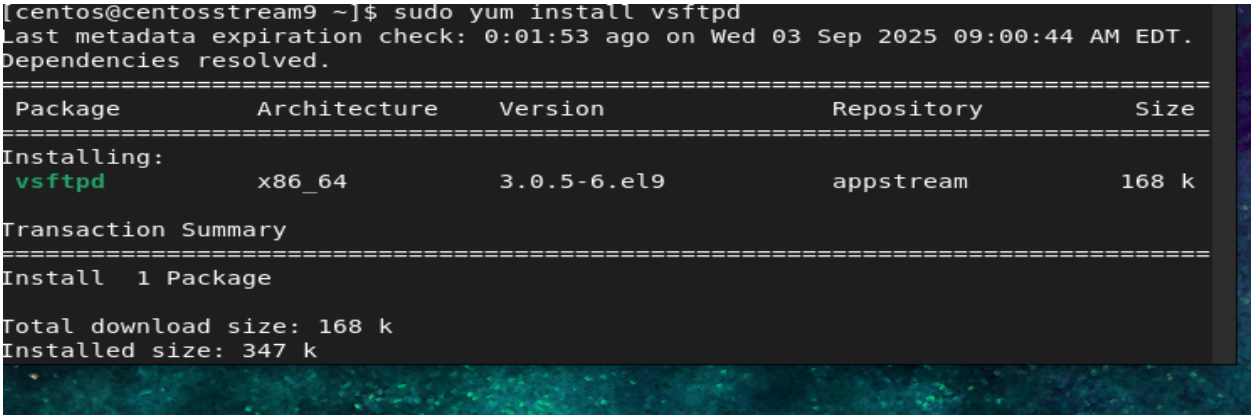
Honeypot Deployment

Step 1: Simulate Real Server Environment - Key Installations

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



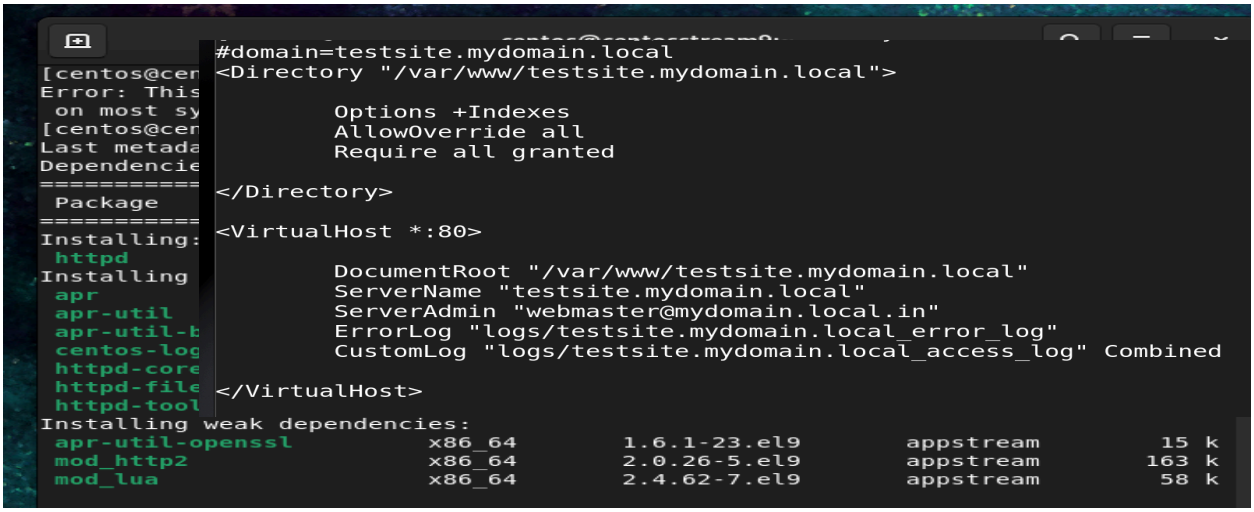
command on the terminal



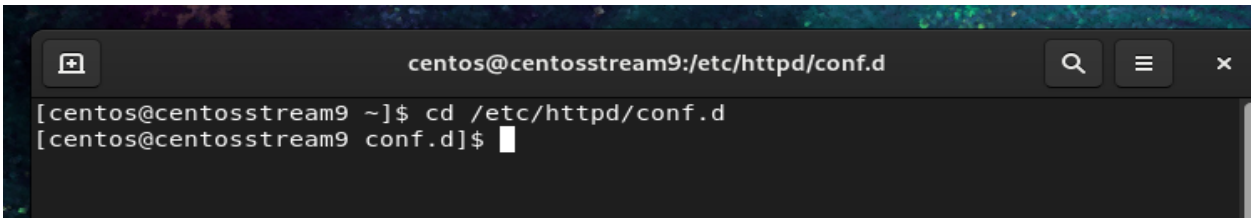
2. Install vsftpd. Use `sudo yum install vsftpd` command on the terminal

- 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

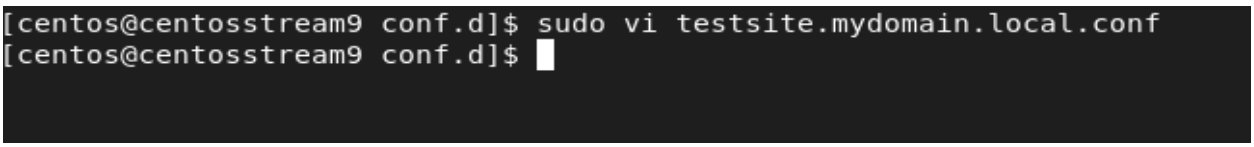


- This is the Script:

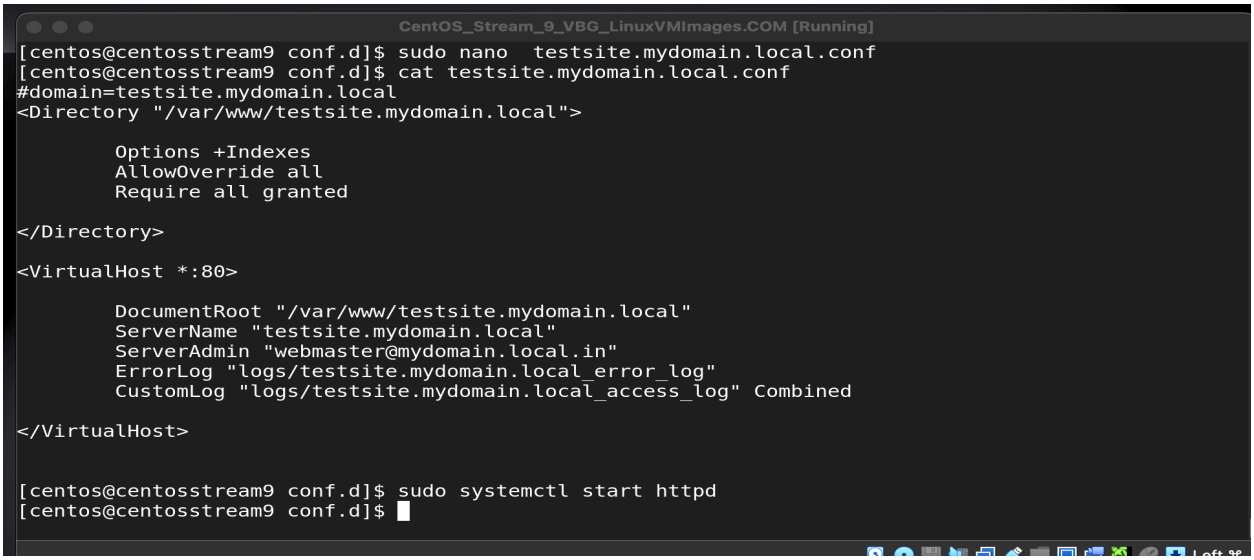


  - Navigate to the config file `cd /etc/httpd/conf.d`

  - 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`.

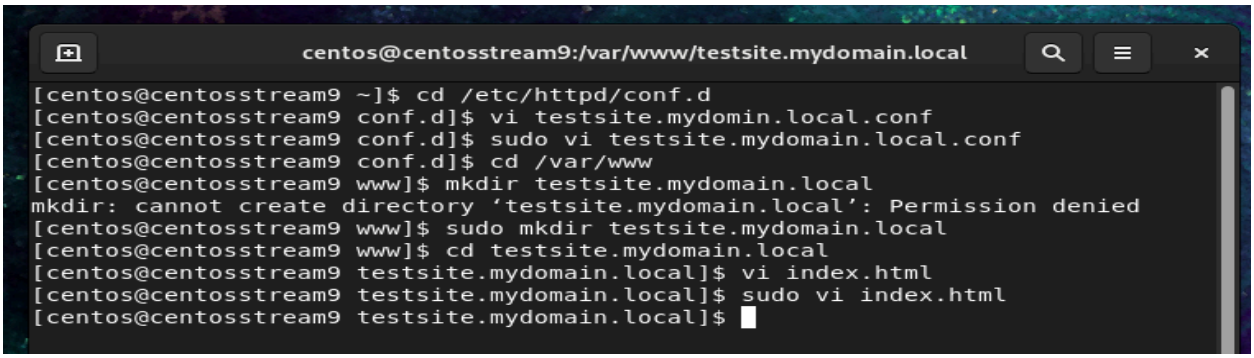


  - Press i to enter insert mode and paste the script in first step



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`



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

```
[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
```

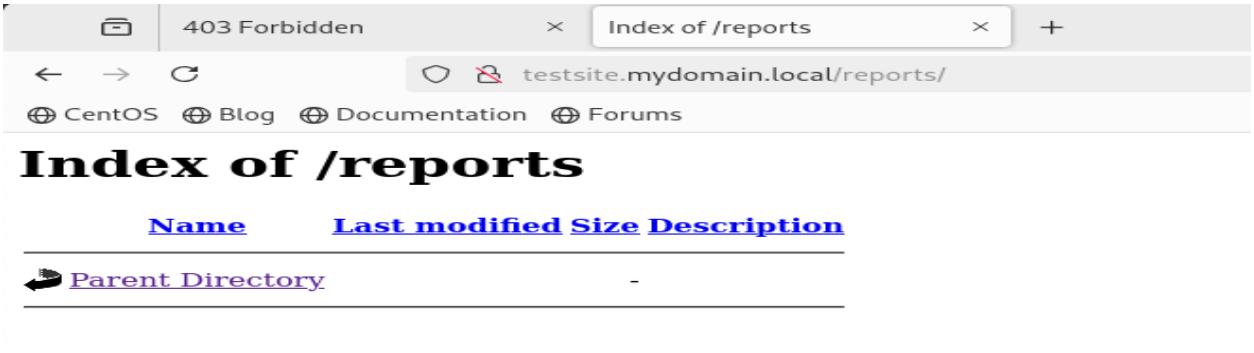
- Open website using the command `firefox testsite.mydomain.local`
- Navigate to the http://testsite.mydomain.local/reports

[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/`

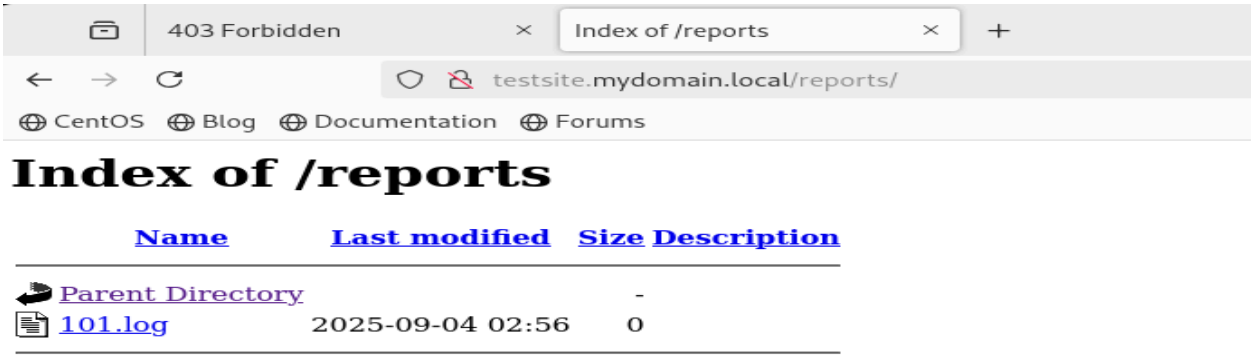
```
[centos@centosstream9 testsite.mydomain.local]$ sudo chcon unconfined_u:object_r:ht
tpd_sys_content_t:s0 /reports/
[centos@centosstream9 testsite.mydomain.local]$
```

