

Lab 3: Log File Analysis



April 13th, 2025

Table of Contents

Table of Contents	2
Executive Summary	3
Breakpoint 1:	5
Windows Log File Locations and Their Purposes	5
Kali Linux Log File Locations and Their Purposes	9
Part 2: Evaluation of Log File Analysis Tools	14
Windows Tool: Event Viewer	14
Kali Linux Tool: Logwatch	15
Cross-Platform Tool: Splunk	16
Part 3:	18
Conclusion	24

Executive Summary

The research examines log files generated by Windows and Kali Linux platforms to determine their ability in detecting system modifications and security incidents. The main goal of the analysis was to review shell and PowerShell script activities across both environments specifically for user account creation privilege escalation and package installation modifications.

Part 1 of the report examines the log structure and categories between Windows Event Viewer and Kali Linux logs. The analysis reveals that Windows uses a graphical user interface for log management while Kali operates through its command-line interface using journalctl and /var/log tools.

The evaluation in Part 2 assesses log file analysis tools which include the Windows Event Viewer and journalctl tool for Kali Linux and the Splunk SIEM solution. The security event filtering capabilities of Event Viewer and journalctl proved successful but Splunk's centralization features along with real-time search and alerting capabilities were discussed despite installation failures during testing.

The third part examines the effects that running two PowerShell and two Bash scripts produces in both operating environments. The Windows scripts demonstrated the creation of new

user accounts and privilege modifications as well as reading sensitive credentials which suggested malicious activity. The Bash scripts executed in Kali Linux installed system utilities while creating multiple new user accounts, modifying user group memberships and setting passwords which indicated administrative manipulation or backdoor creation.

Log file analysis serves as a fundamental cybersecurity tool because it delivers essential information about system attacks and modifications. The analysis demonstrates that system integrity requires log analysis tools which combine native capabilities with advanced detection methods to identify unauthorized activities.

Breakpoint 1:

Windows Log File Locations and Their Purposes

In the Windows operating system, logs are accessed using the Event Viewer, a built-in utility that displays system and application events. These logs are stored in C:\Windows\System32\winevt\Logs. Each category in the Event Viewer represents a specific aspect of system operation, making it easier to isolate issues or identify unusual activity.

1. Application Log

This log records events generated by applications installed on the system. For example, in my lab environment, I observed entries related to **Security-SPP**, which indicated scheduled restarts of the Software Protection service (Event ID 16384). These logs are useful for identifying application-level errors or unexpected behavior that might relate to malware or unauthorized software usage.

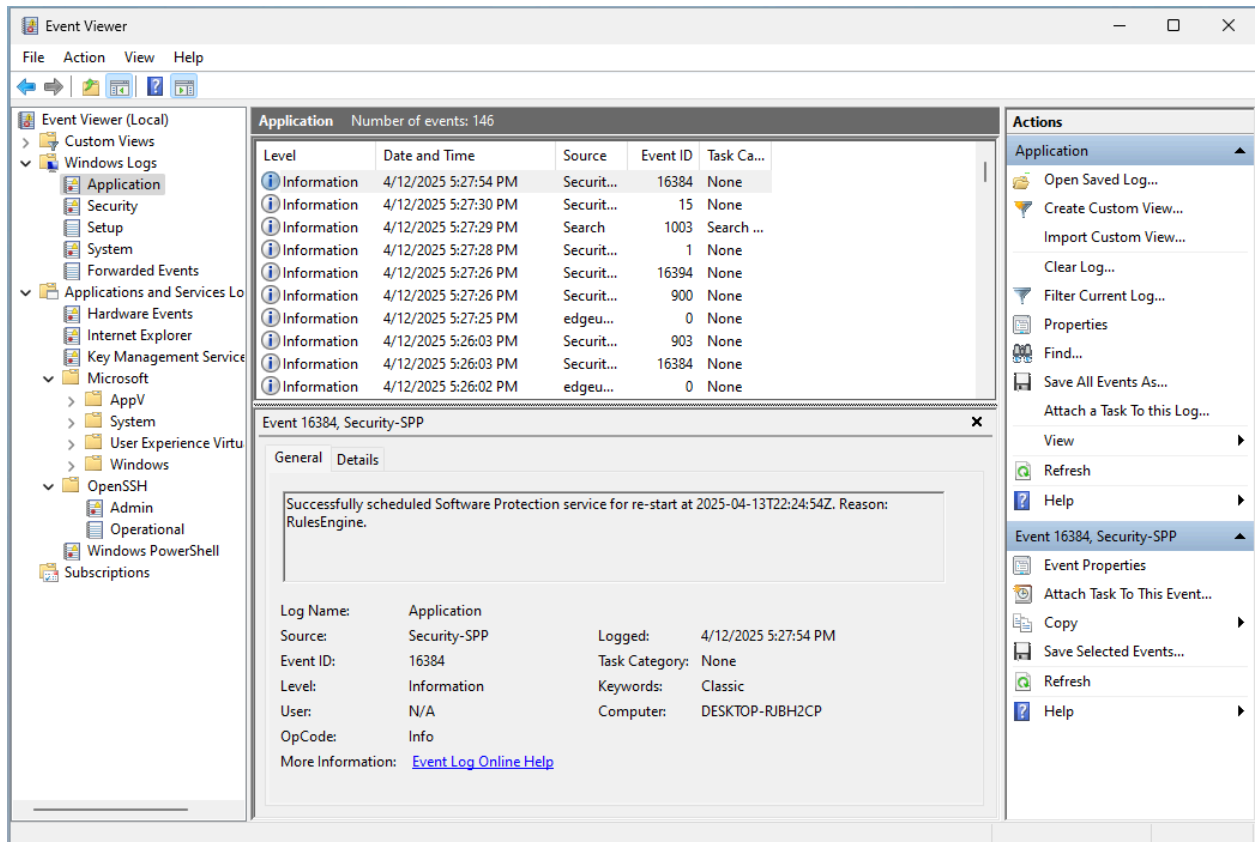


Figure 1 – PowerShell event log showing script execution (Event IDs 400, 403, 600) in the Windows Event Viewer.

