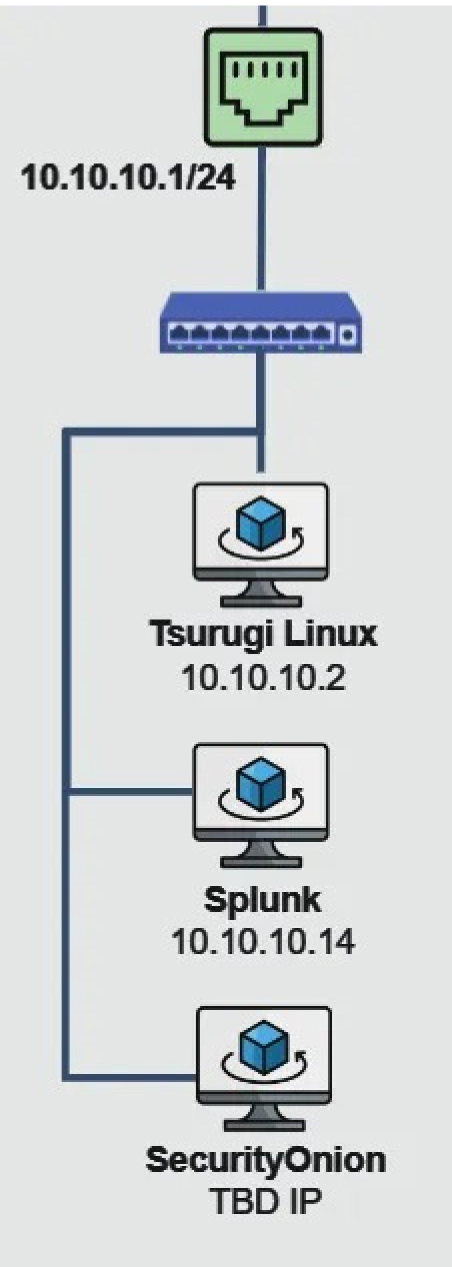
Log Analysis using Splunk

In the final installment of my Home Lab Series, I'm diving into the setup and configuration of Splunk, a powerful Security Information and Event Management (SIEM) tool. Splunk will be installed on a Ubuntu VM within my SECURITY subnet, and I'll also configure the Splunk Universal Forwarder on my Windows Server 2019 Domain Controller (DC) to ingest logs. This setup will allow me to monitor and analyze security events within my home lab, providing critical insights into network activity and potential threats.



Why Splunk?

Splunk is widely used in cybersecurity for its ability to collect, analyze, and visualize vast amounts of data from various sources. It's targeted for:

- **Centralized Log Management**: Aggregating logs from multiple sources makes monitoring and analyzing system and network activity easier.
- **Real-Time Monitoring**: Detecting and responding to security incidents as they happen, thanks to Splunk's powerful alerting and reporting features.
- **Advanced Analytics**: Using machine learning and custom queries to identify trends, anomalies, and potential security threats.

Integrating Splunk into my home lab will significantly enhance my ability to monitor the environment, detect unusual activity, and improve my overall cybersecurity posture.

Ubuntu VM Setup for Splunk

Downloading Ubuntu

To get started, I downloaded the latest LTS version of Ubuntu from the Ubuntu Downloads page. At the time of writing, the newest version available was Ubuntu 22.04.3, and the ISO file was approximately 5GB.

Creating the Ubuntu VM

With the ISO downloaded, I proceeded to create the Ubuntu VM:

Create a New VM:

- I opened VirtualBox, clicked on “New,” and named the VM (e.g., “Ubuntu Splunk”).
- I selected the Ubuntu ISO and “Skip Unattended Installation” to configure the VM manually.

Configure VM Settings:

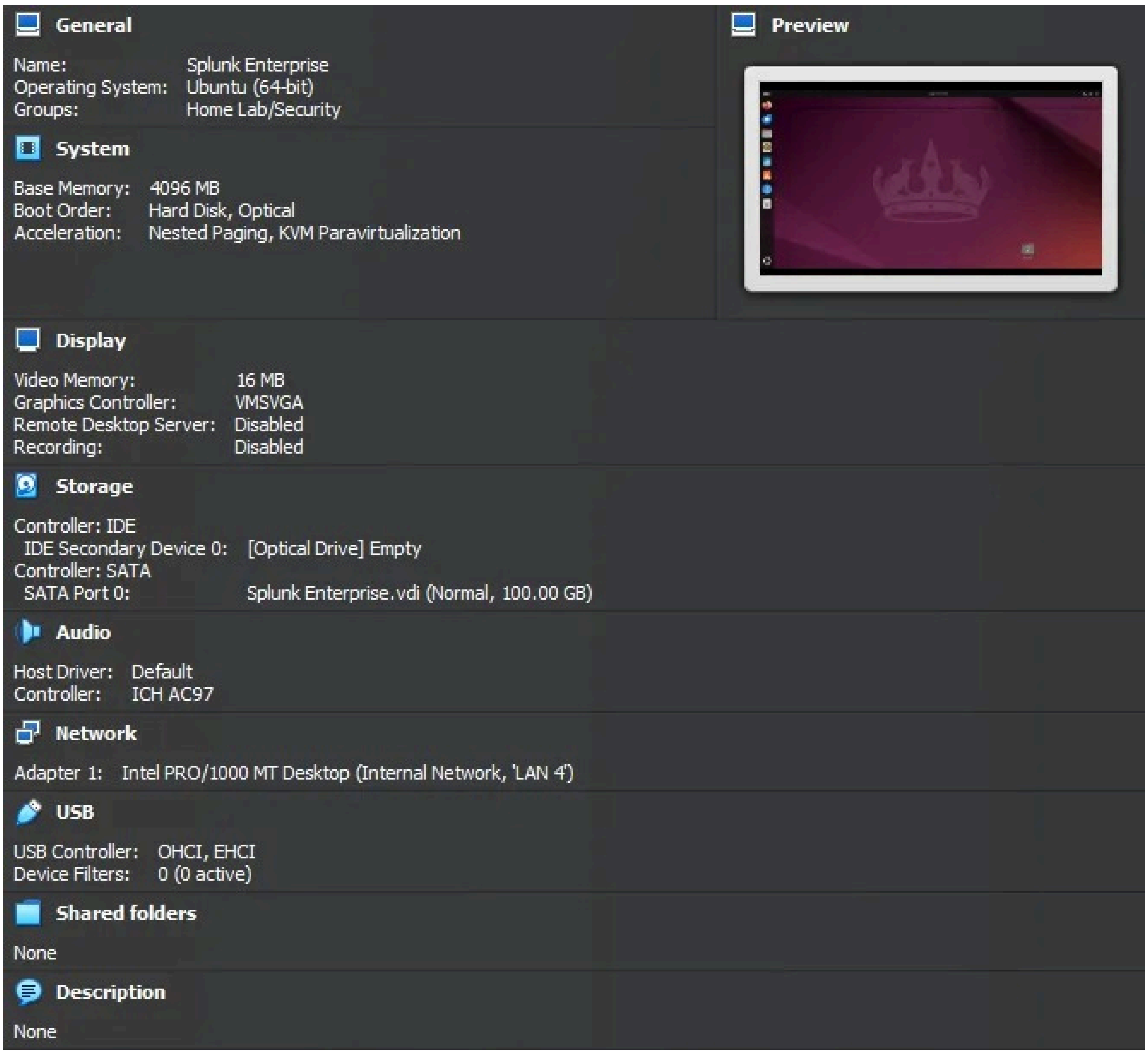
- **Memory**: I allocated 4096MB (4GB) of RAM.
- **Hard Disk**: I set the hard disk size to 100GB, providing needed space for Splunk and the logs it will collect.