- Now start firefox again and open /reports




```
[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



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

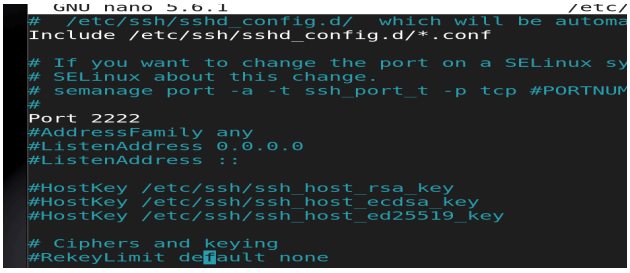
This we had installed already

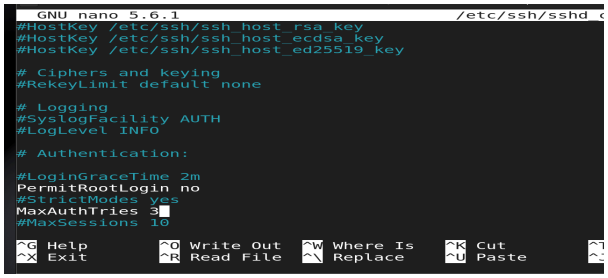
Step 2: Edit SSH Configuration

- Check current configuration by opening the file

  `sudo cat /etc/ssh/sshd_config`

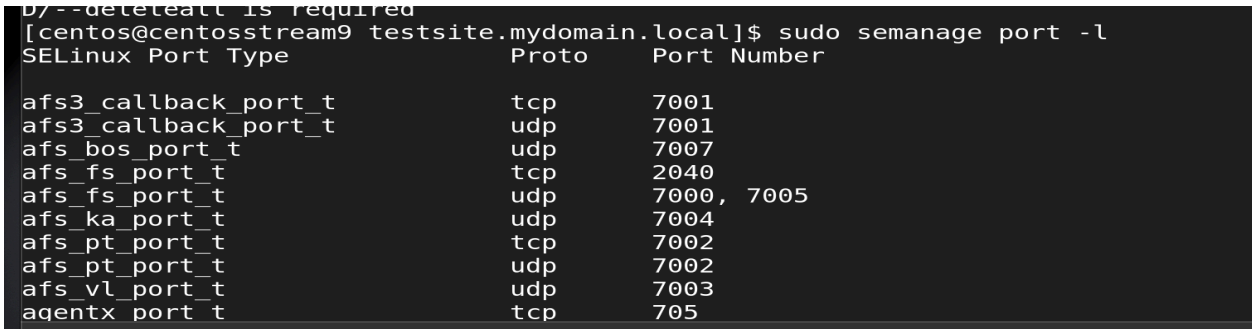
  Make the following changes for better security:
- Change default SSH port (e.g., from 22 to 2222)
  `Port 2222`
- Disable root login via SSH:
  `PermitRootLogin no`
- Disable password authentication (require key-based auth):
  `PasswordAuthentication no`
- Allow only specific users (optional):
  `AllowUsers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





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

- Check current SELinux SSH port labels:
  `sudo semanage port -l`



- 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:/var/www/testsite.mydomain.local — sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled;
     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
```

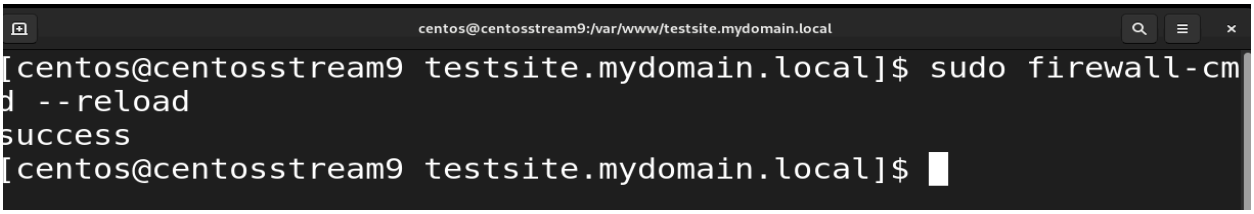
Step 5: Configure Firewall to Allow New Port

- 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
```



- 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

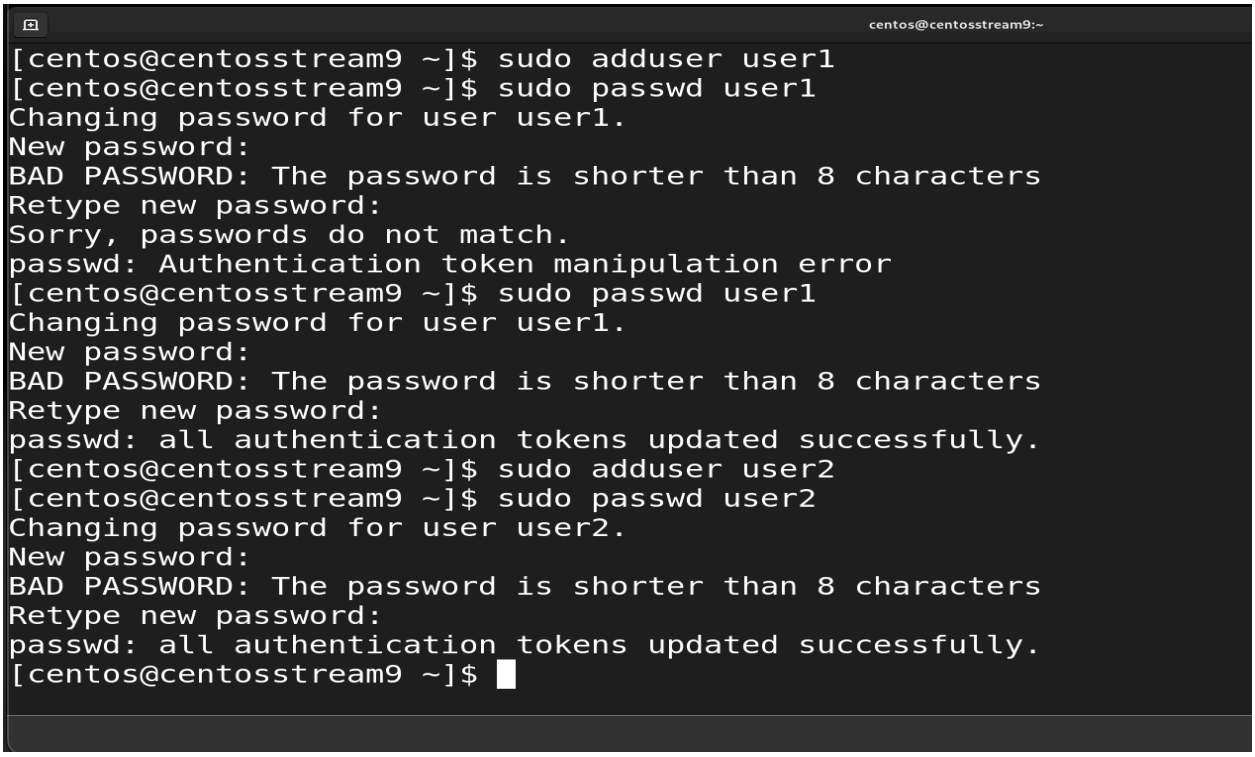
- To test the SSH Hardening lets first create some users - some who will be granted access in the config file and some can be denied access.

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

sudo adduser user2
sudo passwd user2
```

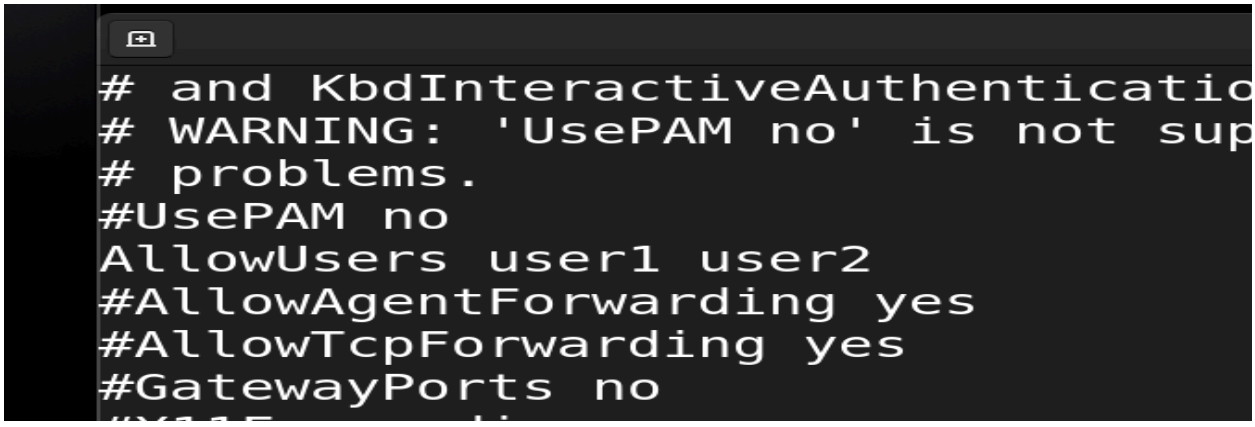
Edit SSH Config to Allow Only These Users