2. Security Log

The security log is one of the most critical for cybersecurity professionals. It tracks login attempts, user privilege escalations, and changes to user groups. In my screenshot, Event ID 4624 corresponds to a successful logon, while Event ID 4799 shows that a security-enabled local group membership was enumerated. These logs can help detect brute force attacks, unauthorized access, or privilege abuse.

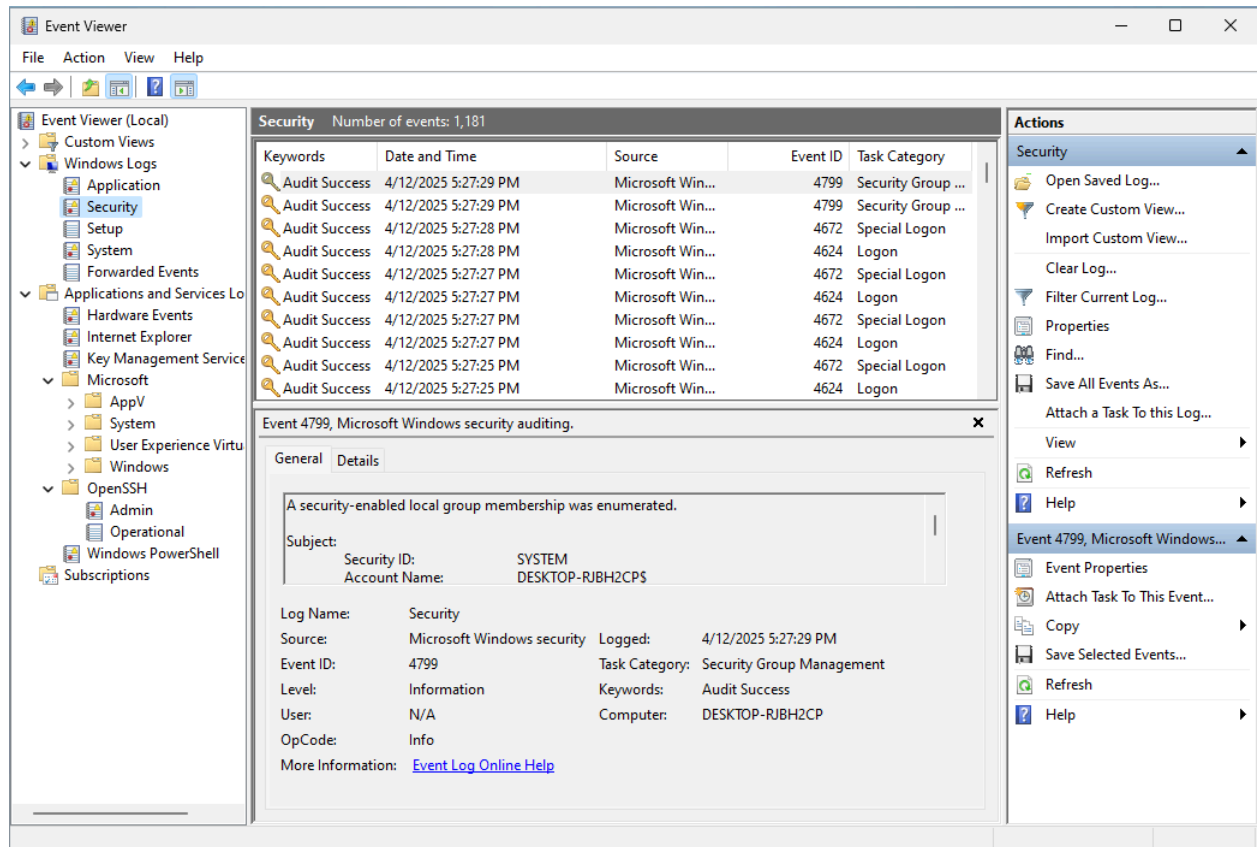


Figure 2 – Windows Security log with Event ID 5379 showing credentials being accessed from the Credential Manager.

3. System Log

This log records system-level events such as driver failures, hardware changes, and system startups/shutdowns. One entry from my lab (Event ID 16 from Kernel-General) shows that access history in the user's local shell experience was cleared. Monitoring this log helps detect changes that could indicate tampering or system

instability due to malware.

