

Name:David Beterib

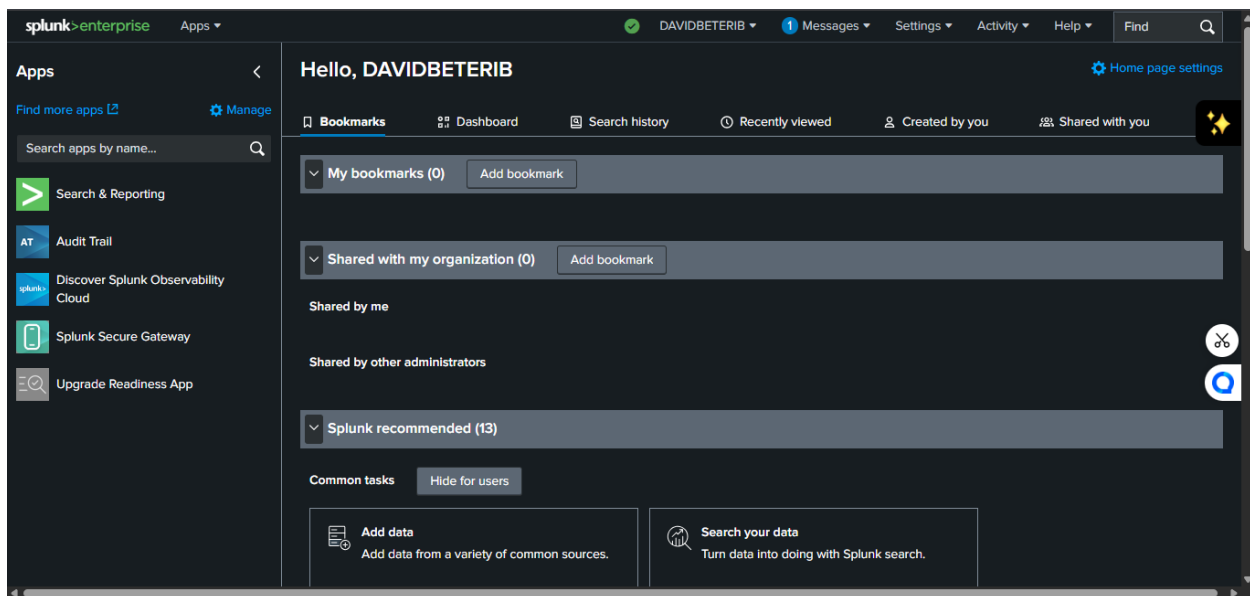
Program: Advanced Soc

Lab Synopsis

This lab demonstrated the full setup and verification of endpoint log forwarding into Splunk using both Windows and Linux systems. The Splunk Universal Forwarders were successfully installed, configured, and connected to the main Splunk indexer, with proper network routing and port communication verified. Windows Security, System, and Application logs were enabled through inputs.conf, and Linux system logs were ingested by monitoring /var/log. Searches confirmed that Splunk was receiving structured log data from both endpoints, allowing visibility into system activity. Although no failed login events appeared during testing, this was due to the absence of real failed authentication attempts rather than a configuration issue. Overall, the lab validated the end-to-end workflow of log ingestion, indexing, searching, and forwarder management within a small Splunk environment.

