CSPP Experiment–09

9.1 Aim: Analyzing System Logs for Security Events: Learn to navigate and search system logs to identify potentially relevant security events.

Analyzing System Logs for Security Events:
Learn to navigate and search system logs to identify potentially relevant security events.

9.2 Course Outcome:

Analyze and interpret system and application logs to detect, investigate, and respond to potential security events on both Windows and Linux systems.

9.3 Lab Objective:

To understand how to access, search, and filter system logs using built-in tools and commands in Windows and Linux operating systems to identify possible security-related incidents.

9.4 Requirements:

Operating Systems:

- Windows 10/11 or Windows Server
- Linux (Ubuntu/Kali/CentOS)

Tools/Utilities:

- **Windows:** Event Viewer, PowerShell
- **Linux:** grep, awk, cut, tail, rsyslog, Log Management Tools (e.g., Loggly)

Sample Logs:

- Windows: Security, System, and Application Logs
- Linux: /var/log/auth.log, /var/log/syslog, /var/log/messages

9.5 Theory:

System logs are critical for monitoring and investigating security events.
They record authentication attempts, system changes, service failures, and network activities.

- **Windows Logs:** Centrally stored and accessible via Event Viewer or PowerShell.
- **Linux Logs:** Stored in /var/log/ directory, readable using text-based utilities.

Analyzing logs helps in identifying anomalies such as failed logins, privilege escalations, malware activity, or policy violations.
Log management solutions further automate event correlation, visualization, and alerting.

9.6 Procedure

Part A – Working with Windows Event Logs

1. Understanding Logs

- Logs record system and application events for troubleshooting and monitoring.
- Windows Event Logs include:
    - **Application**
    - **Security**
    - **Setup**
    - **System**
    - **Forwarded Events**

2. Accessing Windows Event Viewer

You can open Event Viewer through:

- **Control Panel → Administrative Tools → Event Viewer**
- **Server Manager → Tools → Event Viewer**
- **Windows Admin Center → Events**
- **Computer Management → Event Viewer**
- **Command Prompt:** eventvwr

3. Navigating Event Viewer

- **Navigation Pane:** Select log type (Application, Security, etc.)
- **Detail Pane:** View event list and details
- **Action Pane:** Filter, clear, or save logs

Event Severity Levels:
Information | Warning | Error | Critical | Audit Success | Audit Failure

4. Managing Logs

- **Filter Events:** Use *Filter Current Log*
- **Clear Logs:** *Clear Log* to delete entries
- **Export Logs:** *Save All Events As (.evtx)*

5. Creating Custom Views

- Navigate to **Custom Views → Create Custom View**
- Define filters (e.g., Critical & Error for .NET Runtime)
- Save and export/import custom views

6. Using Summary Views

- Overview shows total event counts and recent errors
- Recently Viewed Nodes lists recent logs
- Log Summary expands details per log type

7. Viewing Other Application Logs

- **DNS Manager:** DNS server logs
- **Failover Cluster Manager:** Cluster events
- **IIS Logs:** %SystemDrive%\inetpub\logs\LogFiles
- **Task Scheduler History:** Task execution history

8. Managing Logs with PowerShell

Task	Command Example
List all logs	Get-WinEvent -ListLog *
View specific log	Get-WinEvent -LogName 'Application'
Limit results	Get-WinEvent -LogName 'Application' -MaxEvents 5
Filter by Event ID	Get-WinEvent -FilterHashtable @{Logname='Security'; Id='4672'}

Filter by Time	Get-WinEvent -FilterHashtable @{Logname='Security'; Id='4672'; StartTime=(Get-Date).AddHours(-1)}

9. Maintenance & Best Practices

- Monitor logs regularly for errors or suspicious activity
- Archive or clear old logs periodically
- Automate log review using PowerShell scripts

Part B – Analyzing Linux Logs

1. Understanding Linux Log Analysis

- Logs store system and application data in /var/log/
- Common tools: grep, awk, cut, tail
- Log analyzers: Loggly, Papertrail

2. Searching Logs Using grep

Command:
grep "search_string" /path/to/logfile
Example:
grep "user hoover" /var/log/auth.log

3. Using Regular Expressions (Regex) with grep

Example:
grep -P "(?<=port\s)4792" /var/log/auth.log
Matches “4792” only if preceded by “port ”.