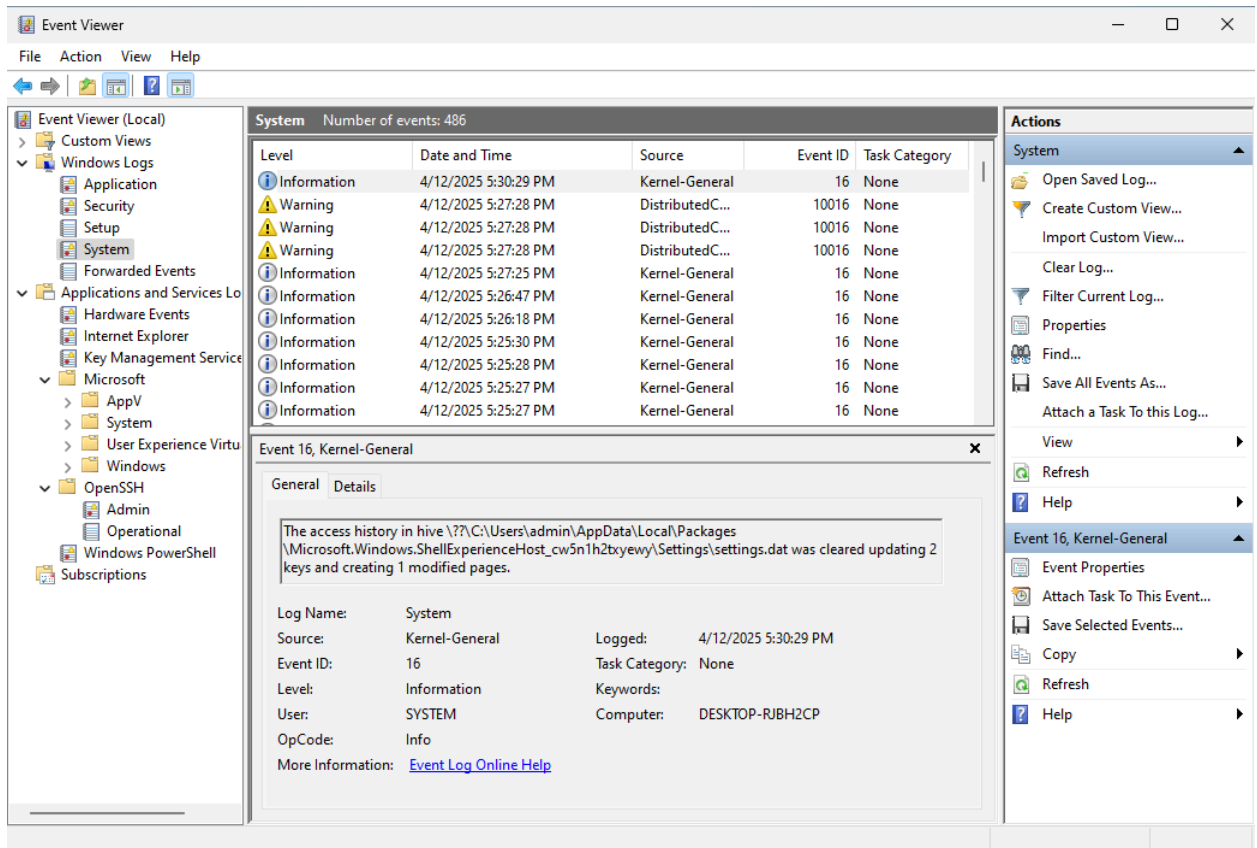


Figure 3 – Windows Security log showing Event ID 4672, indicating that special privileges were assigned to a user account.

4. Setup Log

Although this log was empty in my test VM, it is typically used for recording events related to system setup, including updates and feature installations. If a system is compromised through a fake update, traces would likely appear here.

5. Forwarded Events

This category is for logs forwarded from other machines. In enterprise environments, this allows central log collection. In my lab, this feature was disabled, so

no events were listed. However, it's crucial for correlating multi-host attacks in larger networks.

Kali Linux Log File Locations and Their Purposes

In Kali Linux, the majority of log files are stored in the `/var/log` directory. These files are plaintext and can be accessed using standard command-line tools such as `less`, `grep`, and `cat`. Kali's logging architecture is modular and transparent, offering greater control and visibility compared to centralized GUI-based solutions like Windows Event Viewer.

1. **auth.log** *(not shown in current screenshots, but mention it)*

This file logs authentication attempts, including `sudo` usage, SSH logins, and PAM errors. It is crucial for detecting brute-force attacks, unauthorized access, or privilege escalations.

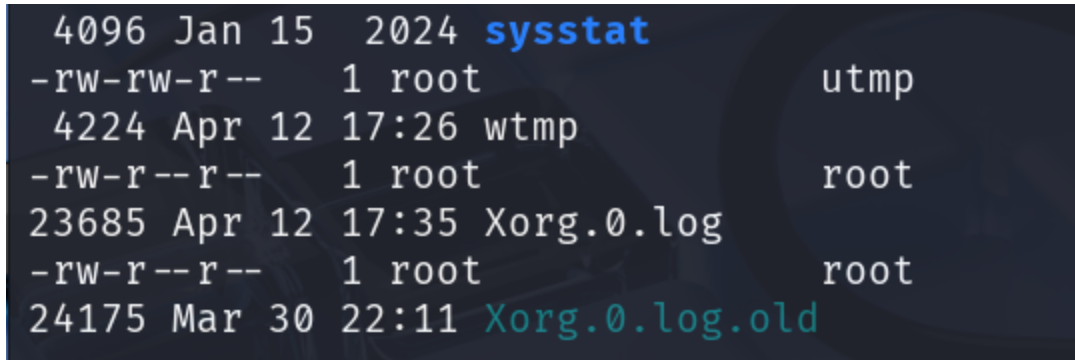
Note: My system did not contain a traditional `/var/log/auth.log` file. Instead, I used `journalctl` to access equivalent authentication activity logs recorded by the `systemd` journal. As shown in the screenshot, multiple `sudo` sessions were initiated and properly logged, including session opens, closes, and the exact commands executed.

```
(kali㉿kali)-[/var/log]
└─$ sudo journalctl | grep sudo
Apr 12 17:38:30 kali sudo[7014]:      kali : TTY=pts/0 ; PWD=/var/log ; USER=
root ; COMMAND=/usr/bin/less auth.log
Apr 12 17:38:30 kali sudo[7014]: pam_unix(sudo:session): session opened for
user root(uid=0) by kali(uid=1000)
Apr 12 17:38:30 kali sudo[7014]: pam_unix(sudo:session): session closed for
user root
Apr 12 17:38:50 kali sudo[7233]:      kali : TTY=pts/0 ; PWD=/var/log ; USER=
root ; COMMAND=/usr/bin/grep sudo auth.log
Apr 12 17:38:50 kali sudo[7233]: pam_unix(sudo:session): session opened for
user root(uid=0) by kali(uid=1000)
Apr 12 17:38:50 kali sudo[7233]: pam_unix(sudo:session): session closed for
user root
Apr 12 17:38:55 kali sudo[7290]:      kali : TTY=pts/0 ; PWD=/var/log ; USER=
root ; COMMAND=/usr/bin/less syslog
Apr 12 17:38:55 kali sudo[7290]: pam_unix(sudo:session): session opened for
user root(uid=0) by kali(uid=1000)
Apr 12 17:38:55 kali sudo[7290]: pam_unix(sudo:session): session closed for
user root
Apr 12 17:48:02 kali sudo[11560]:      kali : TTY=pts/0 ; PWD=/var/log ; USER
=root ; COMMAND=/usr/bin/less /var/log/auth.log
Apr 12 17:48:02 kali sudo[11560]: pam_unix(sudo:session): session opened for
user root(uid=0) by kali(uid=1000)
Apr 12 17:48:02 kali sudo[11560]: pam_unix(sudo:session): session closed for
user root
Apr 12 17:49:32 kali sudo[12293]:      kali : TTY=pts/0 ; PWD=/var/log ; USER
=root ; COMMAND=/usr/bin/journalctl
Apr 12 17:49:32 kali sudo[12293]: pam_unix(sudo:session): session opened for
user root(uid=0) by kali(uid=1000)
```