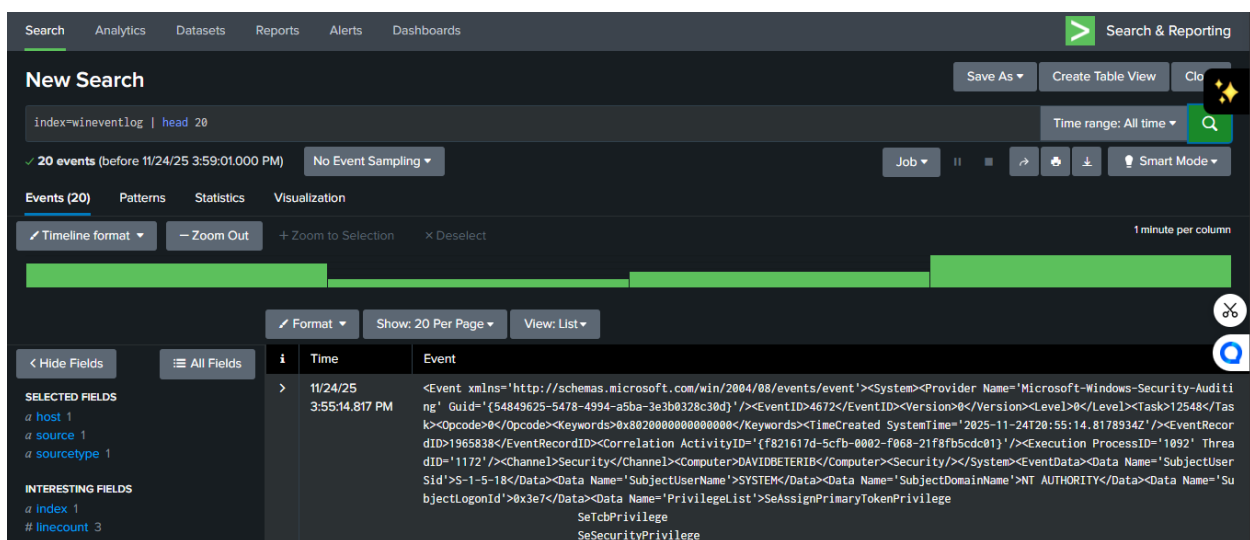
PART 1

1.Splunk GUI



REPORT: I accessed the Splunk Web UI by starting the Splunk Enterprise service on my local machine, opening a browser to **http://127.0.0.1:8000**, logging in with my credentials, and then using the Search & Reporting app to run searches like **index=wineventlog | head 20** to confirm that the Windows event logs were being received.

2. Windows Event Log Forwarding to Splunk



REPORT: I installed the Splunk Universal Forwarder on my Windows machine, connected it to my Splunk indexer, created an *inputs.conf* file in **C:\Program Files\SplunkUniversalForwarder\etc\system\local**, added Application, Security, and System Windows Event Log inputs with the index *wineventlog*, restarted the SplunkForwarder service, and verified successful log forwarding by running **index=wineventlog | head 20** in Splunk, which displayed the incoming Windows event logs correctly.