Network Configuration:

- I connected the VM’s network adapter to the “LAN 4” interface, part of my SECURITY subnet.

Finalize and Start the VM:

- After reviewing the settings, I clicked “Finish” to create the VM.



General

Name: Splunk Enterprise
Operating System: Ubuntu (64-bit)
Groups: Home Lab/Security

System

Base Memory: 4096 MB
Boot Order: Hard Disk, Optical
Acceleration: Nested Paging, KVM Paravirtualization

Preview

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Splunk Enterprise.vdi (Normal, 100.00 GB)

Audio

Host Driver: Default
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (Internal Network, 'LAN 4')

USB

USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Shared folders

None

Description

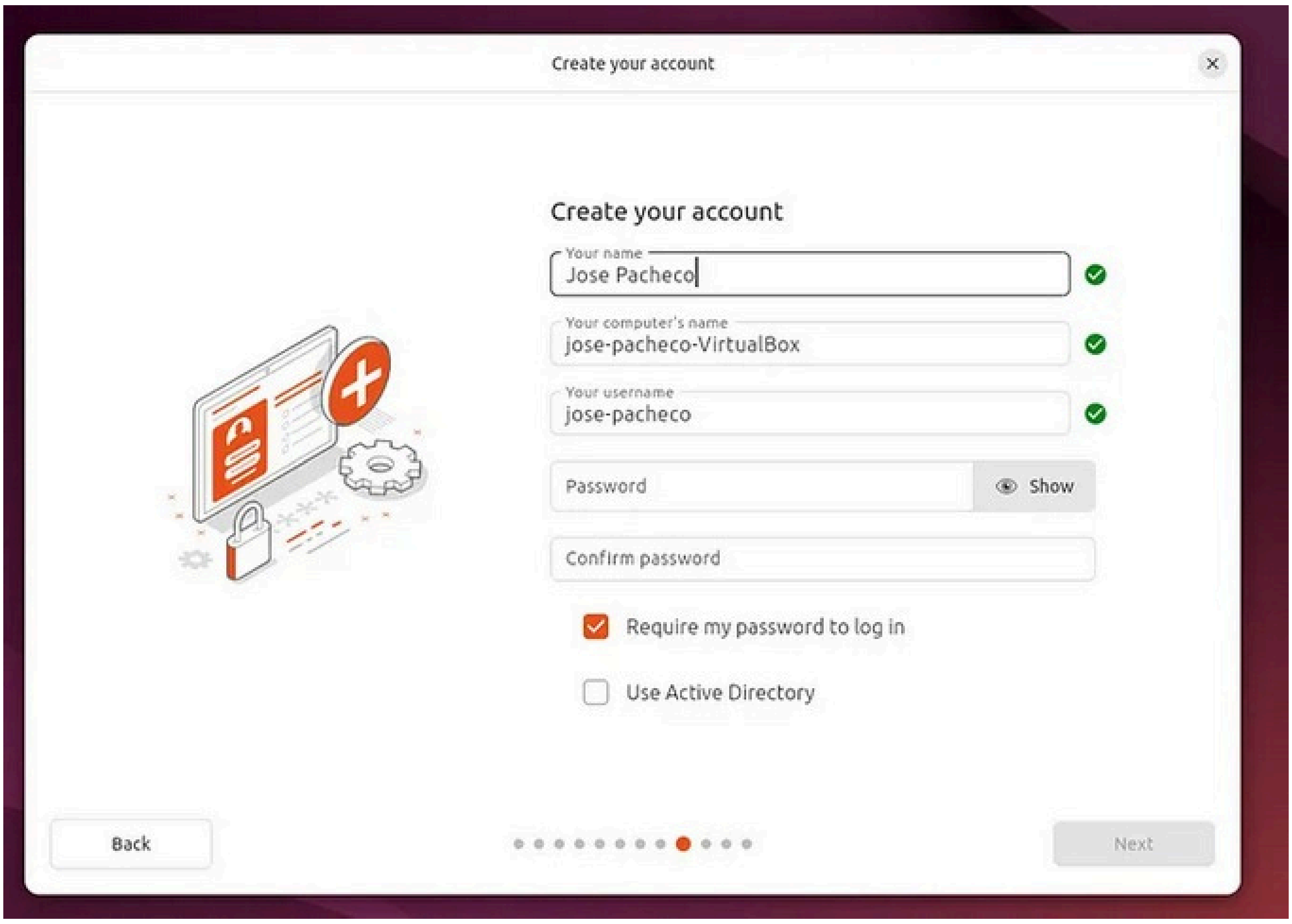
None

Installing Ubuntu

With the VM set up, I installed Ubuntu:

1. **Boot the VM** and launch the graphical installer.

2. **Follow the installation prompts** to select a language keyboard layout and configure the system. I opted to install third-party software for better compatibility.
3. **Set up the user account** and hostname, then complete the installation.
4. **Restart the VM** and log in to finish the setup.



Create your account

Create your account

Your name: Jose Pacheco

Your computer's name: jose-pacheco-VirtualBox

Your username: jose-pacheco

Password

Confirm password

☑ Require my password to log in

□ Use Active Directory

Post-Installation Configuration

Once Ubuntu was running, I performed a few additional configurations:

Install VirtualBox Guest Additions:

- This improves performance and allows for dynamic resizing of the VM display.
- I mounted the Guest Additions CD image and ran the installation script from the terminal:

```
sudo ./VBoxLinuxAdditions.run
```

Update the System:

- Keeping the system updated is crucial, so I ran:

```
sudo apt update && sudo apt full-upgrade
```

Create a Snapshot:

- I took a snapshot of the VM, which is in its clean, fully updated state. This will allow me to revert to this baseline if needed.

Splunk Installation

Downloading Splunk

Next, I downloaded Splunk Enterprise:

Visit the Splunk Website:

- I went to the Splunk Enterprise Free Trial page and registered an account.
- After logging in, I downloaded the .deb package for Splunk Enterprise. The latest version available was 9.1.2.

Installing Splunk on Ubuntu

With the .deb file downloaded, I installed Splunk:

Install Dependencies:

- I opened the terminal and installed curl, which Splunk requires:

```
sudo apt install curl
```

Install Splunk:

- I navigated to the Downloads folder and ran:

```
sudo dpkg -i splunk-9.1.2-b6b9c8185839-linux-2.6-amd64.deb
```

- After installation, I started Splunk and accepted the license agreement:

```
sudo /opt/splunk/bin/splunk start - accept-license - answer-yes
```

- I set up the admin username and password as prompted.

Enable Splunk to Start at Boot:

- To ensure Splunk starts automatically whenever the VM is booted, I ran the following:

```
sudo /opt/splunk/bin/splunk enable boot-start
```

Create a Snapshot:

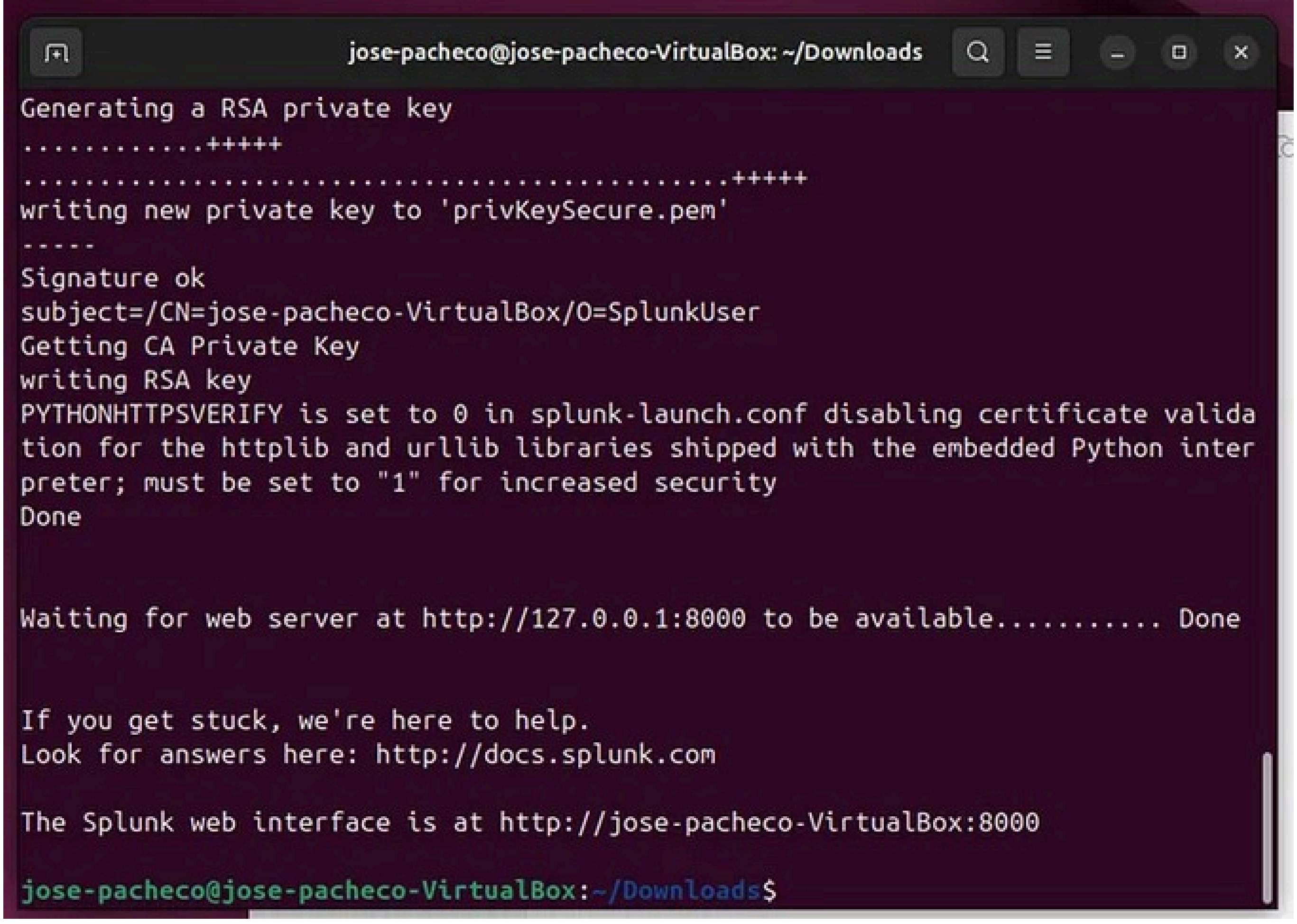
- After verifying the installation, I shut down the VM and took a snapshot, preserving this state for future use.