- Open ssh configuration file



`sudo nano /etc/ssh/sshd_config`

- Add the code
  `AllowUsers user1 user2`



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@teststite.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
```

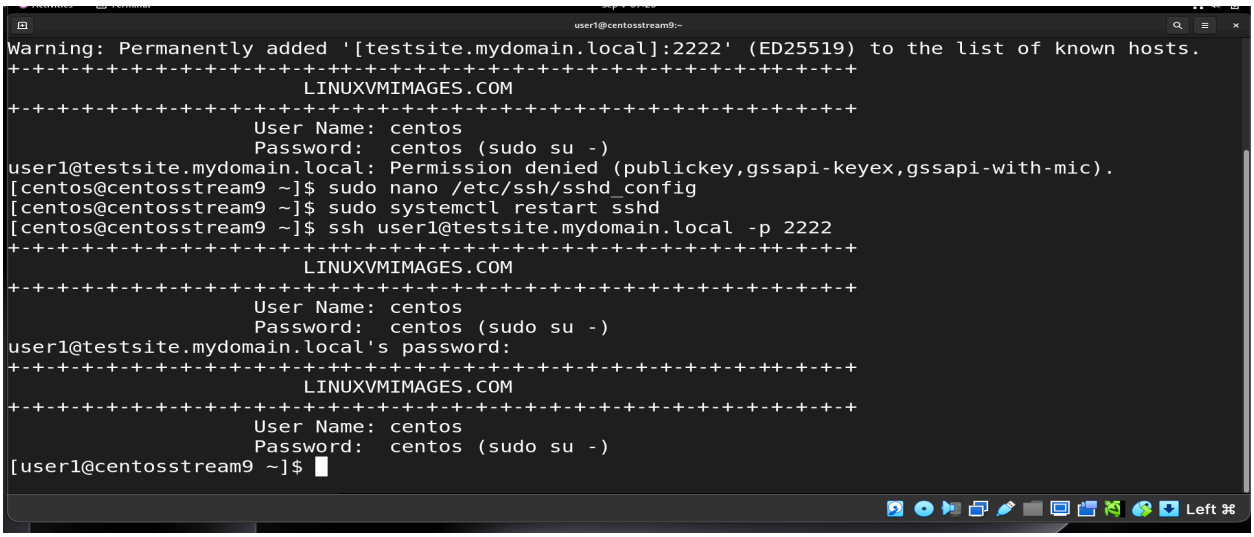
Comment out `PasswordAuthentication no` using `#`

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

- Now restart the ssh service again

- Then try again with same command
  `ssh user1@teststite.mydomain.local -p 2222`

It works.



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.

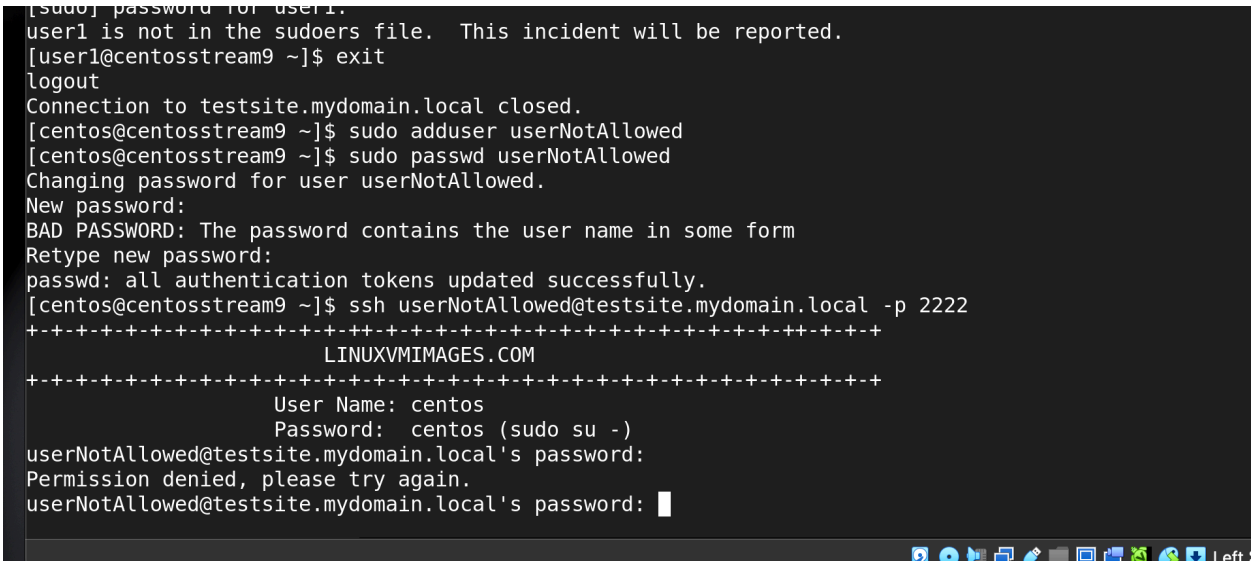
Points to note:

- 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

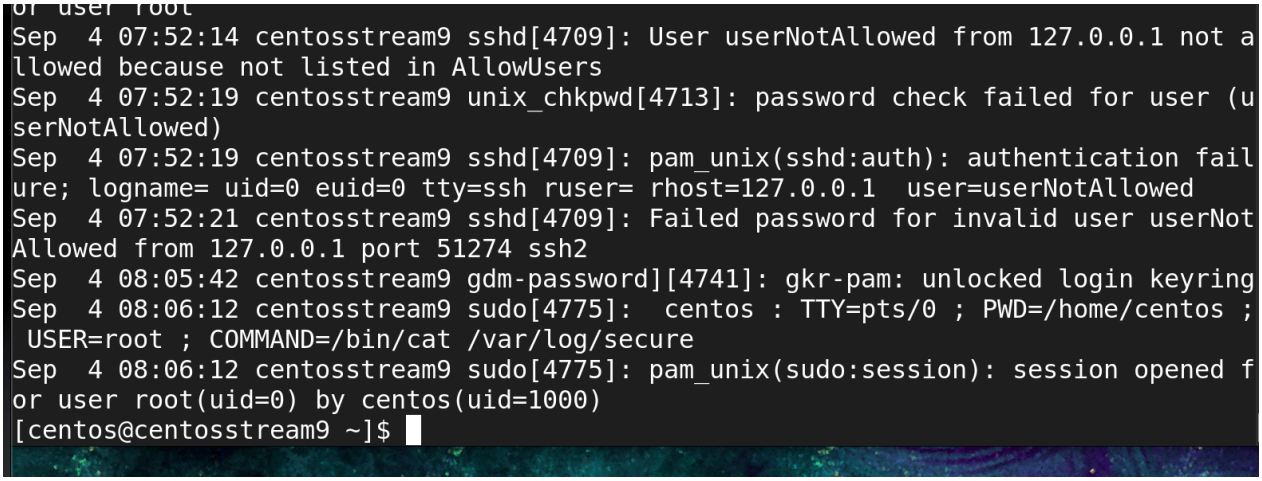
- Before adding new user
  - Exit from current logged in user
  - Use command `exit`
- Add a new user
  `sudo adduser userNotAllowed`
  `sudo passwd userNotAllowed`
- Next try to access testsite.mydomain.local
  `ssh unknownuser@your_server_ip -p 2222`
- All above commands are in the screenshot below:



The message says Permission Denied on the terminal.

Check Logs for The Failed Authentication

`sudo cat /var/log/secure`

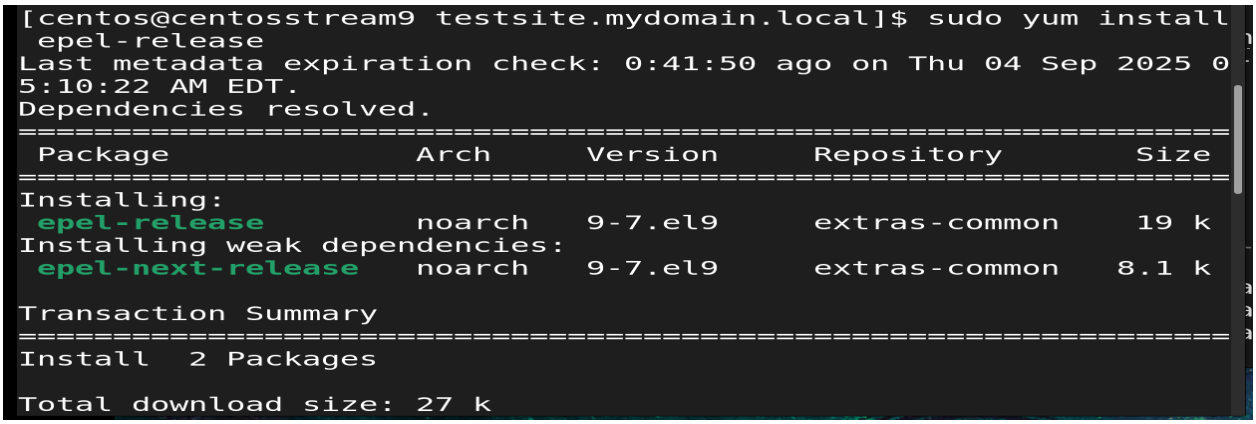

of user root
Sep 4 07:52:14 centosstream9 sshd[4709]: User userNotAllowed from 127.0.0.1 not allowed because not listed in AllowUsers
Sep 4 07:52:19 centosstream9 unix_chkpwd[4713]: password check failed for user (userNotAllowed)
Sep 4 07:52:19 centosstream9 sshd[4709]: pam_unix(sshd:auth): authentication 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`



`sudo yum install fail2ban`

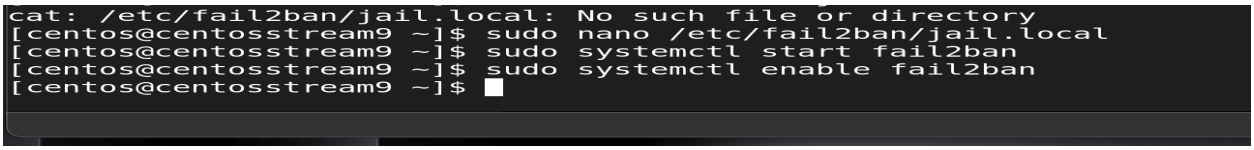
Create / edit jail.local

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

Add the code like below in the editor:

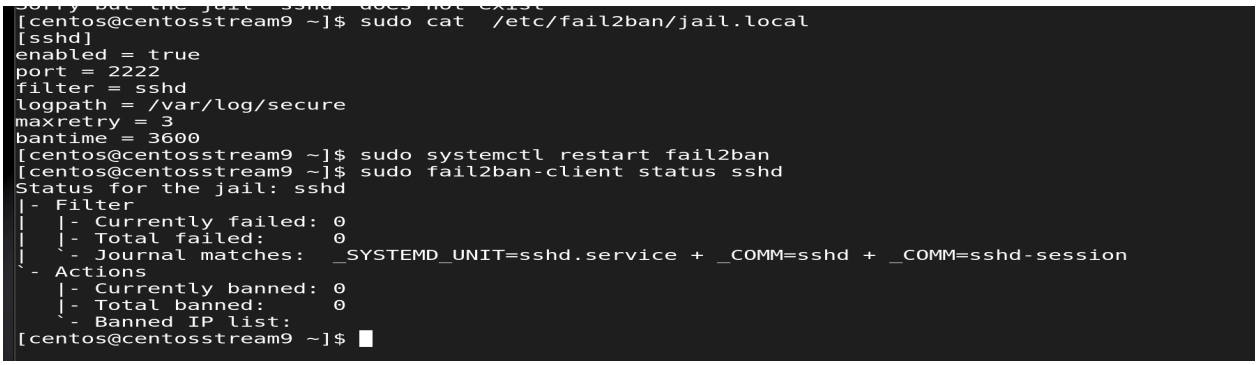
```
[sshd]
enabled = true
port = 2222
filter = sshd
logpath = /var/log/secure
maxretry = 3
bantime = 3600
```

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



Start and enable the fail2ban

Verify jail status
`sudo fail2ban-client status sshd`



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: e
     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 --nopid

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)

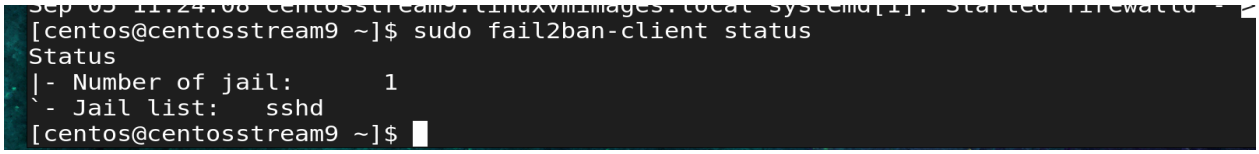
Step 3: Configure Fail2Ban to Use Firewallcmd for Active Banning

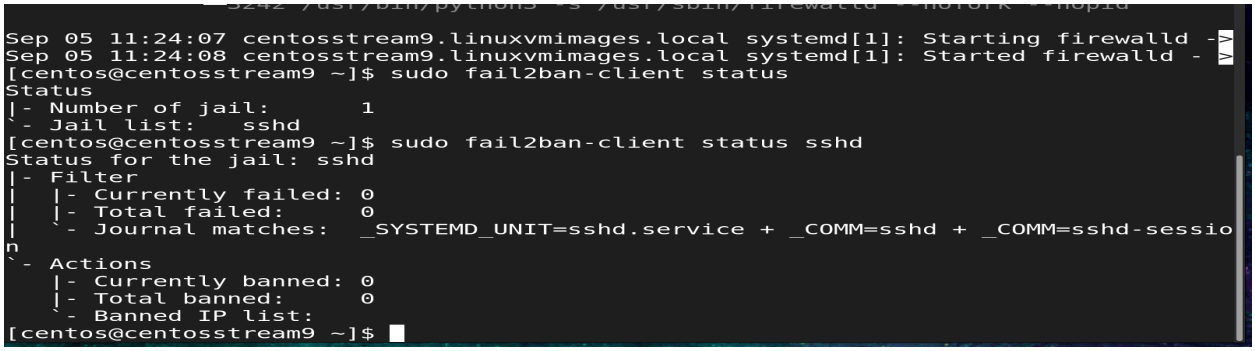
- Set `action = firewallcmd-ipset` in /etc/fail2ban/jail.local

- Restart Fail2ban to activate new settings

- Check status of fail2ban enabled jails and ssh jail

  `sudo fail2ban-client status`

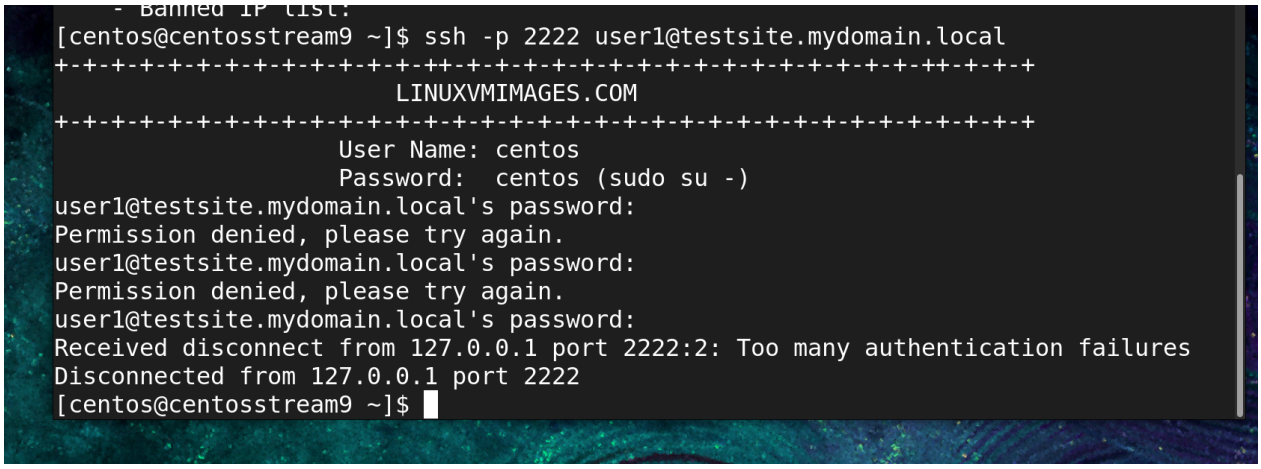




`sudo fail2ban-client status sshd`

- Try wrong password (more than 3 times)

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



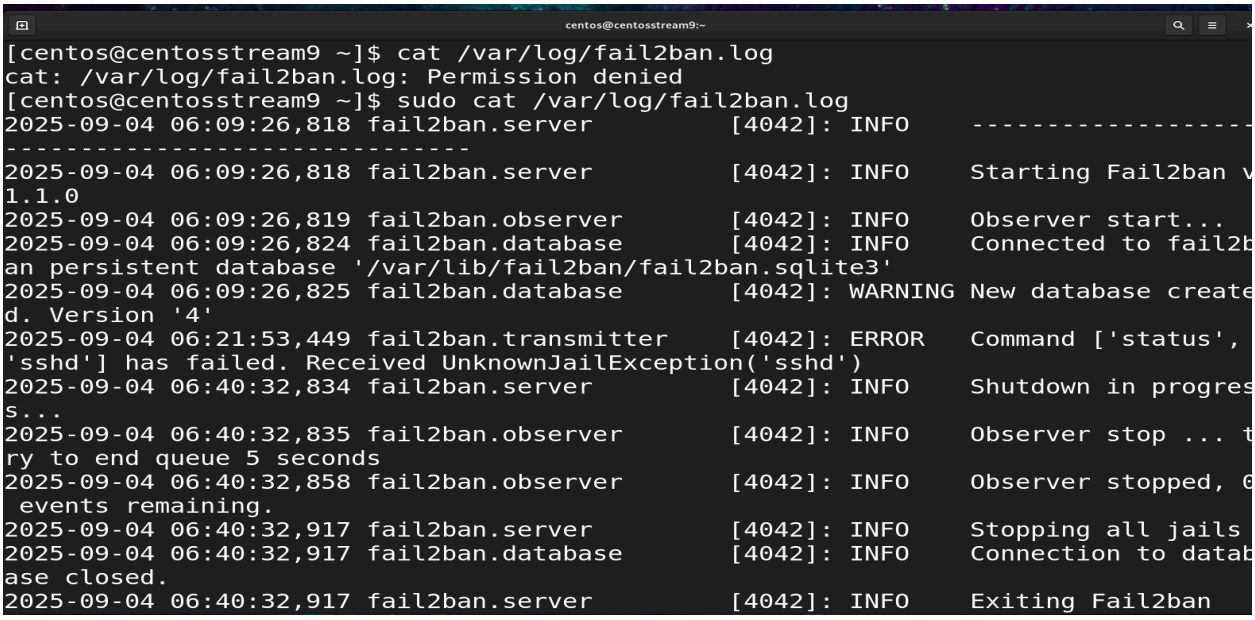
- Confirm Fail2Ban jail status again and it has blocked ip
  `sudo tail -n 50 /var/log/fail2ban.log`

- 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 dhcpv6-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
stemd {}
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
.0.1 by ignoreself rule
2025-09-05 11:42:34,118 fail2ban.filter         [3272]: INFO    [sshd] Ignore 127.0
.0.1 by ignoreself rule
2025-09-05 11:42:40,105 fail2ban.filter         [3272]: INFO    [sshd] Ignore 127.0
.0.1 by ignoreself rule
2025-09-05 11:42:41,368 fail2ban.filter         [3272]: INFO    [sshd] Ignore 127.0
.0.1 by ignoreself rule
[centos@centosstream9 ~]$
```