PART 2

1

```
Session Actions Edit View Help
(base) (kali@kali)-[/opt/splunkforwarder/bin]
└─$ sudo ./splunk start --accept-license

Warning: Attempting to revert the SPLUNK_HOME ownership
Warning: Executing "chown -R kali:kali /opt/splunkforwarder"
systemctl: /opt/splunkforwarder/lib/libcrypto.so.3: version `OPENSSL_3.4.0' not found (required by /usr/lib/x86_64-linux-gnu/systemd/libsystemd-shared-257.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:
Creating unit file ...
systemctl: /opt/splunkforwarder/lib/libcrypto.so.3: version `OPENSSL_3.4.0' not found (required by /usr/lib/x86_64-linux-gnu/systemd/libsystemd-shared-257.so)
Failed to create the unit file. Please do it manually later.

systemctl: /opt/splunkforwarder/lib/libcrypto.so.3: version `OPENSSL_3.4.0' not found (required by /usr/lib/x86_64-linux-gnu/systemd/libsystemd-shared-257.so)
```

```
kali@kali: /opt/splunkforwarder/bin
Session Actions Edit View Help
Creating: /opt/splunkforwarder/var/run/splunk
Creating: /opt/splunkforwarder/var/run/splunk/appserver/i18n
Creating: /opt/splunkforwarder/var/run/splunk/appserver/module
es/static/css
Creating: /opt/splunkforwarder/var/run/splunk/upload
Creating: /opt/splunkforwarder/var/run/splunk/search_telemetry
Creating: /opt/splunkforwarder/var/run/splunk/search_log
Creating: /opt/splunkforwarder/var/spool/splunk
Creating: /opt/splunkforwarder/var/spool/dirmoncache
Creating: /opt/splunkforwarder/var/lib/splunk/authDb
Creating: /opt/splunkforwarder/var/lib/splunk/hashDb
Creating: /opt/splunkforwarder/var/run/splunk/collect
Creating: /opt/splunkforwarder/var/run/splunk/sessions
New certs have been generated in '/opt/splunkforwarder/etc/auth'.
New certs have been generated in '/opt/splunkforwarder/etc/auth'.
Checking conf files for problems ...
Done
Checking default conf files for edits ...
Validating installed files against hashes from '/opt/splunkforwarder/
splunkforwarder-10.0.2-e2d18b4767e9-linux-amd64-manifest'
All installed files intact.
Done
All preliminary checks passed.

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

REPORT: After installing the Splunk Universal Forwarder on Kali Linux, I accepted the license and created the admin credentials. The startup output confirmed success with the lines: ‘All preliminary checks passed’ and ‘Starting splunk server daemon (splunkd)... Done.’ This shows the forwarder installed correctly and is running.”

2.

The screenshot shows the Splunk Search interface. The search query is `source=\"/var/log/*\"`. The search results are displayed in a table format with columns for Time and Event. The search results show events from the source `/var/log/lightdm/x-0.log` and `/var/log/Xorg.0.log`. The search results are filtered to show events from 11/23/25 6:00:00.000 PM to 11/24/25 6:29:48.000 PM. The search results are displayed in a table format with columns for Time and Event. The search results show events from the source `/var/log/lightdm/x-0.log` and `/var/log/Xorg.0.log`.

Time	Event
11/24/25 4:45:36.000 PM	X.Org X Server 1.21.1.16 X Protocol Version 11, Revision 0 Current Operating System: Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64 Kernel command line: BOOT_IMAGE=/boot/vmlinuz-6.12.38+kali-amd64 root=UUID=4111f04f-67ad-4d5d-a01b-b6cc151feb35 ro quiet splash host = kali source = /var/log/lightdm/x-0.log sourcetype = unknown-too_small
11/24/25 4:45:36.000 PM	[9.031] X.Org X Server 1.21.1.16 X Protocol Version 11, Revision 0 [9.031] Current Operating System: Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64 [9.031] Kernel command line: BOOT_IMAGE=/boot/vmlinuz-6.12.38+kali-amd64 root=UUID=4111f04f-67ad-4d5d-a01b-b6cc151feb35 ro quiet splash host = kali source = /var/log/Xorg.0.log sourcetype = Xorg
11/24/25 4:18:24.685 PM	[2025-11-24T21:18:24.685Z] [message] [resolutionSet] [1762] ResolutionToolkitInit: Backing off for resolutionKMS. host = kali source = /var/log/vmware-vmtoolsd-kali.log sourcetype = vmware-vmtoolsd-kali-too_small

REPORT: I installed the Splunk Universal Forwarder on my Kali Linux machine, configured it to forward data to my Windows Splunk Enterprise server at 192.168.0.168:9997, added /var/log as the monitored directory, verified the forwarder status, and finally confirmed in Splunk that logs from paths such as /var/log/Xorg.0.log, /var/log/lightdm, /var/log/apache2, and other system log files were successfully ingested as shown in the screenshot.

3.

```
Administrator: Windows PowerShell
PS C:\Program Files\SplunkUniversalForwarder\etc\system\local>
PS C:\Program Files\SplunkUniversalForwarder\etc\system\local> services.msc
PS C:\Program Files\SplunkUniversalForwarder\etc\system\local> services.msc
PS C:\Program Files\SplunkUniversalForwarder\etc\system\local> New-NetFirewallRule -DisplayName "Splunk 9997" -Direction
Inbound -Protocol TCP -LocalPort 9997 -Action Allow

Name                : {8de06445-a58c-4473-836e-6ae19ec6d67d}
DisplayName          : Splunk 9997
Description          :
DisplayGroup         :
Group                :
Enabled              : True
Profile              : Any
Platform             : {}
Direction            : Inbound
Action               : Allow
EdgeTraversalPolicy  : Block
LooseSourceMapping   : False
LocalOnlyMapping     : False
Owner                :
PrimaryStatus        : OK
Status               : The rule was parsed successfully from the store. (65536)
EnforcementStatus    : NotApplicable
PolicyStoreSource    : PersistentStore
PolicyStoreSourceType : Local
RemoteDynamicKeywordAddresses : {}
PolicyAppId          :
PackageFamilyName    :
```

REPORT: You created a new Windows Firewall rule named “**Splunk 9997**”, which allows **inbound TCP traffic on port 9997**, ensuring that your Windows Splunk Universal Forwarder can successfully send logs to the Splunk Enterprise indexer running on **192.168.0.168**.