4. Surround Search with grep

Example:
grep -B 3 -A 2 'Invalid user' /var/log/auth.log
Shows 3 lines before and 2 after the match.

5. Monitoring Logs in Real Time with tail

Purpose	Command Example
View last 5 lines	tail -n 5 /var/log/messages

Live monitor	tail -f /var/log/auth.log
Filter live output	`tail -f /var/log/auth.log

6. Parsing Log Fields with cut

Example:
grep "authentication failure" /var/log/auth.log | cut -d '=' -f 8
Extracts specific field (e.g., username).

7. Filtering and Parsing with awk

Example:
awk '/sshd.*invalid user/ { print $9 }' /var/log/auth.log
Prints usernames of invalid login attempts.

8. Filtering Error Messages with awk

- **Option 1:** Modify rsyslog format

- **Option 2:**
  awk '/.err>/ {print}' /var/log/auth.log

9. Using Log Management Systems

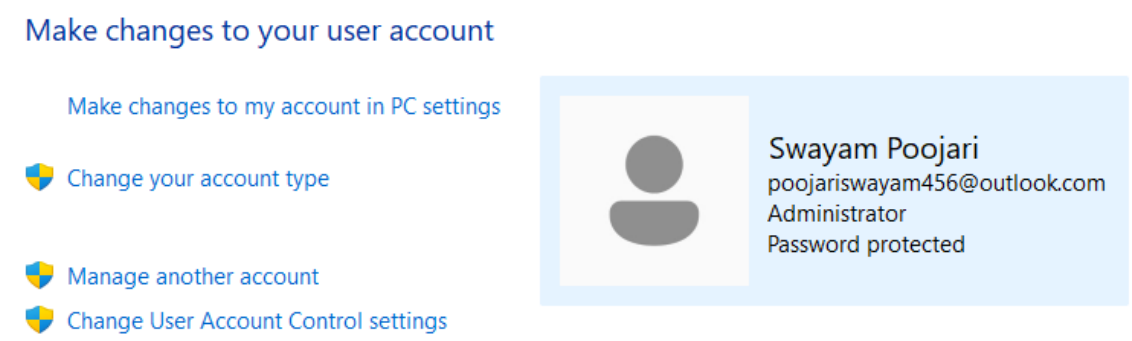
- Examples: SolarWinds Loggly, Papertrail
- Features:
    - Automatic parsing of SSH/syslog
    - Indexed search
    - Centralized dashboard
    - Severity-based filtering