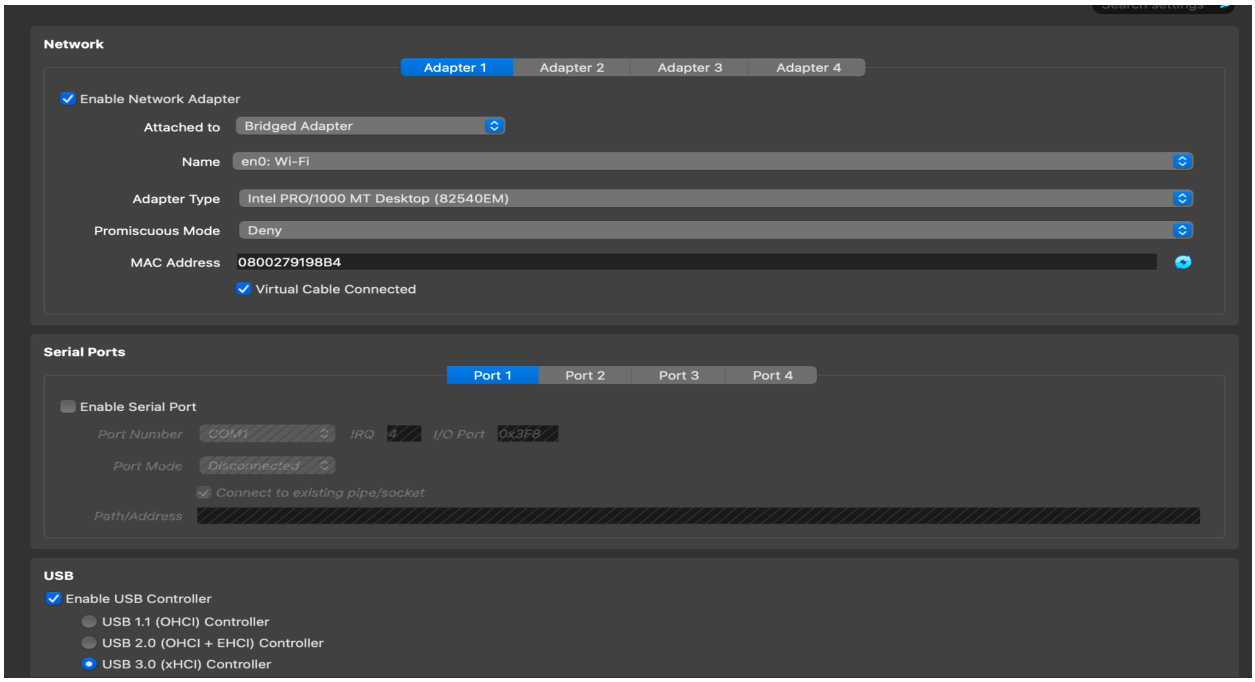
- We can also check fail2ban logs using `sudo cat /var/log/fail2ban.log`
- Log Analysis: Access is denied. “ignore self rule” means the server is protected from accidentally banning itself, and Fail2ban is operating correctly. We can now confidently simulate brute force or failed logins from an external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



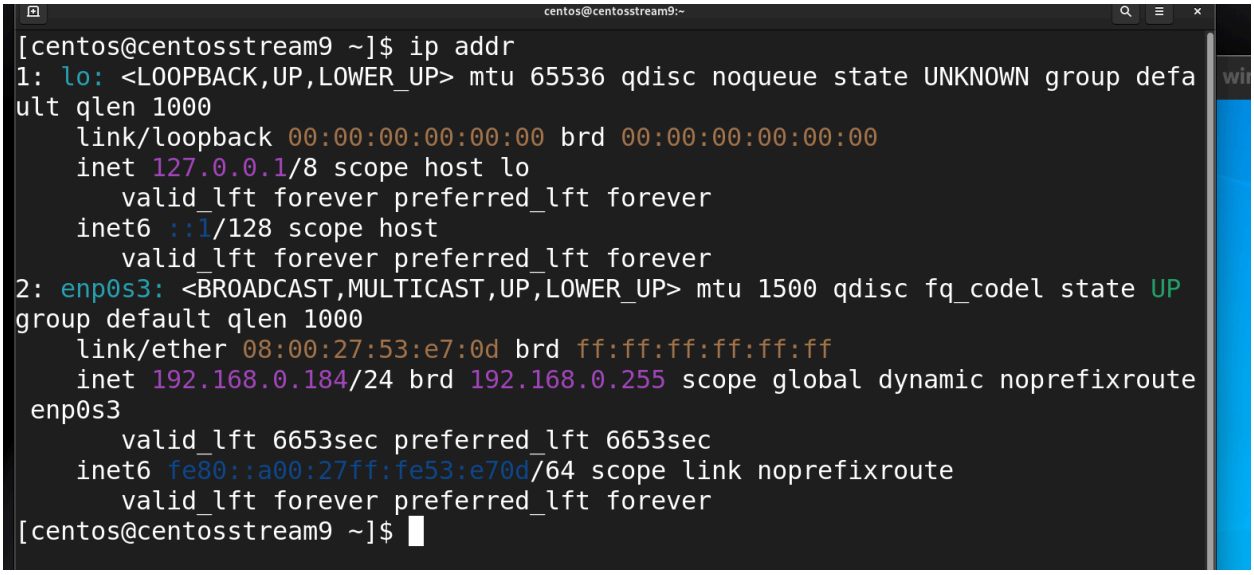
Testing: Manual Attack

Step 1: Setup

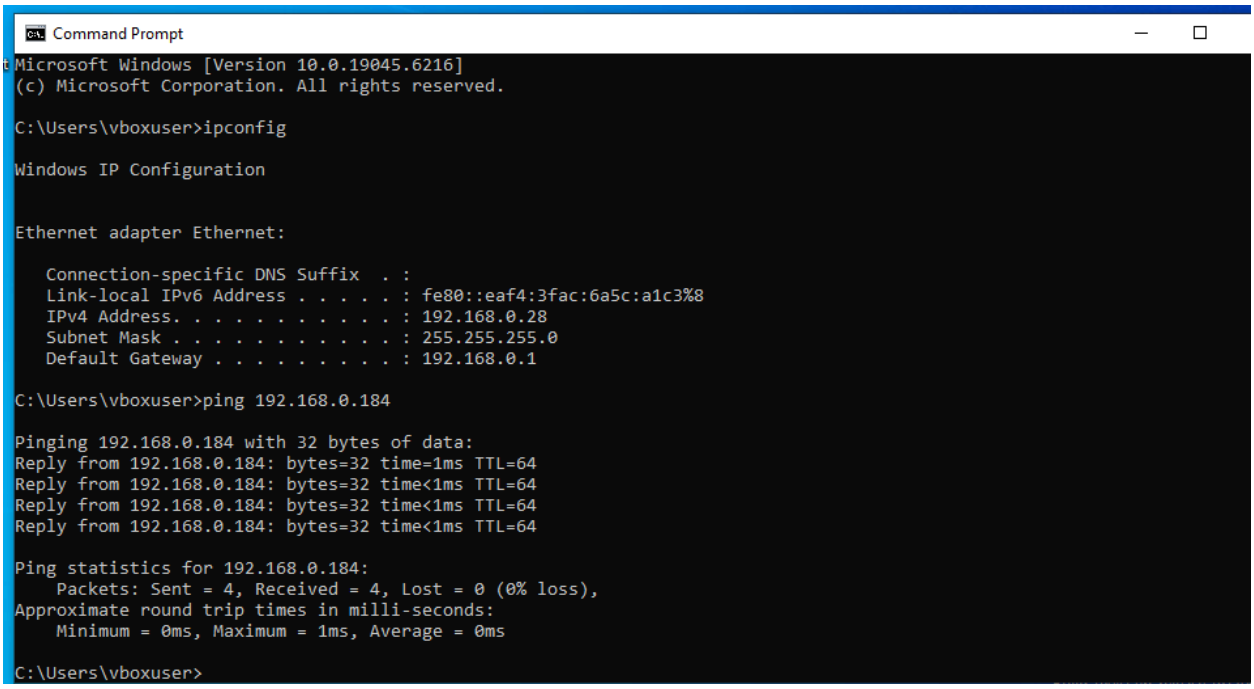
- Switch off the VM in virtual box



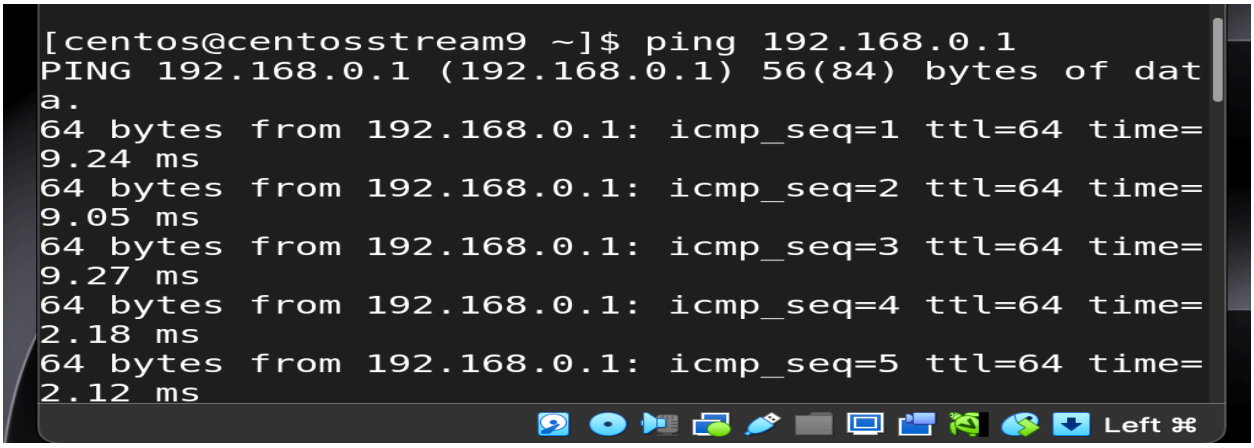
- Configure Network in VM settings to Bridged Adapter, check Virtual Cable and Promiscuous mode must be Denied
- Next, find out IP of the VM now both in which we have honeypot (centOS) and the other which will act as attacker
- To know Window OS device IP → We open command prompt → ipconfig → press enter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