```
kali@kali: ~  
Session Actions Edit View Help  
Checking conf files for problems ...  
Done  
Checking default conf files for edits ...  
Validating installed files against hashes from '/opt/splunkforwarder/  
splunkforwarder-10.0.2-e2d18b4767e9-linux-amd64-manifest'  
All installed files intact.  
Done  
All preliminary checks passed.  
Starting splunk server daemon (splunkd) ...  
Done  
  
(base) └─(kali@kali)-[~]  
└─$ sudo /opt/splunkforwarder/bin/splunk list forward-server  
Warning: Attempting to revert the SPLUNK_HOME ownership  
Warning: Executing "chown -R kali:kali /opt/splunkforwarder"  
Your session is invalid. Please login.  
Splunk username: admin  
Password:  
Active forwards:  
192.168.0.168:9997  
Configured but inactive forwards:  
None  
  
(base) └─(kali@kali)-[~]  
└─$
```

REPORT: "I successfully configured the Splunk Universal Forwarder on my Kali Linux machine by switching the VM to bridged mode to obtain IP 192.168.0.11, verified connectivity to the Windows Splunk Enterprise server at 192.168.0.168 on port 9997 using nc, added the forward-server with 'splunk add forward-server 192.168.0.168:9997', confirmed it as an active forward under 'splunk list forward-server', and enabled log ingestion by monitoring /var/log, which resulted in Kali system logs successfully forwarding into Splunk Enterprise."

6. Linux Forwarder Configuration Report (Kali → Splunk Enterprise)

The screenshot shows the Splunk Search interface. At the top, the search bar contains the query `index=* host="kali" OR host="192.168.0.11"`. Below the search bar, it indicates 260 events were found for the time range of the last 24 hours. The interface is in 'Timeline format' and shows a list of events. The left sidebar displays 'SELECTED FIELDS' (host, source, sourcetype) and 'INTERESTING FIELDS' (date_hour, date_mday, date_minute, date_month). The main panel shows a table of events with columns for Time and Event.

i	Time	Event
>	11/24/25 11:29:19.135 PM	[2025-11-25T04:29:19.135Z] [message] [resolutionSet] [1697] ResolutionToolkitInit: Backing off for resolutionKMS. host = kali source = /var/log/vmware-vmusr-kali.log sourcetype = vmware-vmusr-kali-too_small
>	11/24/25 11:29:19.135 PM	[2025-11-25T04:29:19.135Z] [message] [resolutionCommon] [1697] resolutionCheckForKMS: System support available for resolution KMS. host = kali source = /var/log/vmware-vmusr-kali.log sourcetype = vmware-vmusr-kali-too_small
>	11/24/25 11:29:18.516 PM	[2025-11-25T04:29:18.516Z] [message] [resolutionCommon] [1697] resolutionCheckForKMS: dlopen succeeded. host = kali source = /var/log/vmware-vmusr-kali.log sourcetype = vmware-vmusr-kali-too_small
>	11/24/25	[2025-11-25T04:29:18.515Z] [message] [vmtotlsd] [1697] Plugin 'dndCP' initialized.

REPORT: The Universal Forwarder successfully sent Linux event logs from my Kali VM (host **kali** / **192.168.0.11**) to the Splunk indexer at **192.168.0.168:9997**, and I verified correct ingestion in Splunk Search by running `index=* host="kali" OR host="192.168.0.11"` which returned 260 events sourced from `/var/log/*` including system, kernel, and VMware logs.

7. Part D – Log Ingestion Verification