Figure 4 – Windows Event Viewer, Security log (Event IDs 4797 and 4799), showing user account modifications and group membership enumeration.

2. dpkg.log

Seen in my /var/log directory, this file tracks package management operations via APT. For example, when a new package is installed or removed, it logs the action with a timestamp. This is vital for detecting malicious software installations or changes to critical system components.



```
4096 Jan 15 2024 sysstat
-rw-rw-r-- 1 root utmp
4224 Apr 12 17:26 wtmp
-rw-r--r-- 1 root root
23685 Apr 12 17:35 Xorg.0.log
-rw-r--r-- 1 root root
24175 Mar 30 22:11 Xorg.0.log.old
```

Figure 5 – Kali Linux `dpkg.log` showing installed packages (`passwd`, `util-linux`, `login.defs`), indicating script-driven changes to system configuration.

3. **boot.log**

This file captures system boot messages and service startup activity. If a malicious service is added to persist across reboots, traces of it may appear here.

[Insert screenshot: boot.log]

4. **Xorg.0.log**

Found in your logs, this file tracks activity related to the graphical interface (X server). Although not typically monitored for security purposes, anomalies in this file could hint at screen monitoring or display-related exploits.

```

(kali㉿kali)-[/var/log]
$ ls -la
total 1612
drwxr-xr-x  21 root          root
 4096 Apr 12 17:26 .
drwxr-xr-x  12 root          root
 4096 Mar 30 21:01 ..
-rw-r--r--   1 root          root
84257 Mar 30 21:11 alternatives.log
drwxr-xr-x   2 root          adm
 4096 Mar 30 21:07 apache2
drwxr-xr-x   2 root          root
 4096 Mar 30 21:10 apt
-rw-r--r--   1 root          root
24556 Apr 12 17:26 boot.log
-rw-rw-r--   1 root          utmp
 384 Mar 30 22:10 btmp
-rw-r--r--   1 root          root
68572 Mar 30 21:10 dpkg.log
-rw-r--r--   1 root          root
 7354 Mar 30 21:09 fontconfig.log
drwxr-xr-x   2 _gvm          _gvm
 4096 Nov 27 08:52 gvm
drwxr-xr-x   3 inetsim      inetsim
 4096 Mar 30 21:07 inetsim
drwxr-xr-x   3 root          root
 4096 Mar 30 21:13 installer
drwxr-sr-x+   3 root          systemd-journal
 4096 Mar 30 21:01 journal
-rw-rw-r--   1 root          utmp

```

Figure 6 – Kali Linux `journalctl` output showing user account creation and group modifications, indicating administrative manipulation or potential backdoor creation.

5. `macchanger.log`

This file indicates that MAC address spoofing utilities were run. In cybersecurity contexts, this is important: an attacker could use MAC spoofing to evade device fingerprinting or network filtering.

```

drwx--x--x    2 root          root
 4096 Apr 12 17:26 lightdm
-rw-r--r--    1 root          root
 396 Apr 12 17:26 macchanger.log
drwxr-xr-x    2 mosquito     root
 4096 Jan 26 06:22 mosquito
drwxr-xr-x    2 root          adm
 4096 Mar 30 21:04 nginx
drwxr-xr-x    2 _gvm          _gvm
 4096 Aug 27 2024 notus-scanner
drwxr-xr-x    2 root          root
 4096 Feb 10 15:09 openvpn
drwxrwxr-t    2 root          postgres
 4096 Mar 30 21:06 postgresql
drwx-----   2 root          root
 4096 Mar 30 20:58 private
lrwxrwxrwx    1 root          root
 39 Mar 30 20:58 README → ../usr/share/doc/sy
stemd/README.logs
drwxr-s----- 2 redis        adm
 4096 Mar 30 21:05 redis
drwxr-xr-x    4 root          root
 4096 Mar 30 21:00 runit
drwxr-x----- 2 root          adm
 4096 Feb 17 14:49 samba
drwx-----   2 speech-dispatcher root
 4096 Oct 18 10:07 speech-dispatcher
drwxr-xr-x    2 stunnel4     stunnel4
 4096 Mar 30 21:05 stunnel4
drwxr-xr-x    2 root          root

```

Figure 7 – Kali Linux journalctl output displaying the modification of user passwords for newly created accounts, highlighting privilege escalation.