Splunk Final Snapshot

Configuring Splunk for Data Ingestion

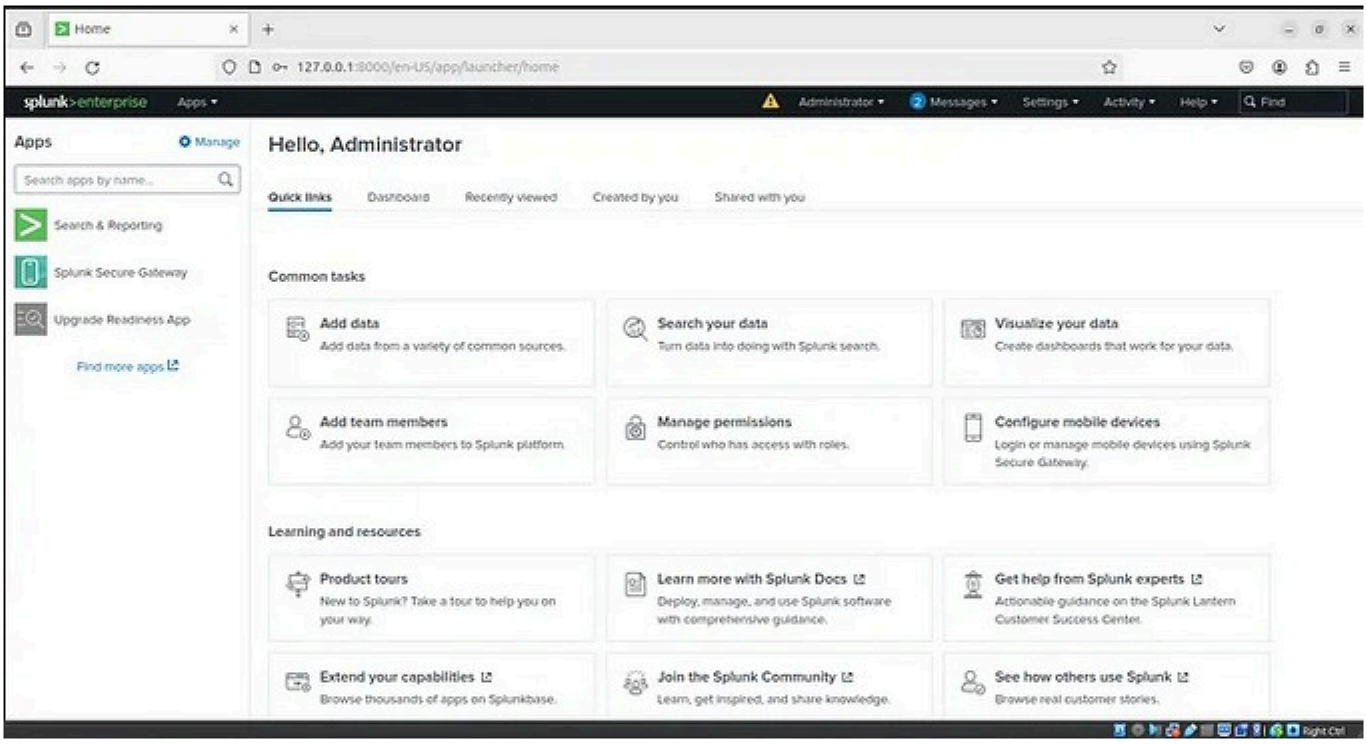
Setting Up Splunk to Receive Data

Before I could start ingesting logs from my Windows Server 2019 DC, I needed to configure Splunk to receive data:



```
Generating a RSA private key
............+++++
..............................................+++++
writing new private key to 'privKeySecure.pem'
-----
Signature ok
subject=/CN=jose-pacheco-VirtualBox/O=SplunkUser
Getting CA Private Key
writing RSA key
PYTHONHTTPSVERIFY is set to 0 in splunk-launch.conf disabling certificate valida
tion for the httplib and urllib libraries shipped with the embedded Python inter
preter; must be set to "1" for increased security
Done


Waiting for web server at http://127.0.0.1:8000 to be available.......... Done


If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://jose-pacheco-VirtualBox:8000

jose-pacheco@jose-pacheco-VirtualBox:~/Downloads$
```

- http://127.0.0.1:8000



Hello, Administrator

Common tasks

Add data
Add data from a variety of common sources.

Search your data
Turn data into doing with Splunk search.

Visualize your data
Create dashboards that work for your data.

Add team members
Add your team members to Splunk platform.

Manage permissions
Control who has access with roles.

Configure mobile devices
Login or manage mobile devices using Splunk Secure Gateway.