- Ping the cent os or honeypot ip from attacker

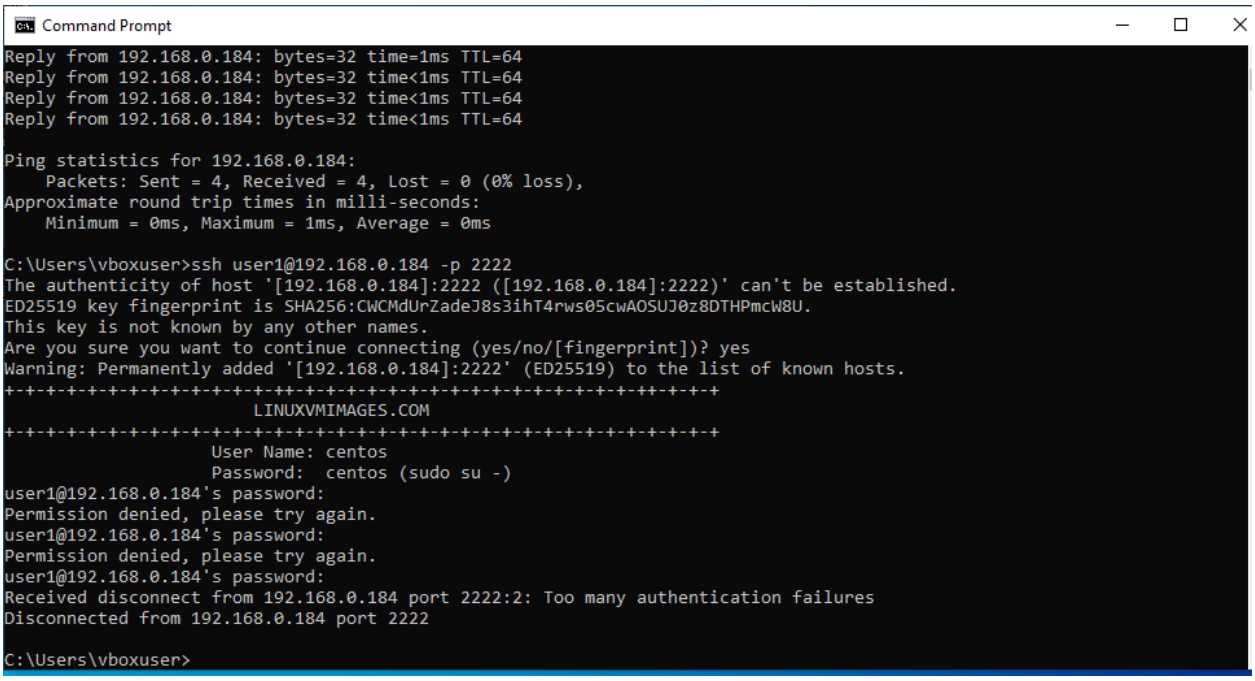


- Ping from centOS / honeypot to windows to confirm bilateral communication

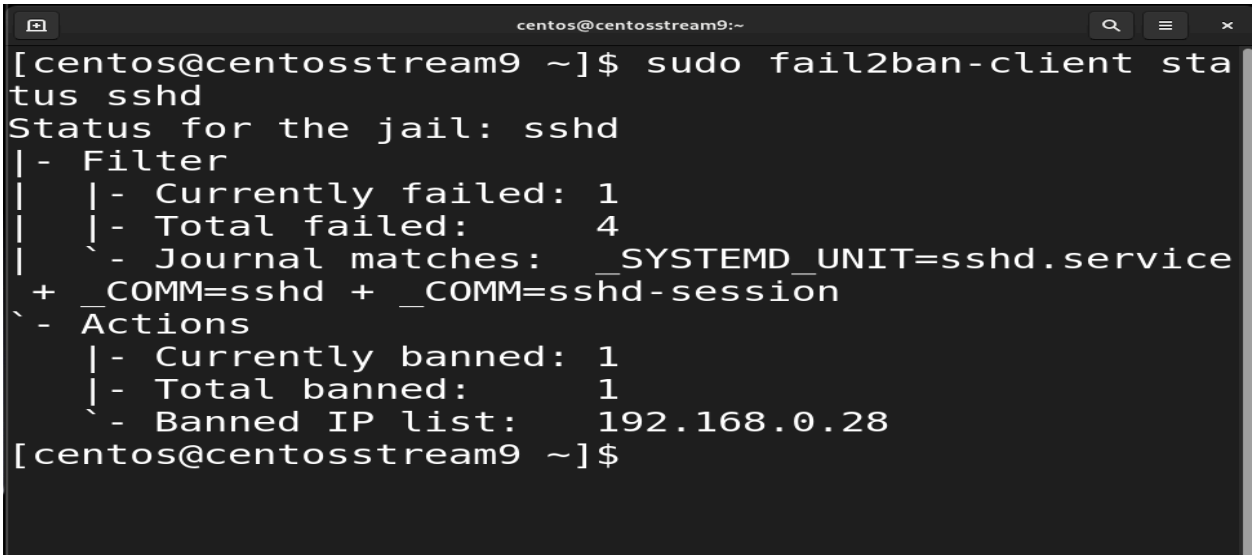


Step 2: Attempt Brute force Attack

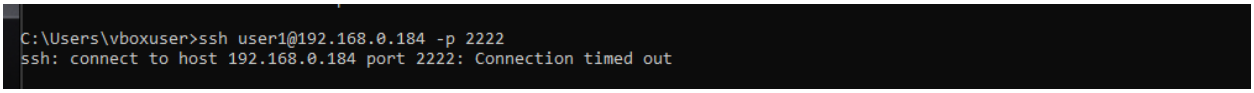
- Now since we have devices ready and can communicate with each other. We now launch our attack from windows OS
- Type command `ssh user1@192.168.0.184 -p 2222`
- Type random password multiple times until it says “Too many authentication failures”



- 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



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



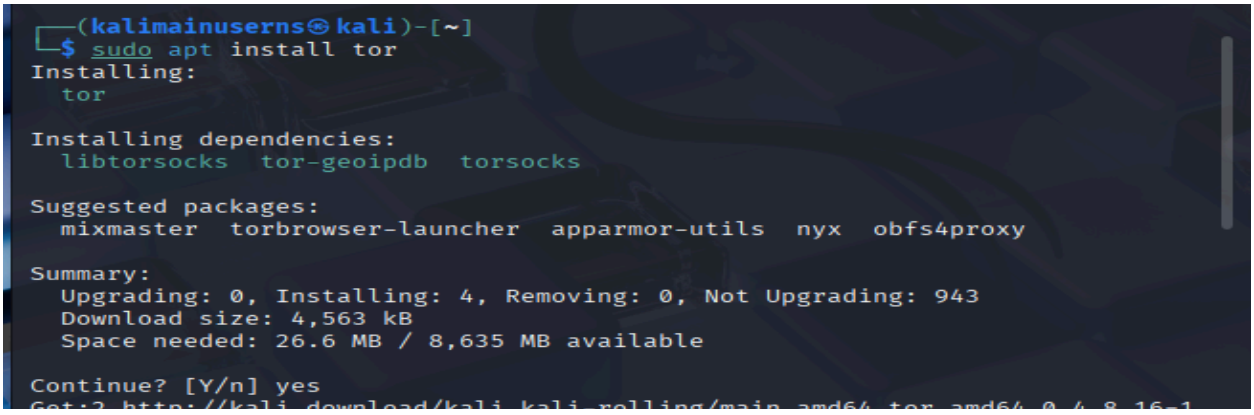
Penetration Testing Through Proxychains and Tor

Setup Kali Linux in your VM (guide)