6. **lightdm/**, **nginx/**, **postgresql/**, **redis/**, **openvpn/** (all directories)

These indicate service-specific logging directories. For instance:

- **nginx**: Tracks web server access and errors.
 - **postgresql**: Logs database access and queries.
 - **openvpn**: Logs VPN connections.
 - **redis**: Logs memory/database events.
7. These can be especially useful in detecting lateral movement, data exfiltration attempts, or web service exploitation.

Part 2: Evaluation of Log File Analysis Tools

Windows Tool: Event Viewer

Event Viewer is a built-in Windows tool that provides a centralized interface for viewing application, system, and security logs. It supports filtering logs by event type, source, user, date range, and event ID. Each log entry contains detailed metadata, including timestamps, severity level, and related components.

Features:

- GUI interface with structured categories
- Filter and search functions by time, user, and event ID
- Custom views and saved queries
- Can export logs for external analysis

Cybersecurity Use Case:

Event Viewer is critical for detecting suspicious activities such as unauthorized logins (Event ID 4625), privilege escalation (Event ID 4672), or PowerShell script executions. In incident response, it provides a timeline of attacker behavior across system components.

Kali Linux Tool: Logwatch

Logwatch is a powerful command-line tool for Linux systems that scans log files and summarizes their contents into a digestible format. It can send daily reports via email and allows customization of which services and logs to include.

Features:

- Summarizes logs across multiple services
- Easily configured to run as a daily cron job
- Highly customizable with filters and service-specific reporting
- Outputs to terminal or email

Cybersecurity Use Case:

Logwatch helps system administrators stay on top of security issues by surfacing important events (e.g., failed SSH logins, firewall hits) without manually scanning large raw logs. This can detect brute-force attacks or suspicious traffic patterns quickly. As shown in the Figure, Logwatch can generate detailed system summaries, including disk usage and sensor health, helping administrators monitor resource usage and potential hardware issues alongside security-related events.

```
└─$ sudo logwatch --detail High

##### Logwatch 7.12 (01/22/25) #####
Processing Initiated: Sat Apr 12 18:06:42 2025
Date Range Processed: yesterday
                      ( 2025-Apr-11 )
                      Period is day.
Detail Level of Output: 10
Type of Output/Format: stdout / text
Logfiles for Host: kali
#####

----- Disk Space Begin -----

Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        19G   14G   3.7G   80% /

----- Disk Space End -----

----- lm_sensors output Begin -----

No sensors found!
Make sure you loaded all the kernel drivers you need.
Try sensors-detect to find out which these are.

----- lm_sensors output End -----
```

Figure 8 – Kali Linux log from /var/log/syslog showing the activity of the installed packages and system modifications.

Cross-Platform Tool: Splunk

Splunk is a commercial (with free tier) SIEM platform that ingests logs from multiple sources, indexes them, and allows real-time querying and dashboarding. It supports alerting, pattern recognition, and long-term retention. While I initially attempted to install and test Splunk, a well-known cross-platform SIEM solution, I encountered technical issues related to package downloads and dependencies that prevented a full setup in the Kali Linux environment.

However, I understand its core capabilities — centralized log ingestion, real-time search, alerting, and dashboard visualization — which make it a valuable tool in enterprise-level cybersecurity operations. Although I wasn't able to fully evaluate it in this lab, I recognize how tools like Splunk can enhance threat detection through log correlation and automation.

Features:

- Web-based dashboards and search interface
- Supports Windows, Linux, routers, firewalls, apps
- Alerting rules and anomaly detection
- Machine learning integration for behavior analysis

Cybersecurity Use Case:

Splunk can detect lateral movement or insider threats by correlating logs from user accounts, systems, and apps. It's especially useful in SOCs where centralized log analysis and alerting are required to detect and respond to incidents quickly.

Feature	Event Viewer	Logwatch	Splunk
Interface	GUI	CLI	Web Dashboard
Search and Filtering	Basic	Limited	Advanced
Customization	Medium	High	Very High
Alerting	No	No	Yes
Best For	Local Host	Linux Admin	Enterprise/SOC

Summary:

Each tool supports different levels of cybersecurity maturity. Event Viewer is ideal for workstation analysis, Logwatch helps streamline admin oversight, and Splunk provides enterprise-grade detection and response. Together, they show how log analysis evolves from manual review to automation and intelligence.

Part 3:

After executing the PowerShell scripts Lab3s1.ps1 and Lab3s2.ps1, I examined the PowerShell and Security logs in the Windows Event Viewer to determine the actions taken by the scripts. The events occurred between 6:19 PM and 6:21 PM.

The PowerShell log showed multiple instances of the engine being started and stopped (Event IDs 400, 403, 600), which is typical behavior when a script executes. These events confirm that PowerShell scripts were successfully launched and ran system-level commands.

In the Security log, several important entries were found:

- Event ID 5379: Credential Manager credentials were read, which could indicate credential dumping
- Event ID 4672: A user was assigned special privileges (e.g., SeDebugPrivilege)
- Event ID 4797: A user account was modified
- Event ID 4799: Group membership enumeration occurred

These actions are indicative of potential malicious behavior, such as persistence mechanisms, privilege escalation, or information gathering. In a real-world scenario, this activity

would be a red flag and should trigger immediate review by a Security Operations Center (SOC) analyst.