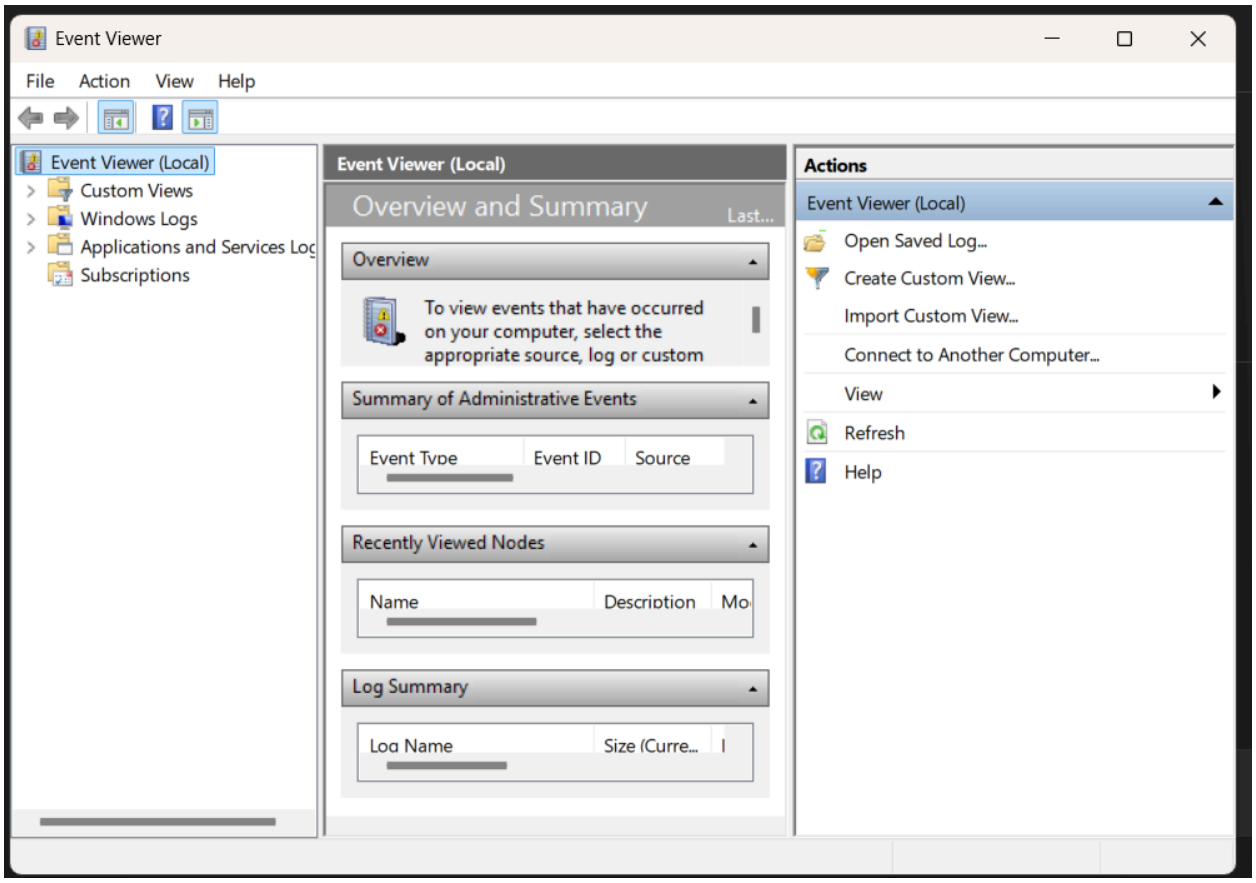
10. Best Practices

- Monitor /var/log/auth.log, /var/log/syslog, /var/log/messages
- Combine grep, awk, tail for detailed searches
- Use log management tools for scalability
- Archive logs periodically

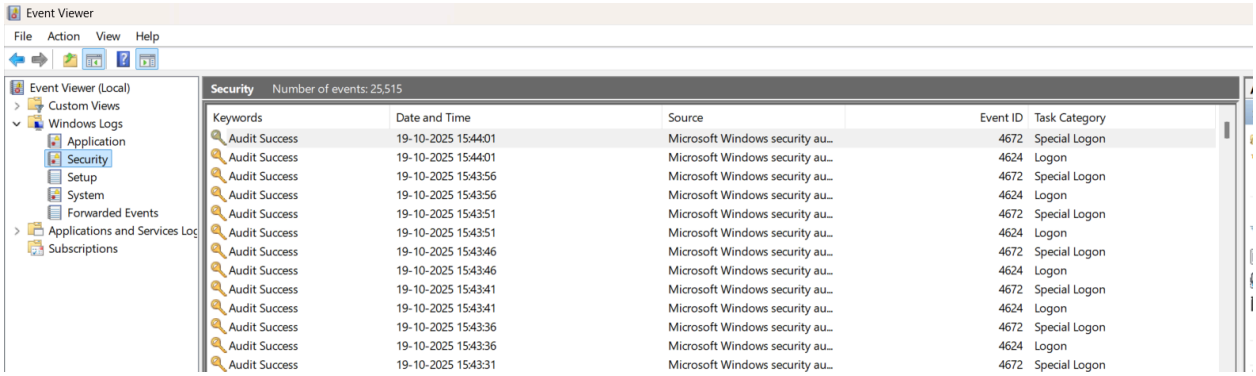
9.7 Output Screenshots:

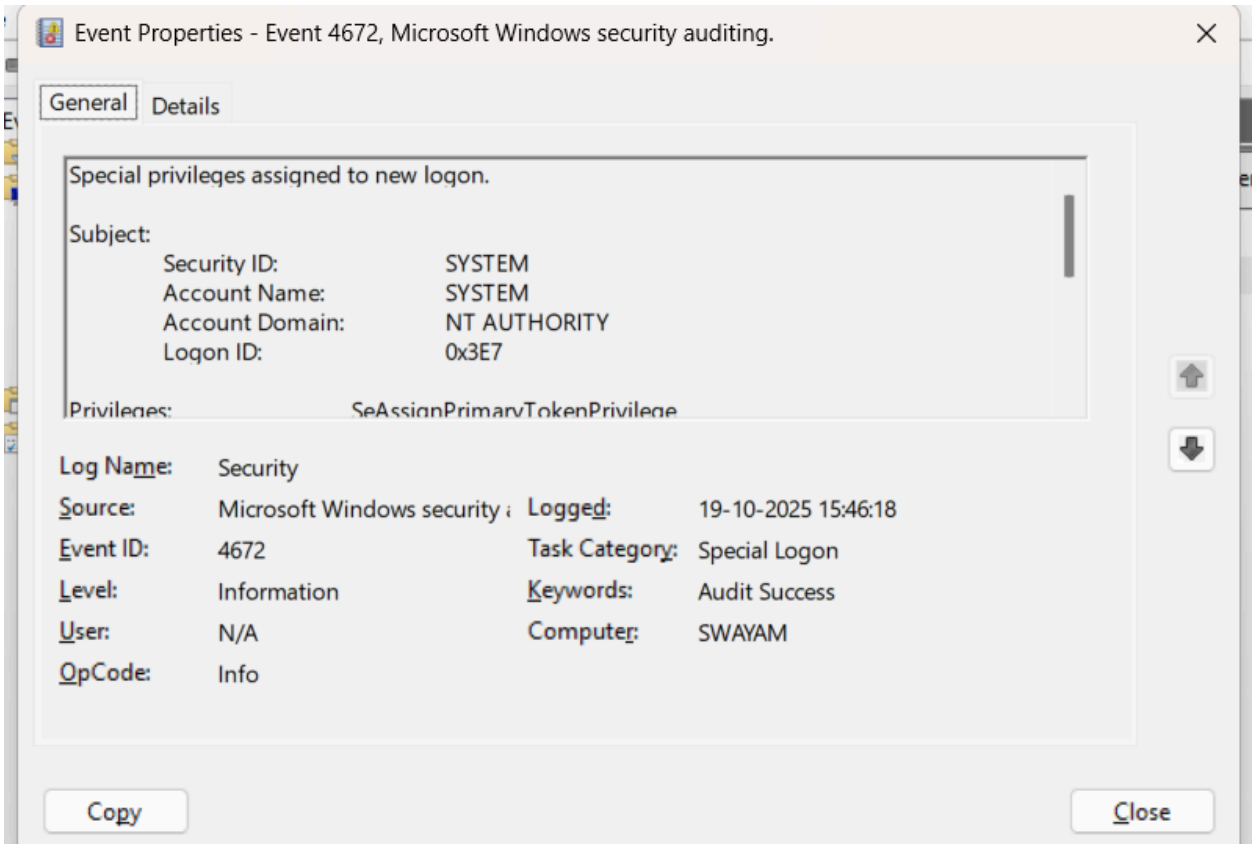
For Windows :