Learning and resources

Product tours
New to Splunk? Take a tour to help you on your way.

Learn more with Splunk Docs
Deploy, manage, and use Splunk software with comprehensive guidance.

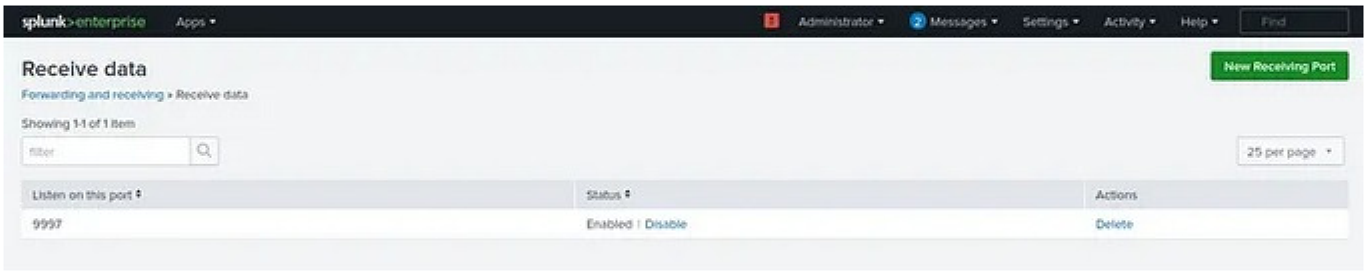
Get help from Splunk experts
Actionable guidance on the Splunk Lantern Customer Success Center.

Extend your capabilities
Browse thousands of apps on Splunkbase.

Join the Splunk Community
Learn, get inspired, and share knowledge.

See how others use Splunk
Browse real customer stories.

-
-



Receive data

Forwarding and receiving » Receive data

Showing 1-1 of 1 item

Listen on this port	Status	Actions
9997	Enabled | Disable	Delete

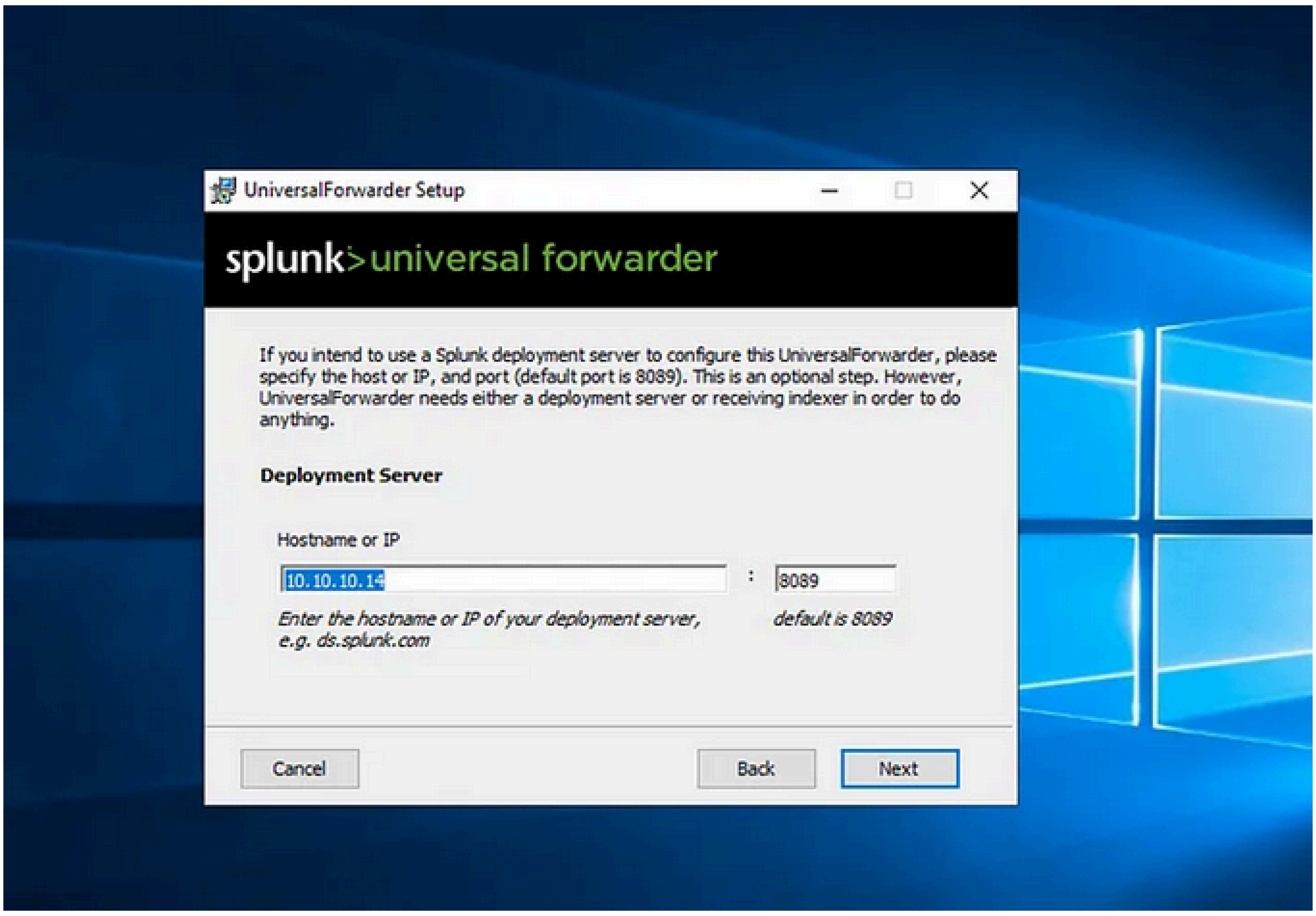
Data Receiving

Download Universal Forwarder:

- I downloaded the Windows version of the Splunk Universal Forwarder from the Splunk Download page.