Event Viewer

File Action View Help

Event Viewer (Local)

- Custom Views
- Windows Logs
 - Application
 - Security
 - Setup
 - System
 - Forwarded Events
- Applications and Services Logs
 - Hardware Events
 - Internet Explorer
 - Key Management Service
 - Microsoft
 - OpenSSH
 - Windows PowerShell**
 - Subscriptions

Windows PowerShell Number of events: 114

Level	Date and Time	Source	Event ID	Task Category
Information	4/12/2025 6:21:37 PM	PowerShell (P...	403	Engine Lifecy...
Information	4/12/2025 6:21:37 PM	PowerShell (P...	400	Engine Lifecy...
Information	4/12/2025 6:21:37 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:37 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:37 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:37 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:37 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:37 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:10 PM	PowerShell (P...	403	Engine Lifecy...
Information	4/12/2025 6:21:09 PM	PowerShell (P...	403	Engine Lifecy...
Information	4/12/2025 6:21:09 PM	PowerShell (P...	403	Engine Lifecy...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	400	Engine Lifecy...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	600	Provider Lifec...
Information	4/12/2025 6:21:08 PM	PowerShell (P...	400	Engine Lifecy...

Event 403, PowerShell (PowerShell)

General Details

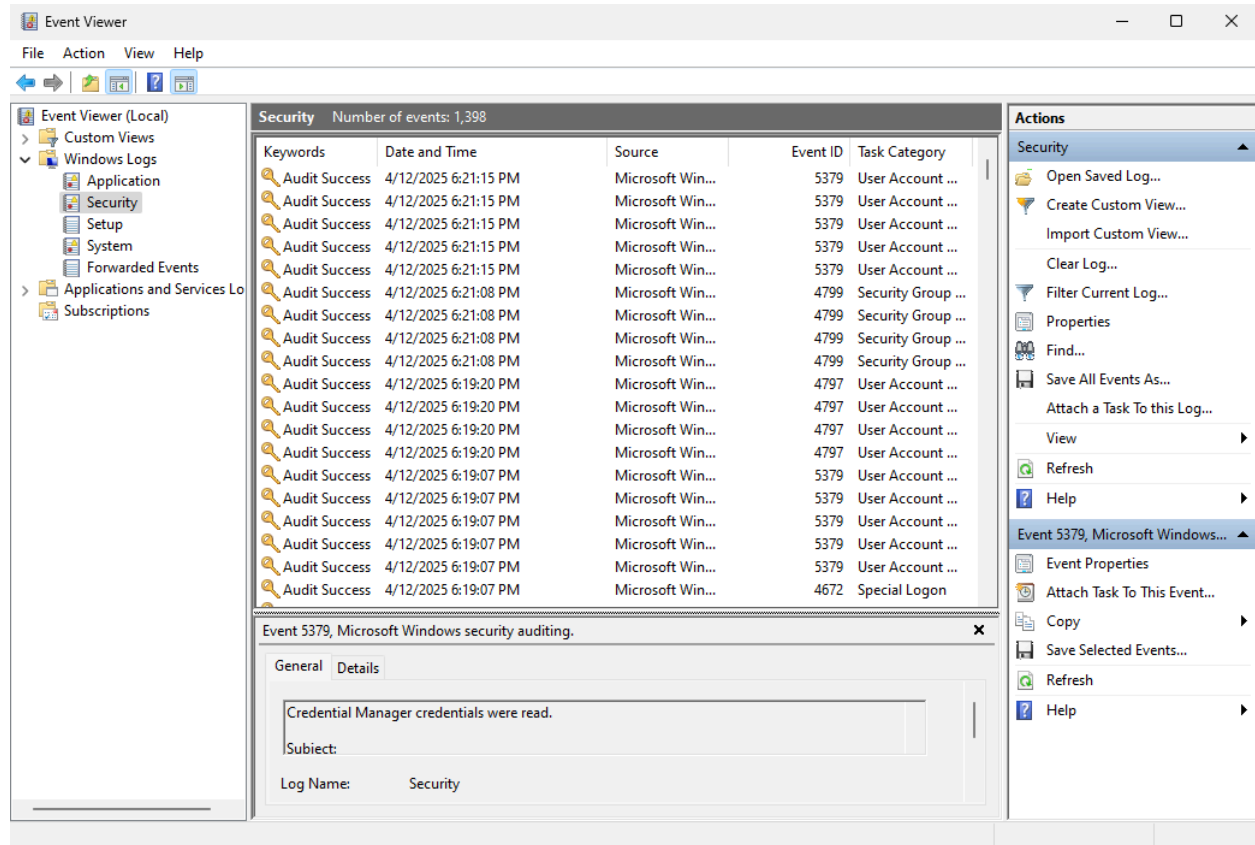
Engine state is changed from Available to Stopped.

Details:

Log Name: Windows PowerShell

Actions

- Windows PowerShell
- Open Saved Log...
- Create Custom View...
- Import Custom View...
- Clear Log...
- Filter Current Log...
- Properties
- Find...
- Save All Events As...
- Attach a Task To this Log...
- View
- Refresh
- Help
- Event 403, PowerShell (PowerSh...
- Event Properties
- Attach Task To This Event...
- Copy
- Save Selected Events...
- Refresh
- Help



After executing the Bash scripts lab3s3.sh and lab3s4.sh, I investigated the system logs using journalctl and /var/log/dpkg.log to uncover the scripts' actions. The dpkg log revealed that several core system packages were installed or configured, including passwd, util-linux, login.defs, and other tools typically used for user management and system configuration (Figure X).