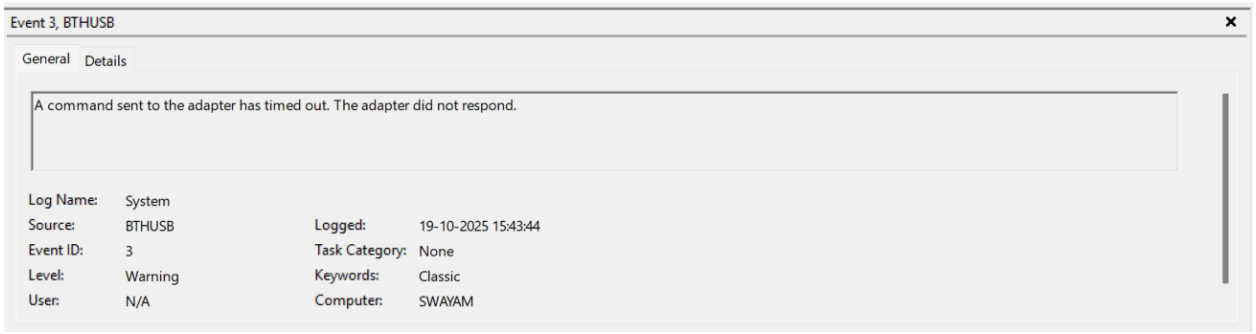


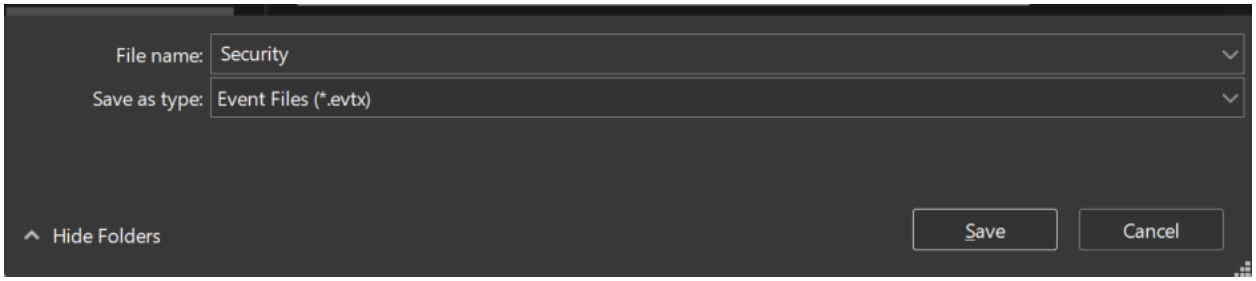



Event Viewer
File Action View Help
Event Viewer (Local)
Custom Views
Windows Logs
Applications and Services Log
Subscriptions
Windows Logs
Name Type Number of Events Size
Application Administrative 15,500 20.00 MB
Security Administrative 25,503 20.00 MB
Setup Operational 2,450 1.00 MB
System Administrative 38,558 20.00 MB
Forwarded Events Operational 0 0 Bytes
Actions
Windows Logs
Open Saved Log...
Create Custom View...
Import Custom View...
View
Refresh
Help


Event Viewer
File Action View Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services Log
Subscriptions
Security Number of events: 25,515
Keywords Date and Time Source Event ID Task Category
Audit Success 19-10-2025 15:44:01 Microsoft Windows security au... 4672 Special Logon
Audit Success 19-10-2025 15:44:01 Microsoft Windows security au... 4624 Logon
Audit Success 19-10-2025 15:43:56 Microsoft Windows security au... 4672 Special Logon
Audit Success 19-10-2025 15:43:56 Microsoft Windows security au... 4624 Logon
Audit Success 19-10-2025 15:43:51 Microsoft Windows security au... 4672 Special Logon
Audit Success 19-10-2025 15:43:51 Microsoft Windows security au... 4624 Logon
Audit Success 19-10-2025 15:43:46 Microsoft Windows security au... 4672 Special Logon
Audit Success 19-10-2025 15:43:46 Microsoft Windows security au... 4624 Logon
Audit Success 19-10-2025 15:43:41 Microsoft Windows security au... 4672 Special Logon
Audit Success 19-10-2025 15:43:41 Microsoft Windows security au... 4624 Logon
Audit Success 19-10-2025 15:43:36 Microsoft Windows security au... 4672 Special Logon
Audit Success 19-10-2025 15:43:36 Microsoft Windows security au... 4624 Logon
Audit Success 19-10-2025 15:43:31 Microsoft Windows security au... 4672 Special Logon


Event Properties - Event 4672, Microsoft Windows security auditing.
General
Details
Special privileges assigned to new logon.
Subject:
Security ID: SYSTEM
Account Name: SYSTEM
Account Domain: NT AUTHORITY
Logon ID: 0x3E7
Privileges: SeAssignPrimaryTokenPrivilege
Log Name: Security
Source: Microsoft Windows security Logged: 19-10-2025 15:46:18
Event ID: 4672 Task Category: Special Logon
Level: Information Keywords: Audit Success
User: N/A Computer: SWAYAM
OpCode: Info
Copy
Close