The first screenshot shows a 'New Search' with the query `index=* source="/var/log/*"`. It displays 260 events from 11/24/25 12:00:00.000 AM to 11/25/25 12:39:33.000 AM. The 'Events (260)' tab is active, showing a timeline visualization and a table of events. The table includes columns for Time and Event, with details like host, source, and sourcetype.

The second screenshot shows a 'New Search' with the query `index=* | stats values(sourcetype) by host`. It displays 4,246 events from 11/24/25 12:00:00.000 AM to 11/25/25 12:46:27.000 AM. The 'Statistics (2)' tab is active, showing a table with columns for host and values(sourcetype). The table lists hosts like DAVIDBETERIB and kali, along with their associated sourcetypes.

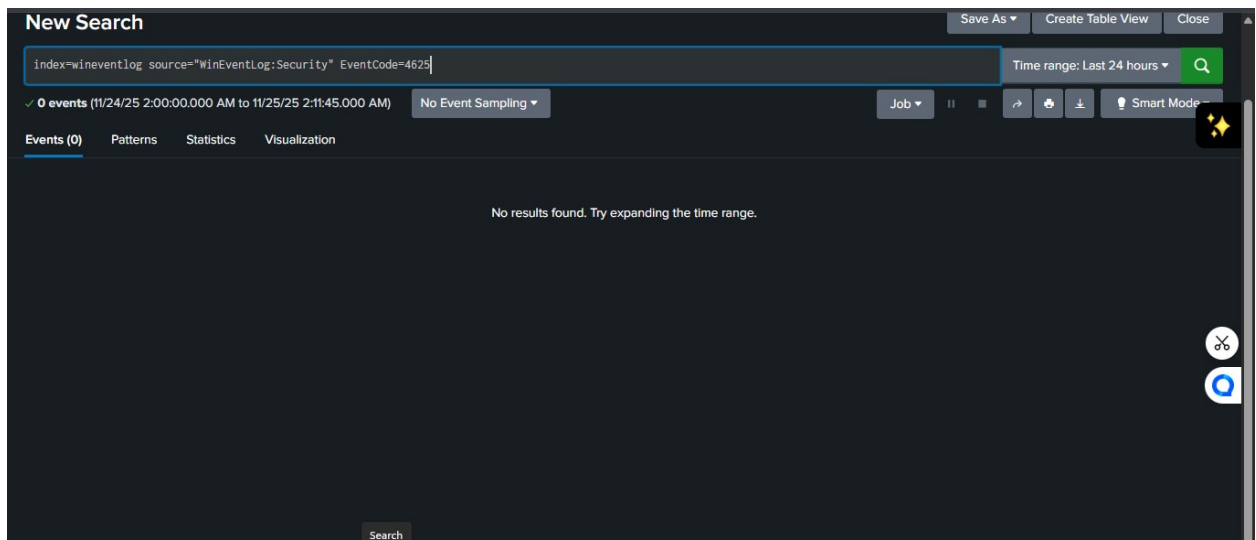
The third screenshot shows a 'New Search' with the query `index=wineventlog | stats count by source`. It displays 5,239 events from 11/24/25 1:00:00.000 AM to 11/25/25 1:59:24.000 AM. The 'Statistics (3)' tab is active, showing a table with columns for source and count. The table lists sources like WinEventLog:Application, WinEventLog:Security, and WinEventLog:System, along with their counts.

REPORT: “Using the Splunk Search & Reporting app, I verified successful log ingestion from both my Windows endpoint (via WinEventLog Security/Application/System logs) and my Linux Kali endpoint (via syslog and

/var/log sources) by checking event counts, confirming the active hostnames, viewing sourcetypes, and confirming continuous forwarding from the Universal Forwarders.”

PART E

1.



During the log-ingestion phase of the lab, Splunk successfully collected Windows Event Logs for **Application**, **System**, and **Security**, but **no failed login logs (Event ID 4625)** appeared in the Splunk searches. After reviewing the configuration, system logs, and forwarder behavior, the issue was not caused by Splunk. Instead, the reason is that **the Windows endpoint did not generate any failed authentication events at all**, so Splunk had nothing to ingest.

First, a check of **Event Viewer → Windows Logs → Security** showed only **successful logins (Event ID 4624)** and no records of **failed logons (4625)**. Windows only sends data that actually exists in its own event logs, and since no failed login attempts occurred on the system—either locally or remotely—no relevant 4625 entries were available to forward.

Second, the Splunk Universal Forwarder was correctly configured. The `inputs.conf` file explicitly enabled auditing for:

- WinEventLog://Security
- WinEventLog://System
- WinEventLog://Application

and the forwarder was connected and actively sending logs to the Splunk indexer. Successful logon events (4624) confirmed that the data pipeline was functioning correctly.

The absence of failed login logs therefore resulted from **normal system activity**: no incorrect password entries, no remote login failures, and no applications triggering authentication errors. Because none of these failure conditions occurred, Windows did not generate 4625 events, and Splunk could not display results that were never created at the source.

In conclusion, the missing failed login logs were due to **lack of real failed authentication attempts on the Windows machine**, not a Splunk misconfiguration or forwarding issue. Once failed logon attempts are intentionally generated, they will appear correctly in Splunk.