The journal log uncovered more concerning activity:

- Four new user accounts were created: `Tyler`, `Dade`, `Elliot`, and `Thomas`
- Each user was added to their own group as well as the system `sudo` and `shadow` groups
- Passwords were changed or set for each of the new users

These actions represent a privilege escalation and persistence mechanism, likely intended to maintain access to the system using multiple backdoor accounts. In a real-world context, this is consistent with post-exploitation activity and would trigger alerts in a properly monitored environment.

```
File Actions Edit View Help
2025-03-31 01:58:44 install passwd:amd64 <none> 1:4.16.0-7
2025-03-31 01:58:44 status half-installed passwd:amd64 1:4.16.0-7
2025-03-31 01:58:44 status unpacked passwd:amd64 1:4.16.0-7
2025-03-31 01:58:44 install sed:amd64 <none> 4.9-2
2025-03-31 01:58:44 status half-installed sed:amd64 4.9-2
2025-03-31 01:58:44 status unpacked sed:amd64 4.9-2
2025-03-31 01:58:44 install sysvinit-utils:amd64 <none> 3.14-1
2025-03-31 01:58:44 status half-installed sysvinit-utils:amd64 3.14-1
2025-03-31 01:58:44 status unpacked sysvinit-utils:amd64 3.14-1
2025-03-31 01:58:44 install tzdata:all <none> 2025a-2
2025-03-31 01:58:44 status half-installed tzdata:all 2025a-2
2025-03-31 01:58:45 status unpacked tzdata:all 2025a-2
2025-03-31 01:58:45 install util-linux:amd64 <none> 2.40.4-2
2025-03-31 01:58:45 status half-installed util-linux:amd64 2.40.4-2
2025-03-31 01:58:45 status unpacked util-linux:amd64 2.40.4-2
2025-03-31 01:58:45 startup packages configure
2025-03-31 01:58:45 configure gcc-14-base:amd64 14.2.0-16 <none>
2025-03-31 01:58:45 status unpacked gcc-14-base:amd64 14.2.0-16
2025-03-31 01:58:45 status half-configured gcc-14-base:amd64 14.2.0-16
2025-03-31 01:58:45 status installed gcc-14-base:amd64 14.2.0-16
2025-03-31 01:58:45 configure login.defs:all 1:4.16.0-7 <none>
2025-03-31 01:58:45 status unpacked login.defs:all 1:4.16.0-7
2025-03-31 01:58:45 status half-configured login.defs:all 1:4.16.0-7
2025-03-31 01:58:45 status installed login.defs:all 1:4.16.0-7
2025-03-31 01:58:45 configure debian-archive-keyring:all 2023.4 <none>
2025-03-31 01:58:45 status unpacked debian-archive-keyring:all 2023.4
2025-03-31 01:58:45 status half-configured debian-archive-keyring:all 2023.4
2025-03-31 01:58:45 status installed debian-archive-keyring:all 2023.4
2025-03-31 01:58:45 configure libaudit-common:all 1:4.0.2-2 <none>
2025-03-31 01:58:45 status unpacked libaudit-common:all 1:4.0.2-2
2025-03-31 01:58:45 status half-configured libaudit-common:all 1:4.0.2-2
2025-03-31 01:58:45 status installed libaudit-common:all 1:4.0.2-2
2025-03-31 01:58:45 configure libsemanage-common:all 3.7-2.1 <none>
2025-03-31 01:58:45 status unpacked libsemanage-common:all 3.7-2.1
:|
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
Apr 12 18:25:01 kali CRON[30085]: pam_unix(cron:session): session opened for user root(uid=0) by root>
Apr 12 18:25:01 kali CRON[30087]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
Apr 12 18:25:01 kali CRON[30085]: pam_unix(cron:session): session closed for user root
Apr 12 18:29:56 kali sudo[32514]:      kali : TTY=pts/0 ; PWD=/home/kali/Desktop ; USER=root ; COMMAND>
Apr 12 18:29:56 kali sudo[32514]: pam_unix(sudo:session): session opened for user root(uid=0) by kali>
Apr 12 18:29:56 kali useradd[32521]: new group: name=Tyler, GID=1001
Apr 12 18:29:56 kali useradd[32521]: new user: name=Tyler, UID=1001, GID=1001, home=/home/Tyler, shel>
Apr 12 18:29:56 kali useradd[32526]: new group: name=Dade, GID=1002
Apr 12 18:29:56 kali useradd[32526]: new user: name=Dade, UID=1002, GID=1002, home=/home/Dade, shell=>
Apr 12 18:29:56 kali useradd[32531]: new group: name=Elliot, GID=1003
Apr 12 18:29:56 kali useradd[32531]: new user: name=Elliot, UID=1003, GID=1003, home=/home/Elliott, sh>
Apr 12 18:29:56 kali useradd[32536]: new group: name=Thomas, GID=1004
Apr 12 18:29:56 kali useradd[32536]: new user: name=Thomas, UID=1004, GID=1004, home=/home/Thomas, sh>
Apr 12 18:29:56 kali chpasswd[32542]: pam_unix(chpasswd:chauthtok): password changed for Tyler
Apr 12 18:29:56 kali chpasswd[32542]: gkr-pam: couldn't update the login keyring password: no old pas>
Apr 12 18:29:56 kali chpasswd[32548]: pam_unix(chpasswd:chauthtok): password changed for Dade
Apr 12 18:29:56 kali chpasswd[32548]: gkr-pam: couldn't update the login keyring password: no old pas>
Apr 12 18:29:56 kali chpasswd[32554]: pam_unix(chpasswd:chauthtok): password changed for Elliot
Apr 12 18:29:56 kali chpasswd[32554]: gkr-pam: couldn't update the login keyring password: no old pas>
Apr 12 18:29:56 kali chpasswd[32560]: pam_unix(chpasswd:chauthtok): password changed for Thomas
Apr 12 18:29:56 kali chpasswd[32560]: gkr-pam: couldn't update the login keyring password: no old pas>
Apr 12 18:29:56 kali usermod[32565]: add 'Elliot' to group 'sudo'
Apr 12 18:29:56 kali usermod[32565]: add 'Elliot' to shadow group 'sudo'
Apr 12 18:29:56 kali usermod[32570]: add 'Thomas' to group 'sudo'
Apr 12 18:29:56 kali usermod[32570]: add 'Thomas' to shadow group 'sudo'
Apr 12 18:29:57 kali sudo[32514]: pam_unix(sudo:session): session closed for user root
Apr 12 18:29:57 kali dbus-daemon[950]: [session uid=1000 pid=950 pidfd=5] Activating via systemd: ser>
Apr 12 18:29:57 kali systemd[922]: Starting tumblerd.service - Thumbnailing service ...
Apr 12 18:29:57 kali dbus-daemon[950]: [session uid=1000 pid=950 pidfd=5] Successfully activated serv>
Apr 12 18:29:57 kali systemd[922]: Started tumblerd.service - Thumbnailing service.
Apr 12 18:30:11 kali sudo[32726]:      kali : TTY=pts/0 ; PWD=/home/kali/Desktop ; USER=root ; COMMAND>
Apr 12 18:30:11 kali sudo[32726]: pam_unix(sudo:session): session opened for user root(uid=0) by kali>
Apr 12 18:30:11 kali sudo[32731]:      root : TTY=pts/1 ; PWD=/home/kali/Desktop ; USER=root ; COMMAND>
```



```

(kali㉿kali)-[~/Desktop]
$ sudo journalctl --since "10 minutes ago"
Apr 12 18:25:01 kali CRON[30085]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
Apr 12 18:25:01 kali CRON[30087]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1)
Apr 12 18:25:01 kali CRON[30085]: pam_unix(cron:session): session closed for user root(uid=0)
Apr 12 18:29:56 kali sudo[32514]:      kali : TTY=pts/0 ; PWD=/home/kali/Desktop ; USER=root ; COMMAND=
Apr 12 18:29:56 kali sudo[32514]: pam_unix(sudo:session): session opened for user root(uid=0) by root(uid=0)
Apr 12 18:29:56 kali useradd[32521]: new group: name=Tyler, GID=1001
Apr 12 18:29:56 kali useradd[32521]: new user: name=Tyler, UID=1001, GID=1001
Apr 12 18:29:56 kali useradd[32526]: new group: name=Dade, GID=1002
Apr 12 18:29:56 kali useradd[32526]: new user: name=Dade, UID=1002, GID=1002
Apr 12 18:29:56 kali useradd[32531]: new group: name=Elliot, GID=1003
Apr 12 18:29:56 kali useradd[32531]: new user: name=Elliot, UID=1003, GID=1003
Apr 12 18:29:56 kali useradd[32536]: new group: name=Thomas, GID=1004
Apr 12 18:29:56 kali useradd[32536]: new user: name=Thomas, UID=1004, GID=1004
Apr 12 18:29:56 kali chpasswd[32542]: pam_unix(chpasswd:chauthtok): password changed for user Tyler
Apr 12 18:29:56 kali chpasswd[32542]: gkr-pam: couldn't update the login keyring for user Tyler
Apr 12 18:29:56 kali chpasswd[32548]: pam_unix(chpasswd:chauthtok): password changed for user Dade
Apr 12 18:29:56 kali chpasswd[32548]: gkr-pam: couldn't update the login keyring for user Dade
Apr 12 18:29:56 kali chpasswd[32554]: pam_unix(chpasswd:chauthtok): password changed for user Elliot
Apr 12 18:29:56 kali chpasswd[32554]: gkr-pam: couldn't update the login keyring for user Elliot
Apr 12 18:29:56 kali chpasswd[32560]: pam_unix(chpasswd:chauthtok): password changed for user Thomas
Apr 12 18:29:56 kali chpasswd[32560]: gkr-pam: couldn't update the login keyring for user Thomas
Apr 12 18:29:56 kali usermod[32565]: add 'Elliot' to group 'sudo'
Apr 12 18:29:56 kali usermod[32565]: add 'Elliot' to shadow group 'sudo'
Apr 12 18:29:56 kali usermod[32570]: add 'Thomas' to group 'sudo'
Apr 12 18:29:56 kali usermod[32570]: add 'Thomas' to shadow group 'sudo'
Apr 12 18:29:57 kali sudo[32514]: pam_unix(sudo:session): session closed for user root(uid=0)
Apr 12 18:29:57 kali dbus-daemon[950]: [session uid=1000 pid=950 pidfd=5] A
Apr 12 18:29:57 kali systemd[922]: Starting tumblerd.service - Thumbnailing
Apr 12 18:29:57 kali dbus-daemon[950]: [session uid=1000 pid=950 pidfd=5] S
lines 1-29 ... skipping ...
Apr 12 18:25:01 kali CRON[30085]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
Apr 12 18:25:01 kali CRON[30087]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1)