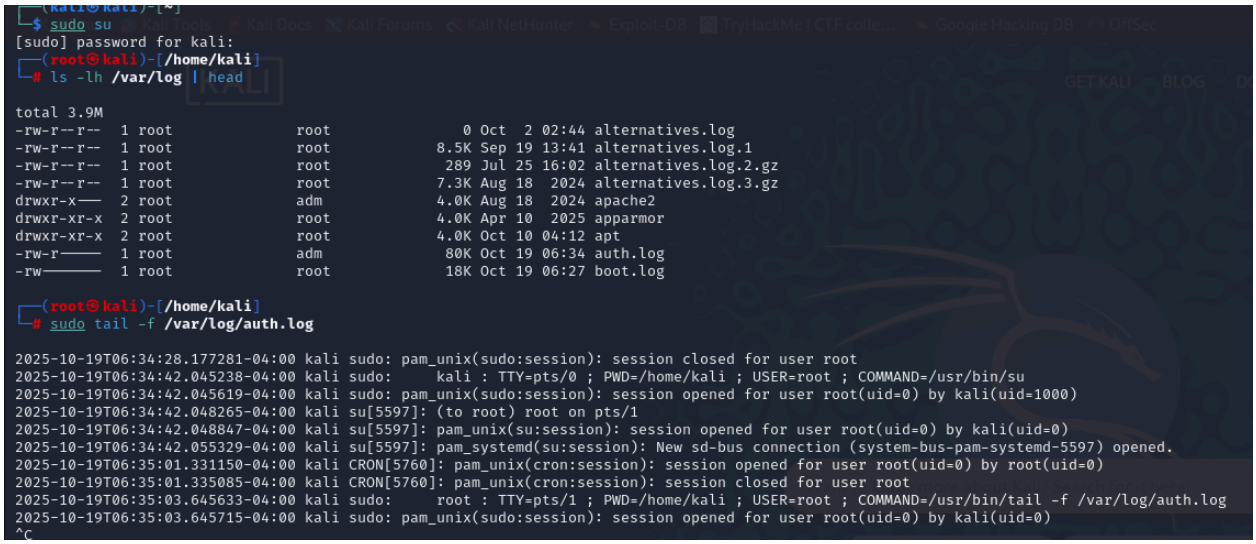





```
                    Get-WinEvent -ListLog * | Select-Object LogName, RecordCount | Format-Table -AutoSize
>> C:\WINDOWS\system32>

LogName                                                                 RecordCount
-------                                                                 -----------
Windows PowerShell                                                            11827
System                                                                        38582
Security                                                                      25543
OneApp_IGCC                                                                    1545
OAlerts                                                                          81
Key Management Service                                                            0
Internet Explorer                                                                 0
IntelAudioServiceLog                                                              0
HardwareEvents                                                                    0
Application                                                                   15471
Windows Networking Vpn Plugin Platform/OperationalVerbose
Windows Networking Vpn Plugin Platform/Operational
Synced-Passkey-Provider/Operatonal                                                0
SMSApi                                                                            0
Setup                                                                          2453
Plugin-Passkey-Providers/Operational                                              0
```

For Linux :