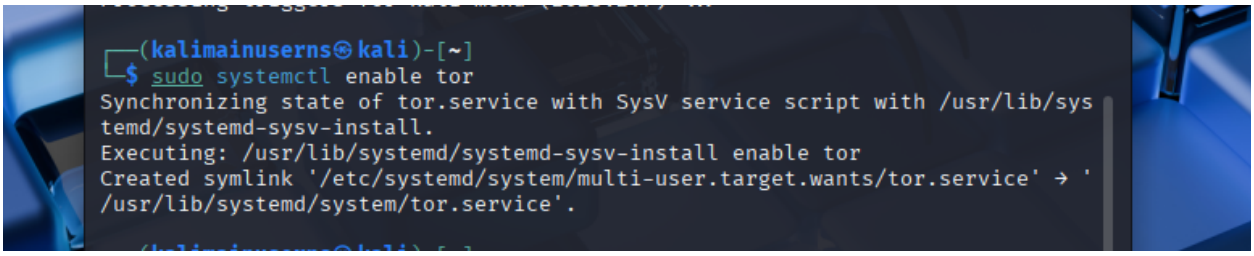
Step 1: Configure Tor on Kali Linux

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

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

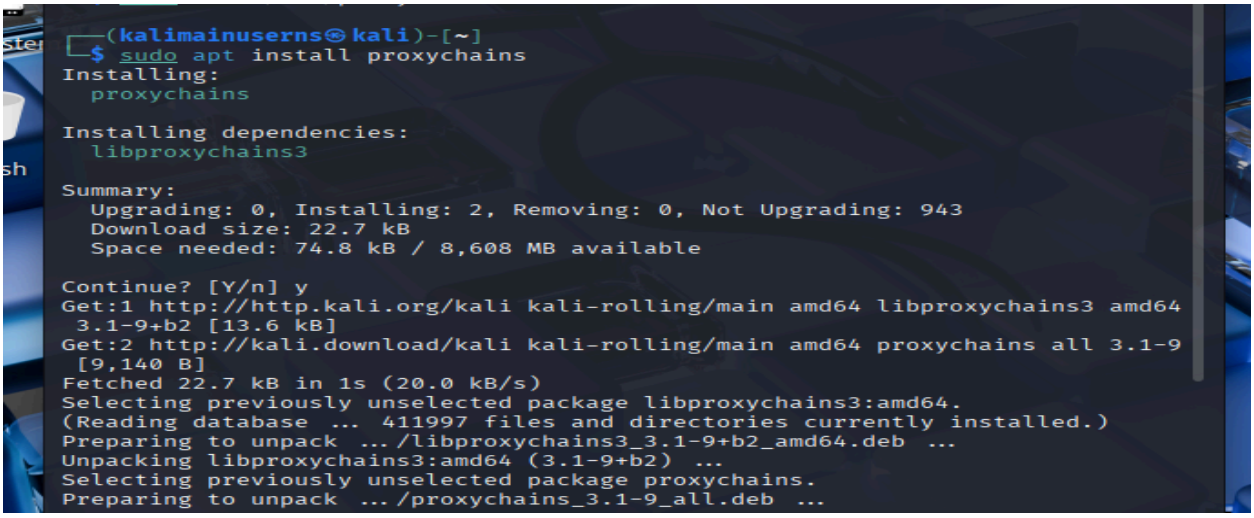


- Enable Tor to start on boot, `sudo systemctl enable tor`

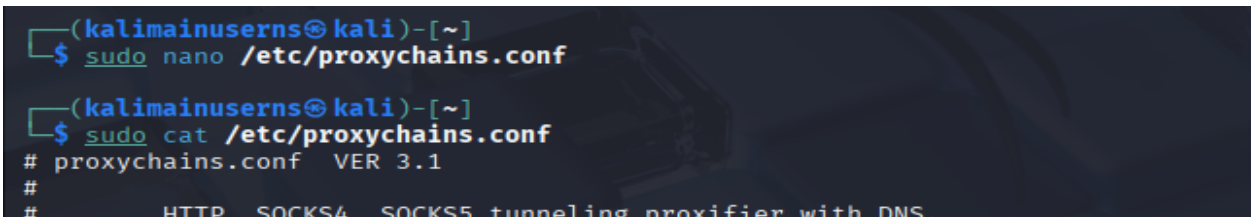


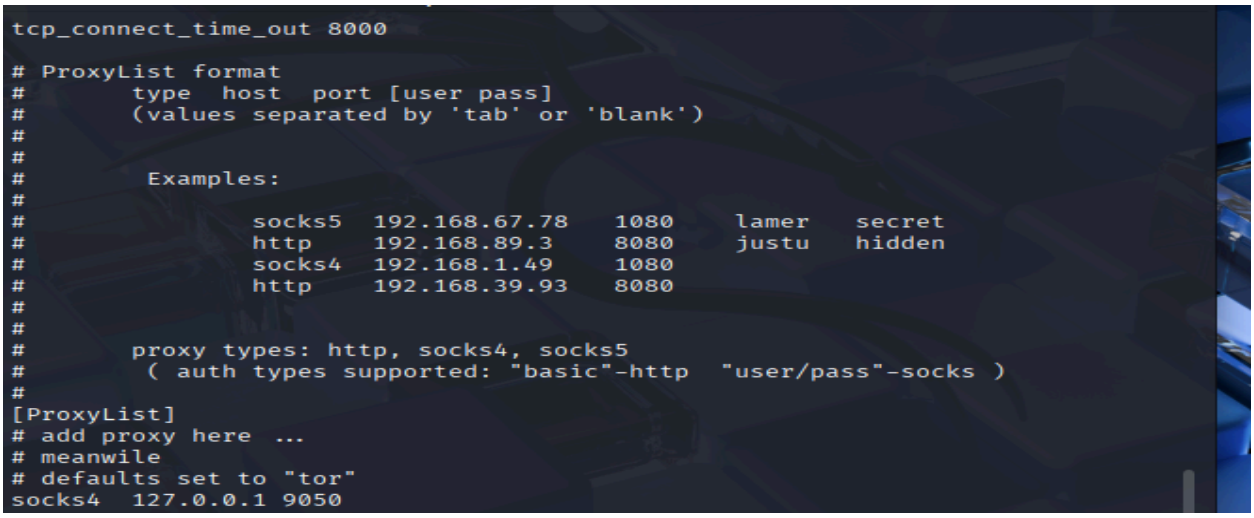
Step 2: Configure Proxychains to Use Tor

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



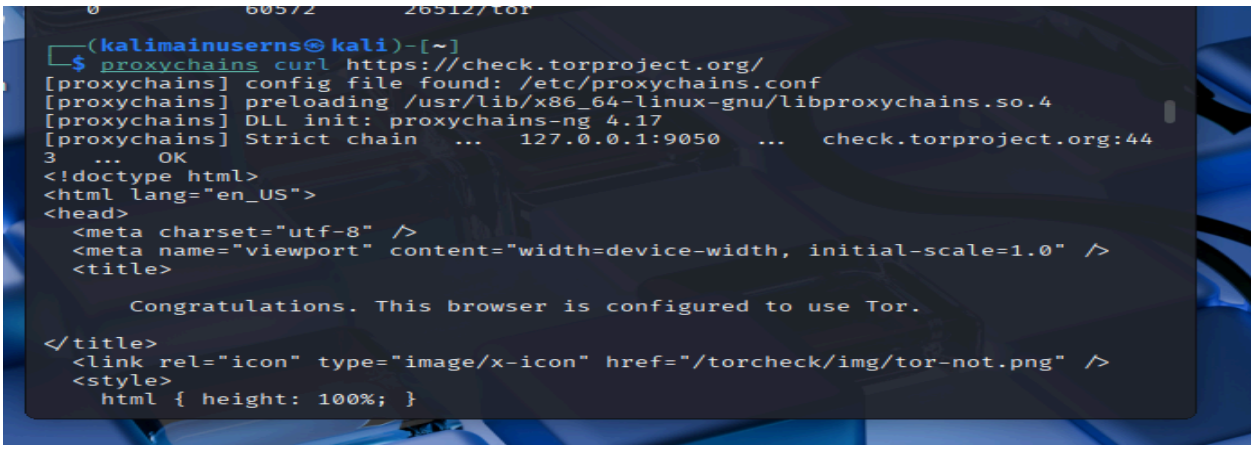
- Open the proxychains configuration file using command `sudo nano /etc/proxychains.conf`
- Scroll to the bottom, find the line that defines your proxy list, and ensure it includes the Tor Socks4 / socks5 proxy(if doesn't exist:
  `socks4  127.0.0.1 9050` or `socks5  127.0.0.1 9050`
- Save and exit (in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





Test Proxychains with Tor

- Test if proxychains and Tor are working by running proxychains `sudo curl https://check.torproject.org/`



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.

Prepare Username List

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

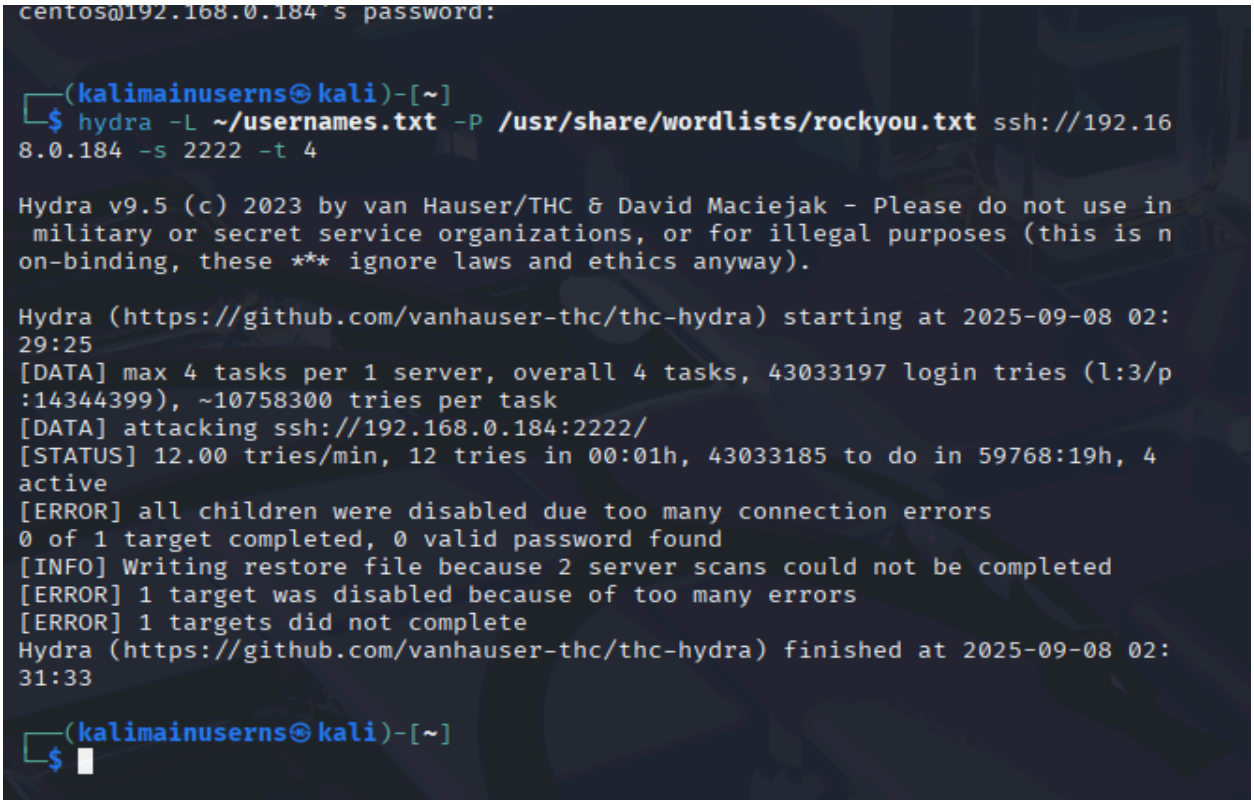
- Identify Your Target (`IP: 192.168.0.184, Port: 2222`)

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`

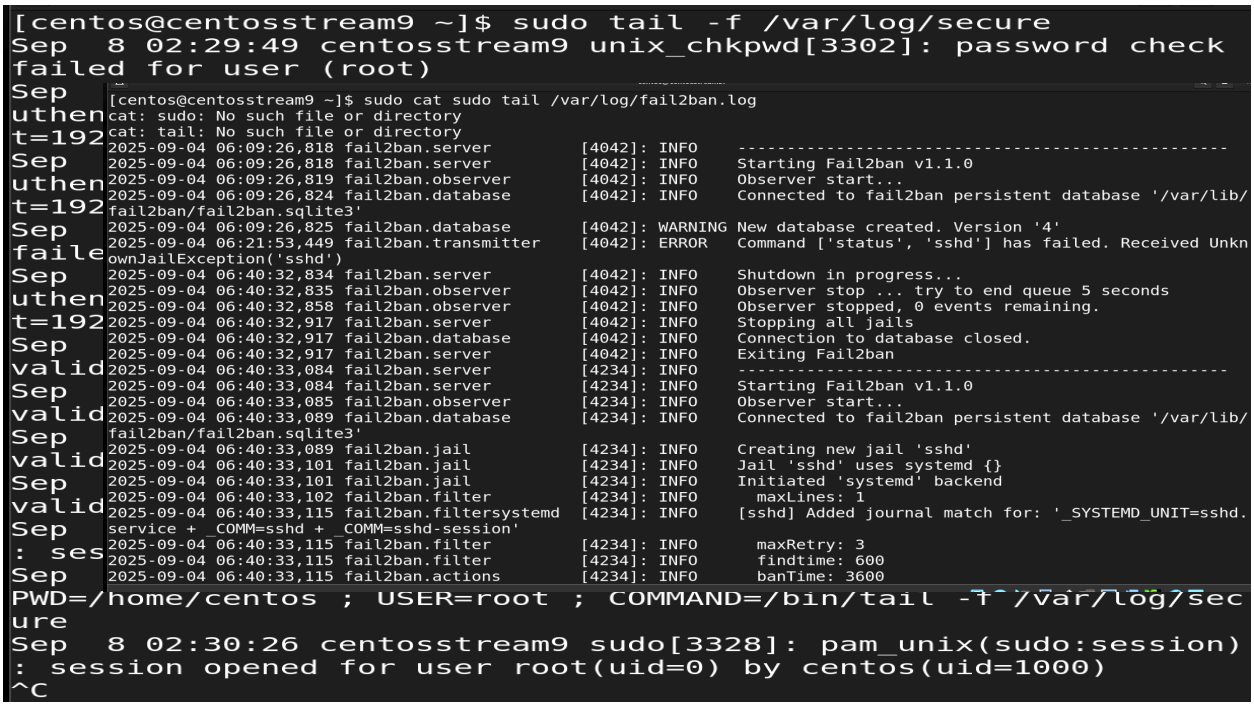


- Once hydra started attacking, you can check logs using `sudo tail -f /var/log/secure`
- Also check logs in fail2ban `sudo tail -f /var/log/fail2ban.log` and also `sudo journalctl -u fail2ban -f`

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).