Install the Forwarder:

- I ran the .msi installer, accepted the license agreement, and entered the Splunk admin credentials during the setup.
- I configured the Universal Forwarder with the IP address of my Splunk VM (found using ip a) and specified ports 8089 and 9997 for communication.

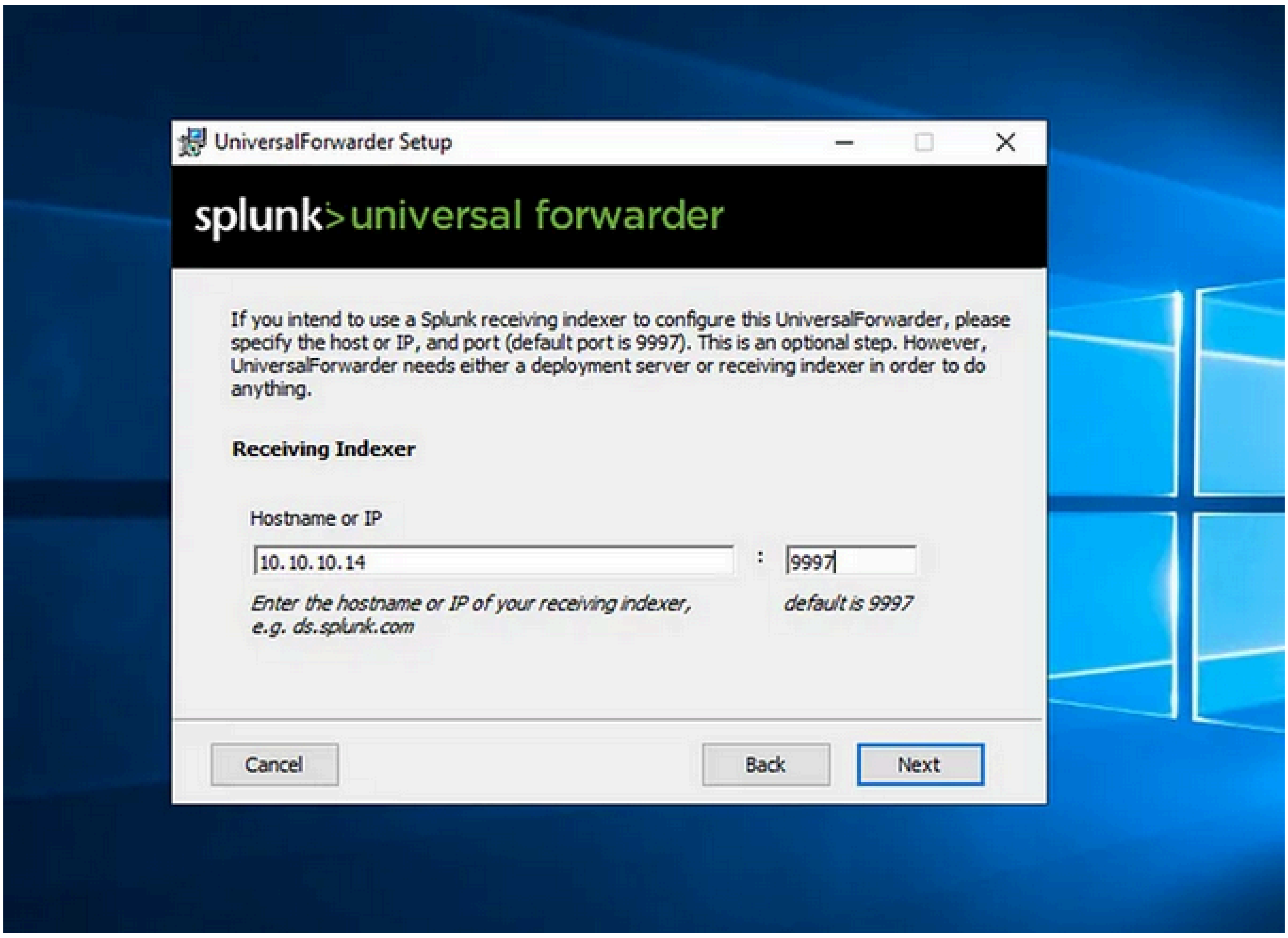


If you intend to use a Splunk deployment server to configure this UniversalForwarder, please specify the host or IP, and port (default port is 8089). This is an optional step. However, UniversalForwarder needs either a deployment server or receiving indexer in order to do anything.

Deployment Server

Hostname or IP: 10.10.10.14 : 8089

Enter the hostname or IP of your deployment server, e.g. ds.splunk.com *default is 8089*



If you intend to use a Splunk receiving indexer to configure this UniversalForwarder, please specify the host or IP, and port (default port is 9997). This is an optional step. However, UniversalForwarder needs either a deployment server or receiving indexer in order to do anything.