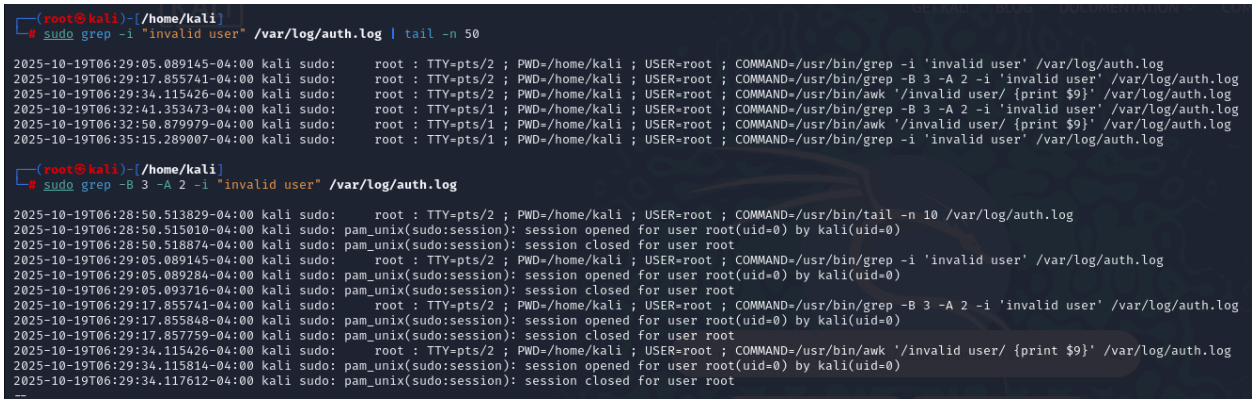


```
$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# ls -lh /var/log | head

total 3.9M
-rw-r--r--  1 root              root                0 Oct  2 02:44 alternatives.log
-rw-r--r--  1 root              root             8.5K Sep 19 13:41 alternatives.log.1
-rw-r--r--  1 root              root              289 Jul 25 16:02 alternatives.log.2.gz
-rw-r--r--  1 root              root             7.3K Aug 18  2024 alternatives.log.3.gz
drwxr-x---  2 root              adm              4.0K Aug 18  2024 apache2
drwxr-xr-x  2 root              root             4.0K Apr 10  2025 apparmor
drwxr-xr-x  2 root              root             4.0K Oct 10 04:12 apt
-rw-r-----  1 root              adm               80K Oct 19 06:34 auth.log
-rw-------  1 root              root              18K Oct 19 06:27 boot.log

┌──(root㉿kali)-[/home/kali]
└─# sudo tail -f /var/log/auth.log

2025-10-19T06:34:28.177281-04:00 kali sudo: pam_unix(sudo:session): session closed for user root
2025-10-19T06:34:42.045238-04:00 kali sudo:     kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/su
2025-10-19T06:34:42.045619-04:00 kali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by kali(uid=1000)
2025-10-19T06:34:42.048265-04:00 kali su[5597]: (to root) root on pts/1
2025-10-19T06:34:42.048847-04:00 kali su[5597]: pam_unix(su:session): session opened for user root(uid=0) by kali(uid=0)
2025-10-19T06:34:42.055329-04:00 kali su[5597]: pam_systemd(su:session): New sd-bus connection (system-bus-pam-systemd-5597) opened.
2025-10-19T06:35:01.331150-04:00 kali CRON[5760]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
2025-10-19T06:35:01.335085-04:00 kali CRON[5760]: pam_unix(cron:session): session closed for user root
2025-10-19T06:35:03.645633-04:00 kali sudo:     root : TTY=pts/1 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/auth.log
2025-10-19T06:35:03.645715-04:00 kali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by kali(uid=0)
^C
```