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

```
2025-09-08 02:04:28,448 fail2ban.filtersystemd [1183]: INFO    [sshd] Jail is in operation now (process new journa
 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

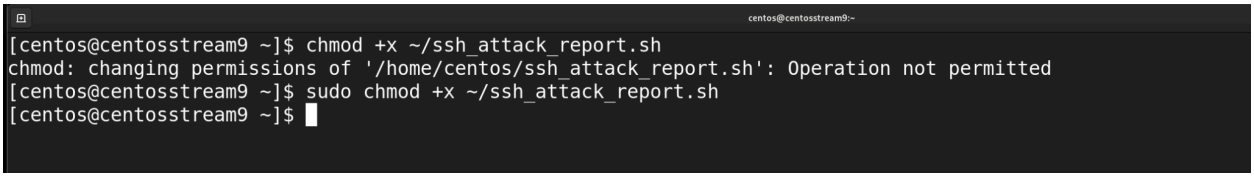
Create a file called `ssh_attack_report.sh` in home directory with the script below (in the screenshot)

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`

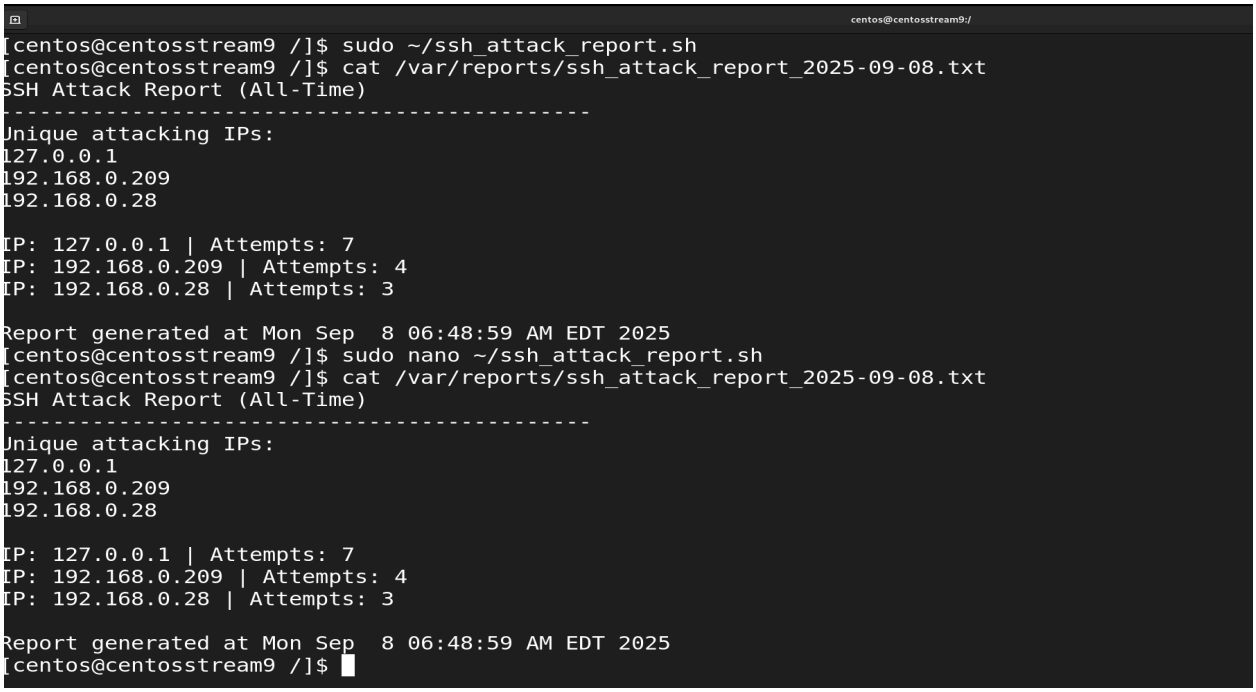


Step 4: Run and Test the Script

Run: `sudo ~/ssh_attack_report.sh`

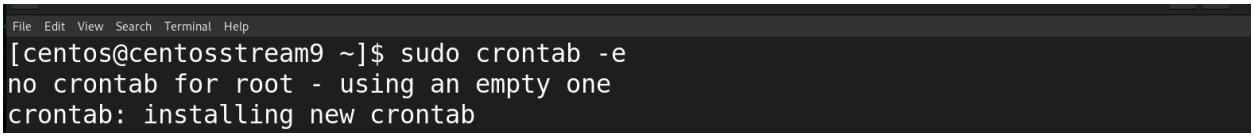
Test: `cat /var/reports/ssh_attack_2025-09-08.txt`

Below screenshot shows the result of the commands above:



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

How to Schedule the Script with Cron

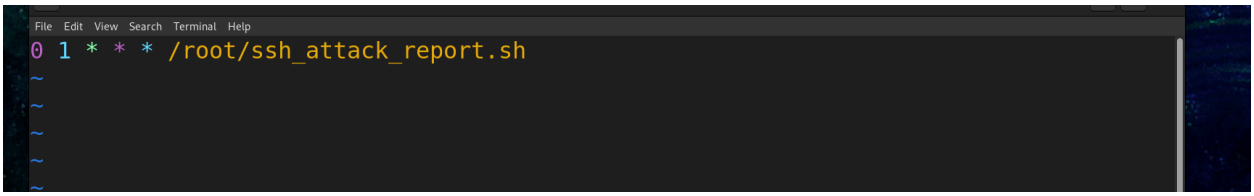


Edit the Root User’s Crontab and Launch crontab `sudo crontab -e`

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`



Verification

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

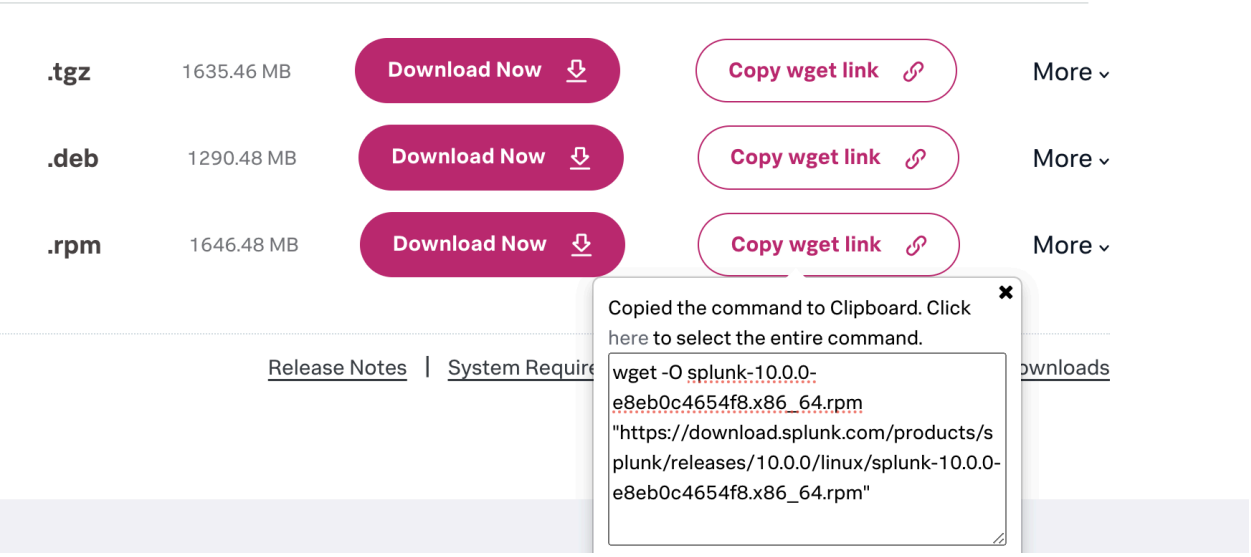
You can also check if the cron job ran by looking in cron logs:

`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: abouthisps: command not found...
[centos@centosstream9 ~]$ wget -O splunk-10.0.0-e8eb0c4654f8.x86_64.rpm "https://do
wnload.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... connect
ed.
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 [17
26454835/1726454835]
```

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

```
splunk:
[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 (requi
red 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:
```

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

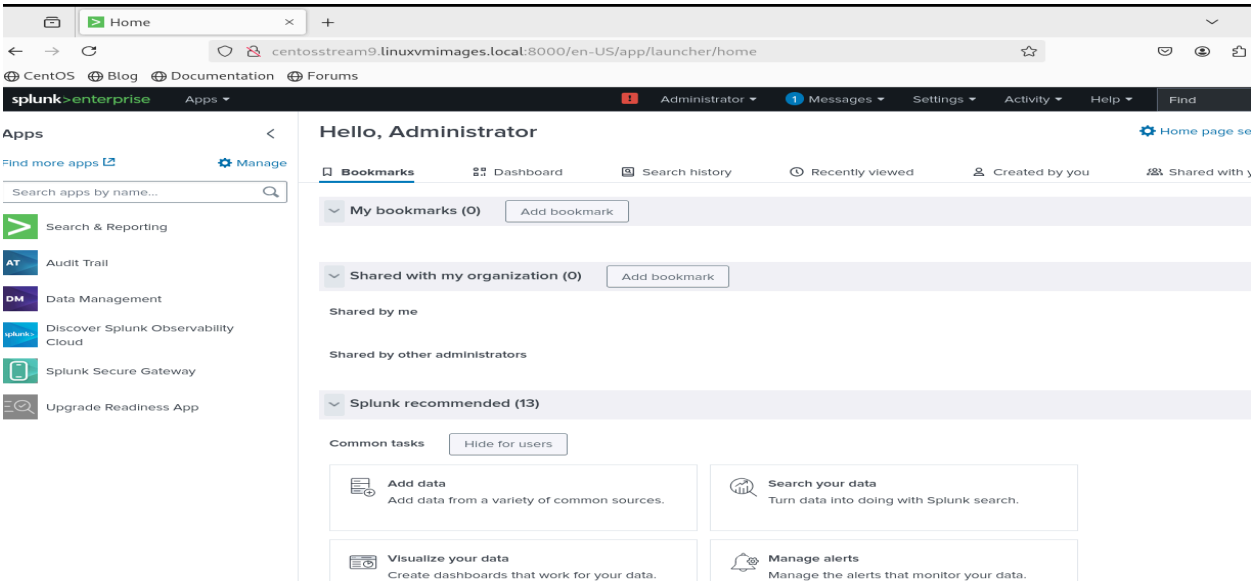
```
..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Warning: ignoring -extensions option without -extfile
Certificate request self-signature ok
subject=CN = centosstream9.linuxvmimages.local, O = SplunkUser
pDone


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
```