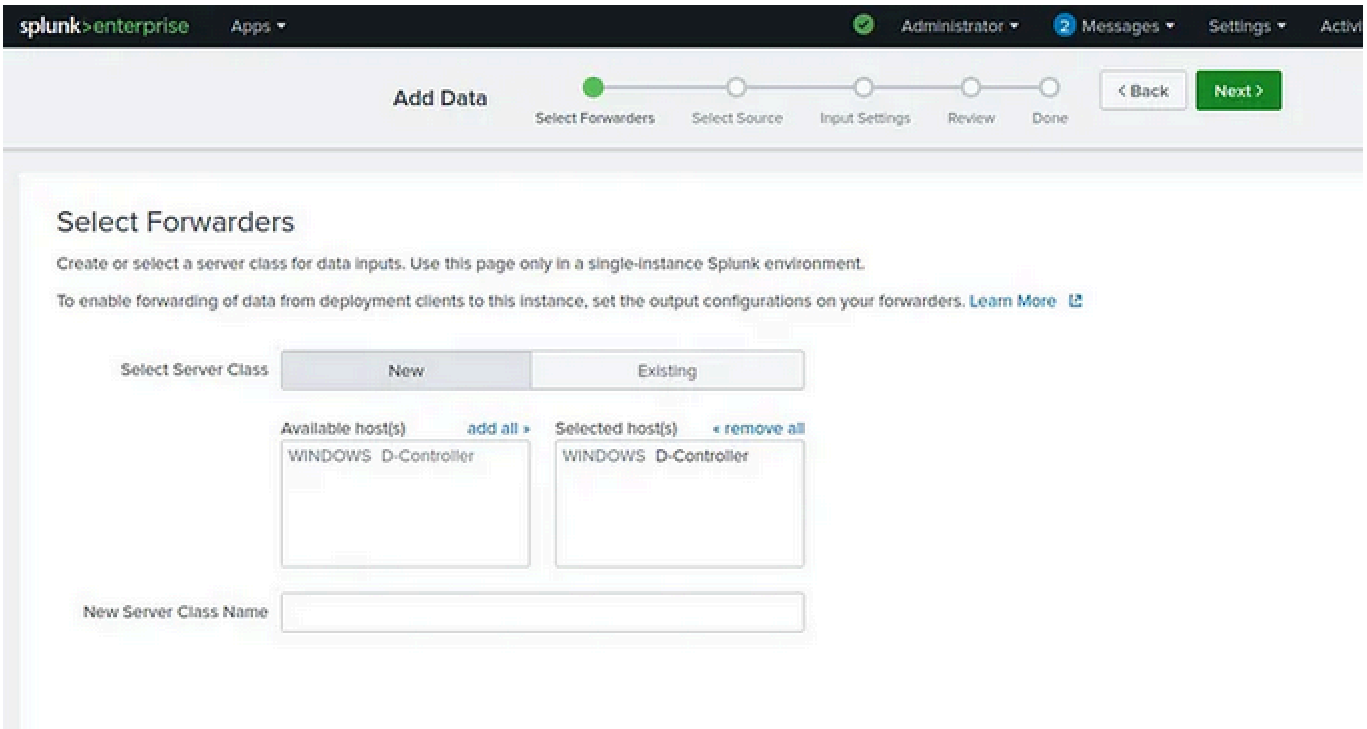
Receiving Indexer

Hostname or IP: 10.10.10.14 : 9997

Enter the hostname or IP of your receiving indexer, e.g. ds.splunk.com *default is 9997*

Add Data Source in Splunk:

- In Splunk, I went to **Settings > Add Data** and selected “Forward.”
- I added my D-Controller VM as a data source and configured it to send all local event logs to Splunk.
- I created a new index named “Windows” to store the logs.



Select Forwarders

Create or select a server class for data inputs. Use this page only in a single-instance Splunk environment.

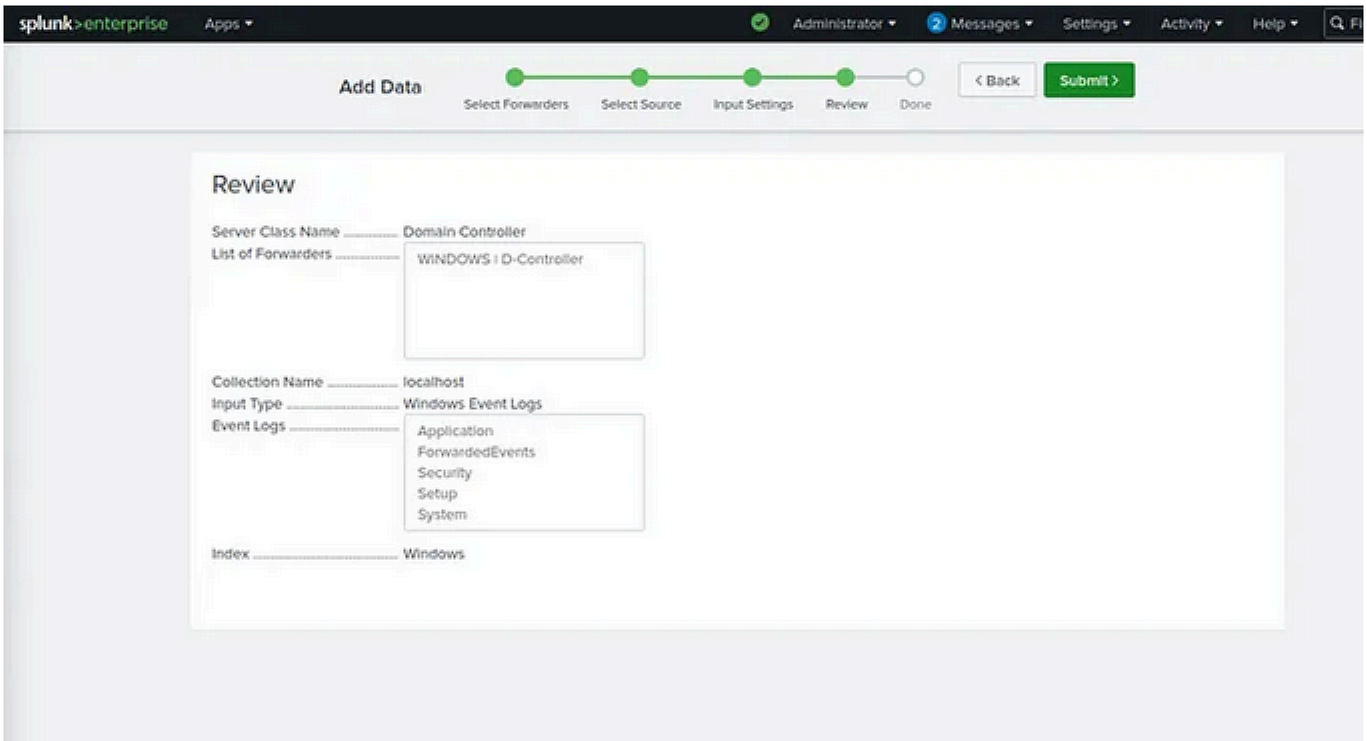
To enable forwarding of data from deployment clients to this instance, set the output configurations on your forwarders. Learn More