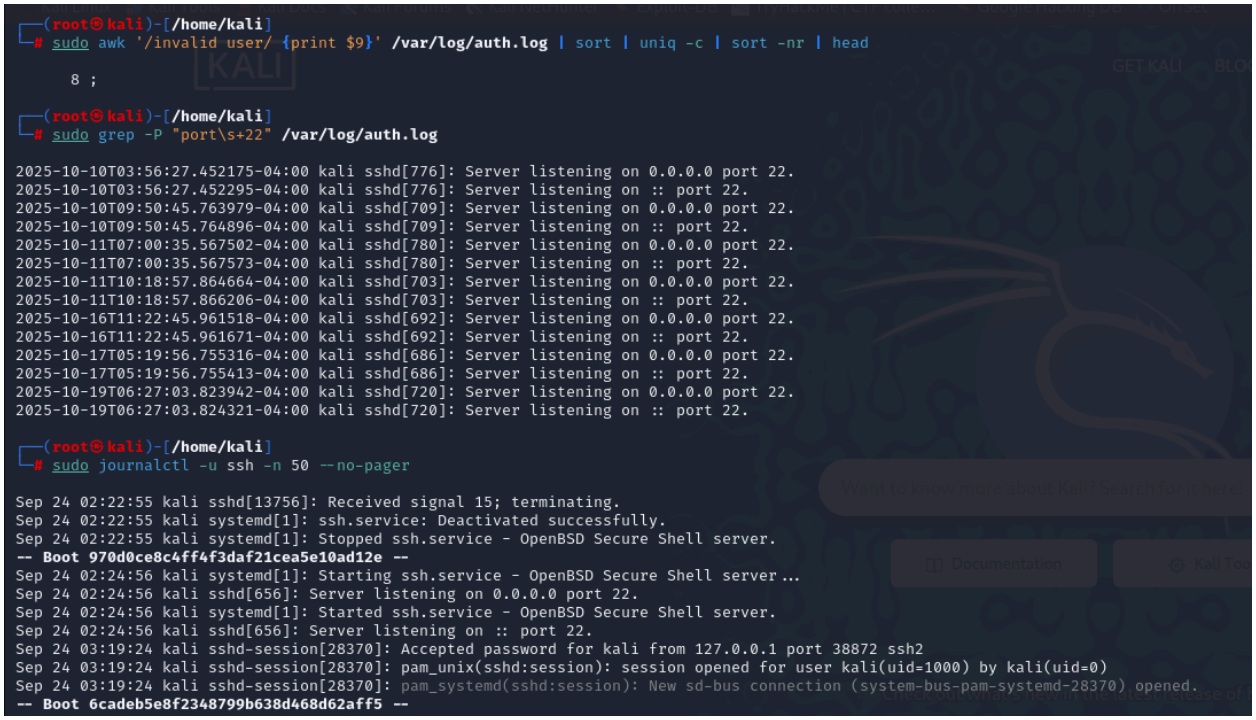


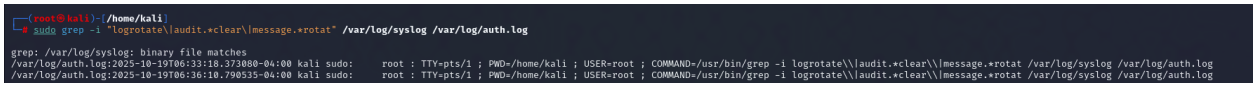
```
┌──(root㉿kali)-[/home/kali]
└─# sudo grep -i "invalid user" /var/log/auth.log | tail -n 50

2025-10-19T06:29:05.089145-04:00 kali sudo:     root : TTY=pts/2 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep -i 'invalid user' /var/log/auth.log
2025-10-19T06:29:17.855741-04:00 kali sudo:     root : TTY=pts/2 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep -B 3 -A 2 -i 'invalid user' /var/log/auth.log
2025-10-19T06:29:34.115426-04:00 kali sudo:     root : TTY=pts/2 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/awk '/invalid user/ {print $9}' /var/log/auth.log
2025-10-19T06:32:41.353473-04:00 kali sudo:     root : TTY=pts/1 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep -B 3 -A 2 -i 'invalid user' /var/log/auth.log
2025-10-19T06:32:50.879979-04:00 kali sudo:     root : TTY=pts/1 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/awk '/invalid user/ {print $9}' /var/log/auth.log
2025-10-19T06:35:15.289007-04:00 kali sudo:     root : TTY=pts/1 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep -i 'invalid user' /var/log/auth.log

┌──(root㉿kali)-[/home/kali]
└─# sudo grep -B 3 -A 2 -i "invalid user" /var/log/auth.log

2025-10-19T06:28:50.513829-04:00 kali sudo:     root : TTY=pts/2 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/tail -n 10 /var/log/auth.log
2025-10-19T06:28:50.515010-04:00 kali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by kali(uid=0)
2025-10-19T06:28:50.518874-04:00 kali sudo: pam_unix(sudo:session): session closed for user root
2025-10-19T06:29:05.089145-04:00 kali sudo:     root : TTY=pts/2 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep -i 'invalid user' /var/log/auth.log
2025-10-19T06:29:05.089284-04:00 kali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by kali(uid=0)
2025-10-19T06:29:05.093716-04:00 kali sudo: pam_unix(sudo:session): session closed for user root
2025-10-19T06:29:17.855741-04:00 kali sudo:     root : TTY=pts/2 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep -B 3 -A 2 -i 'invalid user' /var/log/auth.log
2025-10-19T06:29:17.855848-04:00 kali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by kali(uid=0)
2025-10-19T06:29:17.857759-04:00 kali sudo: pam_unix(sudo:session): session closed for user root
2025-10-19T06:29:34.115426-04:00 kali sudo:     root : TTY=pts/2 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/awk '/invalid user/ {print $9}' /var/log/auth.log
2025-10-19T06:29:34.115814-04:00 kali sudo: pam_unix(sudo:session): session opened for user root(uid=0) by kali(uid=0)
2025-10-19T06:29:34.117612-04:00 kali sudo: pam_unix(sudo:session): session closed for user root
```





9.8 Conclusion:

In this experiment, we successfully analyzed and interpreted system logs on both Windows and Linux platforms to detect potential security events. Using tools such as **Event Viewer** and **PowerShell** in Windows, and commands like **grep**, **awk**, and **tail** in Linux, we learned how to access, filter, and interpret logs for authentication attempts, system errors, and suspicious activities. This hands-on exercise enhanced our understanding of how log analysis supports incident detection, forensic investigation, and overall system security monitoring.