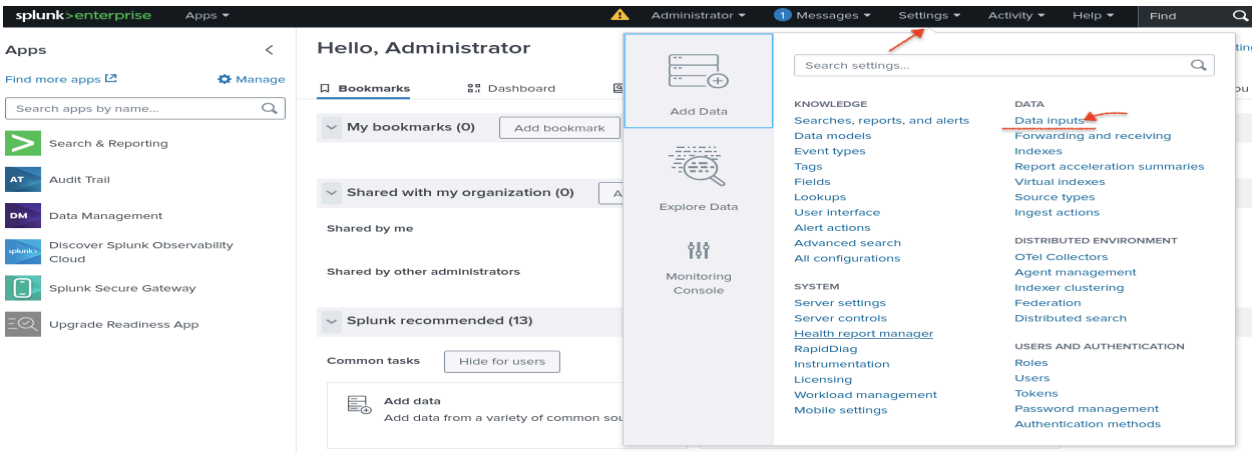
- Dashboard will open after login

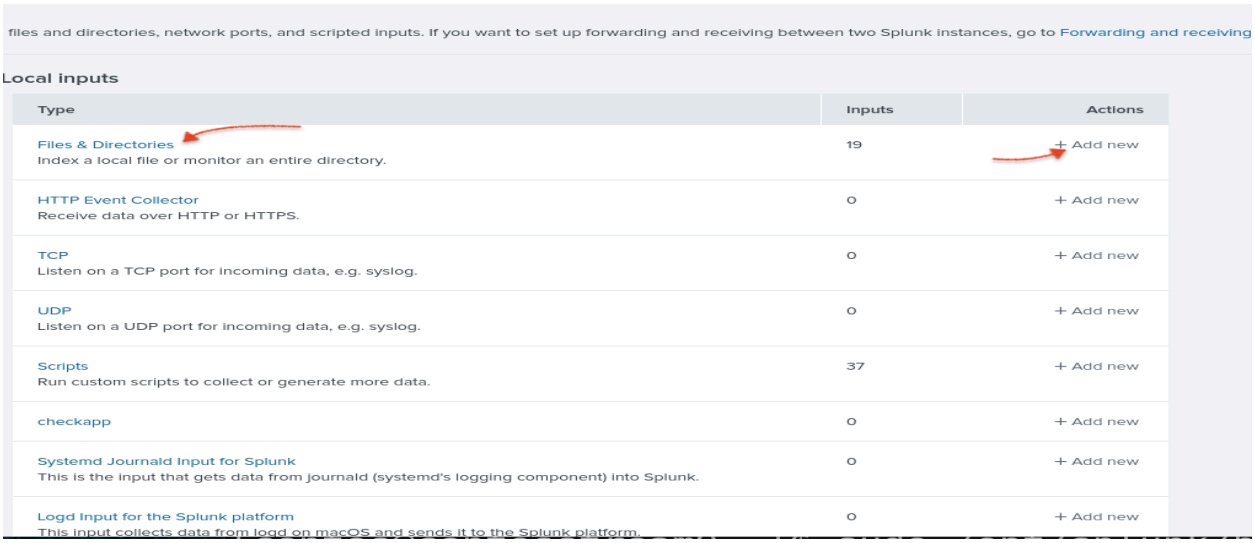


Step 3: Configure Data Inputs for Critical Logs

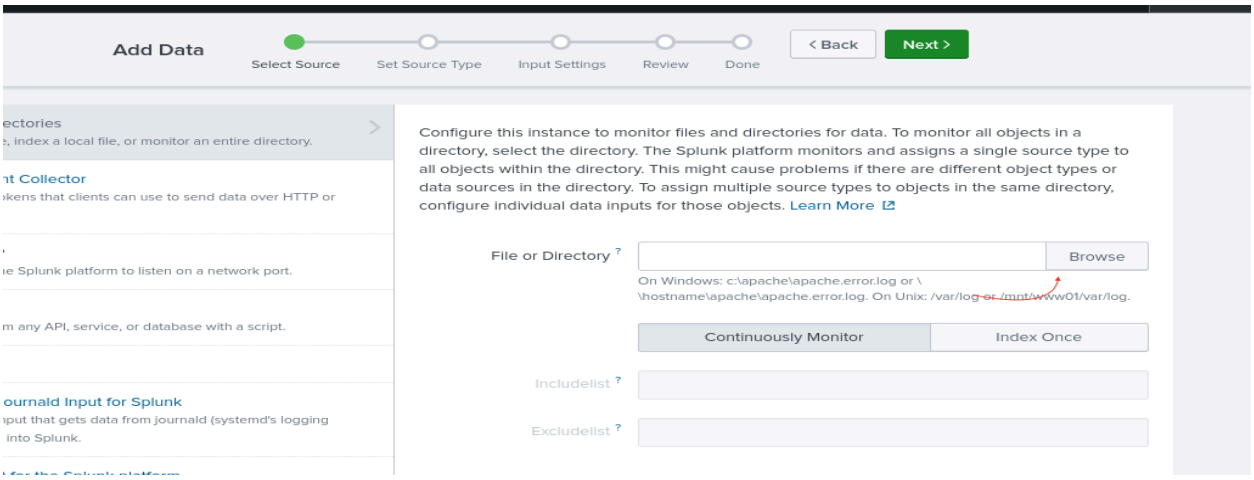
- From the Splunk Web UI, go to `Settings > Data Inputs > Files & Directories.`

- 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.

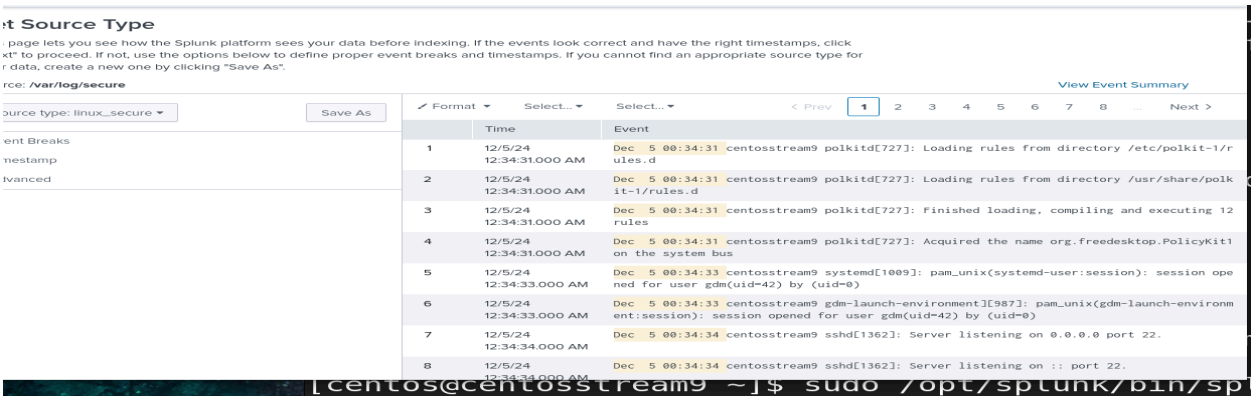


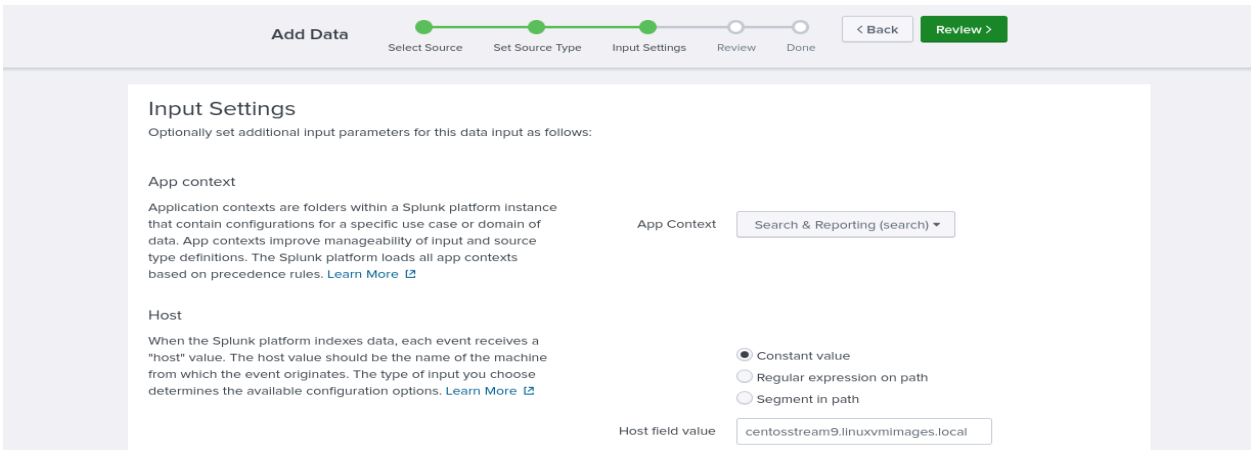


- Select source by browsing or typing the address directly and press `next` button

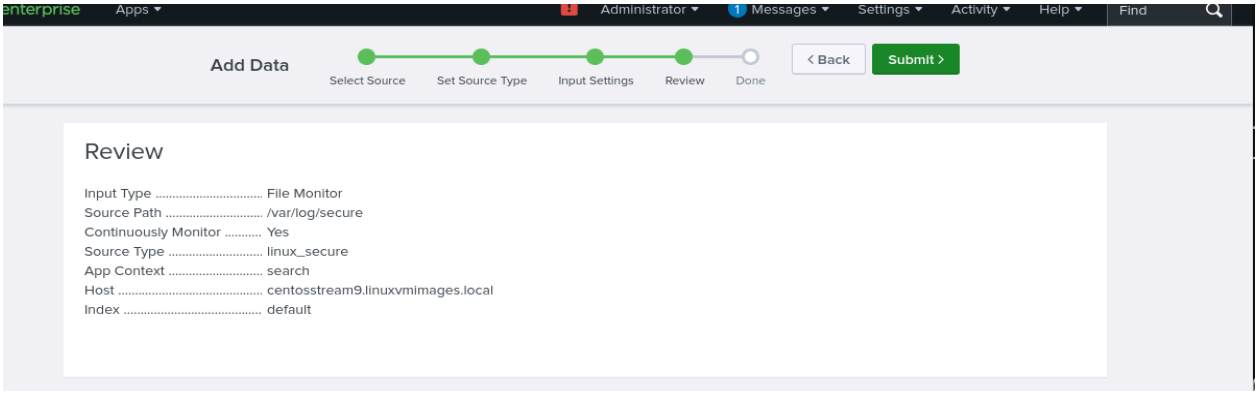


- Data is fetched and displayed you can click `Next >` button





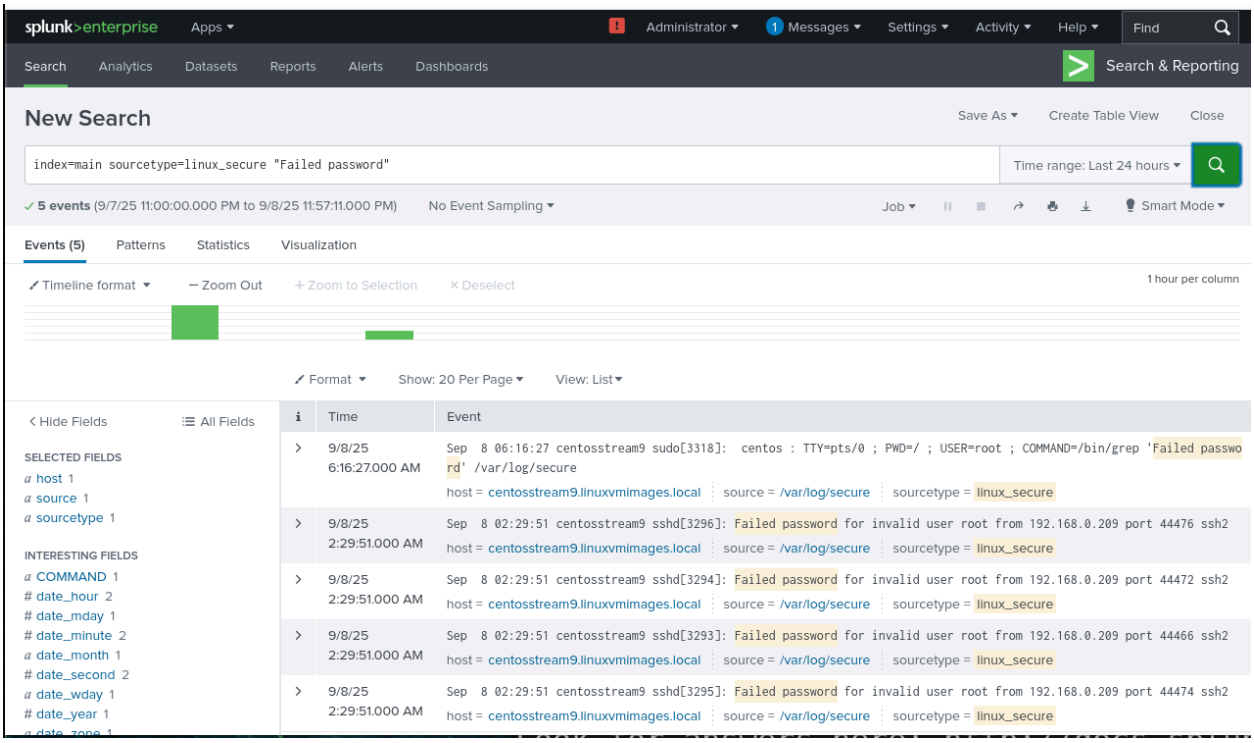
- The data input settings must be kept by default until one knows exactly what they are doing
- Next click `Review >` button



- 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
Messages
Settings
Activity
Help
Find
Search
Analytics
Datasets
Reports
Alerts
Dashboards
Search & Reporting
New Search
Save As
Create Table View
Close
index=main sourcetype=linux_secure "Failed password"
Time range: Last 24 hours
5 events (9/7/25 11:00:00.000 PM to 9/8/25 11:57:11.000 PM)
No Event Sampling
Job
Smart Mode
Events (5)
Patterns
Statistics
Visualization
Timeline format
Zoom Out
Zoom to Selection
Deselect
1 hour per column
Format
Show: 20 Per Page
View: List
Hide Fields
All Fields
SELECTED FIELDS
host 1
source 1
sourcetype 1
INTERESTING FIELDS
COMMAND 1
date_hour 2
date_mday 1
date_minute 2
date_month 1
date_second 2
date_wday 1
date_year 1
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
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
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
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
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
host = centosstream9.linuxvmimages.local source = /var/log/secure sourcetype = linux_secure

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

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

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