Select Server Class: New | Existing

Available host(s) add all » : WINDOWS D-Controller

Selected host(s) « remove all : WINDOWS D-Controller

New Server Class Name:



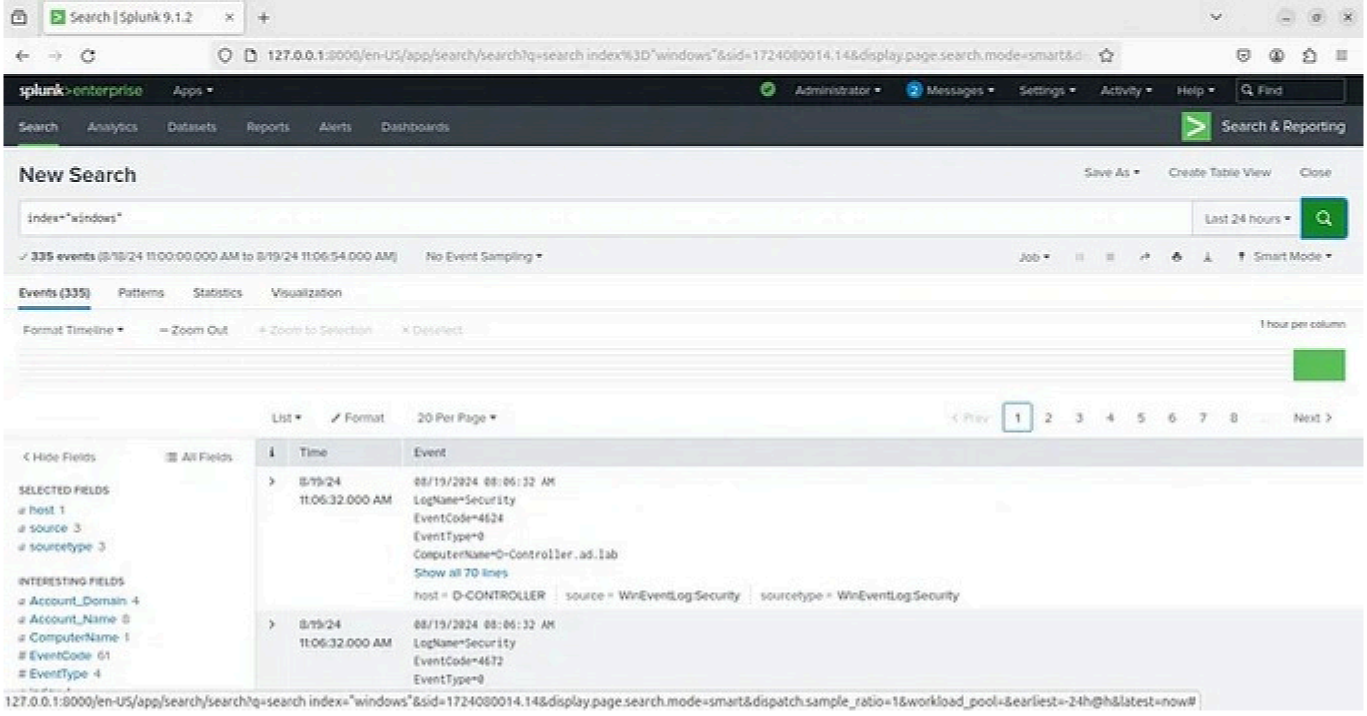
Review

Server Class Name .... Domain Controller
List of Forwarders .... WINDOWS | D-Controller
Collection Name .... localhost
Input Type .... Windows Event Logs
Event Logs .... Application
ForwardedEvents
Security
Setup
System
Index .... Windows

Verify Data Ingestion:

- In Splunk, I navigated to **Apps > Search & Reporting** and ran a search query:

```
index= "windows"
```



```
index="windows"
```

✓ 335 events (8/18/24 11:00:00.000 AM to 8/19/24 11:06:54.000 AM)   No Event Sampling

i	Time	Event

>	8/19/24
	11:06:32.000 AM
		08/19/2024 08:06:32 AM
		LogName=Security
		EventCode=4624
		EventType=0
		ComputerName=D-Controller.ad.lab
		Show all 70 lines
		host = D-CONTROLLER source = WinEventLog:Security sourcetype = WinEventLog:Security

>	8/19/24
	11:06:32.000 AM
		08/19/2024 08:06:32 AM
		LogName=Security
		EventCode=4672
		EventType=0

127.0.0.1:8000/en-US/app/search/search?q=search index="windows"&sid=1724080014.14&display.page.search.mode=smart&dispatch.sample_ratio=1&workload_pool=&earliest=-24h@h&latest=now#

- I verified that logs were being received. If no data appeared, I performed some actions on the DC VM to generate logs and then checked again.

Conclusion

Setting up Splunk in my home lab has significantly enhanced my ability to monitor and analyze security events. With Splunk now collecting and indexing logs from my Windows Server 2019 DC, I have a powerful tool to help detect potential threats, understand network activity, and improve my overall cybersecurity posture.

This final step in my home lab series completes the foundation of a comprehensive, self-contained cybersecurity lab. Each component — from the firewall and segmentation to the forensic tools and SIEM — works together to create a robust environment for learning and practicing essential cybersecurity skills.