```

```

(kali㉿kali)-[~/Desktop]
$ sudo ./lab3s3.sh

(kali㉿kali)-[~/Desktop]
$ sudo ./lab3s4.sh

(kali㉿kali)-[~/Desktop]
$ 

```

Conclusion

This report shows how log file analysis functions as a crucial method to detect and fight cybersecurity threats. The analysis of Windows Event Viewer and Kali Linux logs shows how

operating systems handle system activity recording and classification and presentation methods. The evaluation of system logs from both platforms exposed essential system changes which included user account generation and service modifications and privilege elevation activities that suggested malicious behavior.

Log file analysis tools including Event Viewer, Logwatch and Splunk were evaluated to demonstrate how native and advanced tools work together to streamline the detection of suspicious system activities. Although the Splunk installation presented problems the tool showed its value for real-time alerting and centralized logging which makes it important for enterprise cybersecurity operations.

The execution of scripts on Windows and Kali Linux systems demonstrated how attackers could use system vulnerabilities to achieve unauthorized access and maintain persistence. Log analysis serves as the critical method to identify these activities because it enables prompt incident response and system integrity maintenance.

The lab demonstrates that cybersecurity professionals need log file analysis to identify unauthorized activities and detect elevated privileges and backdoor access. IT environment security and stability depend on continuous monitoring together with efficient log management practices.