Linux Project: Log File Analysis

Invalid Logins

Objective

Analyze system authentication logs to identify and summarize invalid login attempts on the system.

Tasks Performed & Commands Used

Step 1: Verified log files presence

```
ls -la /var/log/
```

Step 2: Searched for failed login attempts (initially with wrong keyword, no results)

```
grep -i 'Failed password' /var/log/auth.log
```

Step 3: Inspected logs manually and identified correct keyword (authentication failure)

```
cat /var/log/auth.log
```

```
sudo grep -i "authentication failure" /var/log/auth.log
```

Step 4: Filtered out command-related log entries

```
sudo grep -i "authentication failure" /var/log/auth.log | grep -v "COMMAND="
```

Step 5: Saved failed login logs to file

```
sudo grep -i "authentication failure" /var/log/auth.log | grep -v "COMMAND="
> ~/failed_logins.txt
```

Step 6: Counted total failed login attempts

```
sudo grep -c "authentication failure" /var/log/auth.log >
~/failed_login_count.txt
```

Findings

Failures detected from multiple sources:

- GUI login (gdm-password)

- Sudo authentication

- Other failures (polkit-agent, passwd)

Total failed login attempts: 11

Challenges Faced

Initially used the wrong keyword (Failed password) which gave no results.

Had to manually inspect logs to identify the correct keyword (authentication failure).

Needed to filter out command-related entries to ensure only actual login failures were analyzed.

SSH Connections

Objective

Analyze system authentication logs to identify and summarize SSH connection attempts on the system.

Tasks Performed & Commands Used

Step 1: Verified log files presence

```
ls -la /var/log/
```

Step 2: Checked authentication logs for SSH entries (no results found initially)

```
sudo grep ssh /var/log/auth.log
```

Step 3: Checked SSH service status and found it inactive

```
systemctl status ssh
```

Step 4: Installed and enabled OpenSSH server

```
sudo apt-get install openssh-server
```

```
systemctl status ssh
```

Step 5: Verified system IP address for testing

```
ip a
```

Step 6: Created a self-SSH connection (loopback login)

```
ssh user@127.0.0.1
```

Step 7: Re-checked authentication logs for accepted SSH entries

```
sudo grep "ssh" /var/log/auth.log | grep "Accepted"
```

Step 8: Saved the accepted connection log to a file

```
sudo grep "ssh" /var/log/auth.log | grep "Accepted" | tail -5 >
~/ssh_connections.txt
```

Step 9: Counted total accepted SSH connections

```
sudo grep "ssh" /var/log/auth.log | grep -c "Accepted" >
~/ssh_connections_accepted_count.txt
```

Findings

No SSH log entries were initially found.

The SSH service was inactive and required installation and activation.

After generating a test connection, only one log entry was captured.

This entry corresponded to the self-established SSH session into the same system.

Challenges Faced

Inactive SSH Service: No logs were available until the service was installed and enabled.

Verification Need: Had to create a self-connection to produce a valid log entry for testing and analysis.

Disk Logs

Objective

Analyze system logs to identify and summarize disk-related messages.

Tasks Performed & Commands Used

Step 1: Verified log files presence

```
ls -la /var/log/
```

Step 2: Initial search for disk-related messages (insufficient detail)

```
sudo dmesg | grep "disk"
```

Step 3: Filtered logs using the keyword sda

```
sudo dmesg | grep "sda"
```

Step 4: Saved first 10 sda-related messages to file

```
sudo dmesg | grep "sda" | head -10 > ~/disk_messages.txt
```

Step 5: Counted total occurrences of sda messages

```
sudo dmesg | grep -c "sda" > ~/disk_messages_count.txt
```

Findings

Disk-related logs were successfully retrieved by focusing on sda entries.

From the logs, the following details were observed:

- Disk [sda] detected with a capacity of 107 GB (100 GiB).

- Write protection reported as off.

- Drive cache unavailable; assumed write-through caching.

- Partitions identified: sda1, sda2, sda3.

- Disk attached as a SCSI device.

- Partition sda3 initially mounted read-only (ro) with EXT4.

- Partition sda3 later re-mounted as read-write (rw).

- Warning noted: block capability attribute has been deprecated.

Evidence files created: disk_messages.txt (first 10 entries), disk_messages_count.txt (count of occurrences).

Challenges Faced

The generic keyword 'disk' gave very limited results; had to refine the search using 'sda' based on actual log output.

Device identifiers (like sda) may differ across systems, so the search approach must be adapted per environment.

System Startup Logs

Objective

Analyze system logs to identify and summarize startup-related events on the system.

Tasks Performed & Commands Used

Step 1: Verified log files presence

```
ls -la /var/log/
```

Step 2: Checked for system startup messages (case-sensitive, no results)

```
sudo grep "systemd" /var/log/syslog | grep "startup"
```

Step 3: Re-ran with case-insensitive search (successful results)

```
sudo grep "systemd" /var/log/syslog | grep -i "startup"
```

Step 4: Displayed first few entries to review

```
sudo grep "systemd" /var/log/syslog | grep -i "startup" | head -5
```

Step 5: Saved startup log messages to file

```
sudo grep "systemd" /var/log/syslog | grep -i "startup" | head -5 >
~/startup_messages.txt
```

Step 6: Counted occurrences of startup messages

```
sudo grep "systemd" /var/log/syslog | grep -ic "startup" >
~/startup_messages_count.txt
```

Findings

The initial case-sensitive search returned no results.

The case-insensitive search captured 3 startup log entries.

From the logs, the following information was found:

- One startup finished in 9.269s.

- Another recorded 46.485s total boot time (19.999s kernel + 26.485s userspace).

- A later startup finished in 17.556s.

Evidence files created: startup_messages.txt (all 3 entries), startup_messages_count.txt (count = 3).

Challenges Faced

Case sensitivity in grep caused the first attempt to fail.

Only after retrying with -i were valid entries retrieved.

Summary of Log Files

Log Type	File Created	Count File
Failed Logins	failed_logins.txt	failed_login_count.txt
SSH Connections	ssh_connections.txt	ssh_connections_accepted_count.txt
Disk Logs	disk_messages.txt	disk_messages_count.txt
Startup Logs	startup_messages.txt	startup_messages_count.txt

Conclusion & Real-World Relevance

- QA testers can use log analysis to validate authentication mechanisms during testing.

- Ensures that security checks (like failed logins) are properly recorded in system logs.

- Helps detect unauthorized access attempts, contributing to penetration testing and security audits.

- Provides evidence for whether server-side logging meets compliance and security standards.

- Confirms that SSH authentication attempts are logged properly once the service is active.

- Useful for QA testers to validate logging of remote access activity during security testing.

- Provides a way to simulate and verify audit trails for compliance and penetration testing.

- Reinforces the importance of checking service availability before relying on log analysis.

- Ensures that disk detection, partitioning, and mounting events are logged properly during system startup.

- QA testers can confirm correct disk capacity and partitions are recognized.

- Filesystems are mounted in the expected mode (read-only vs read-write).

- Warnings or deprecated attributes are captured for review.

- Startup logs validate system boot sequence timing and events, important for performance testing.

- QA testers can verify that systemd initialization events are logged consistently.

- Useful for analyzing boot time performance, service startup validation, and system readiness checks.