

Hi, my name is Rafael José. Welcome to my miniSOC documentation!

1. Installation

I decided to choose Splunk for my miniSOC environment. In my case I will be installing Splunk in Linux.

We go to the official page of Splunk, log in and download the version that fits our OS.

GET STARTED

Choose Your Download

Splunk Enterprise 9.0.3

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.

Choose Your Installation Package

Windows Linux Mac OS

64-bit	3.x+, 4.x+, or 5.4.x kernel Linux distributions	.rpm	573.01 MB	Download Now
		.tgz	572.67 MB	Download Now
		.deb	444.7 MB	Download Now

In my case I downloaded the .deb package. I used dpkg to install it.

```
dpkg -i splunk-9.0.3-dd0128b1f8cd-linux-2.6-amd64.deb
```

2. Setting Splunk

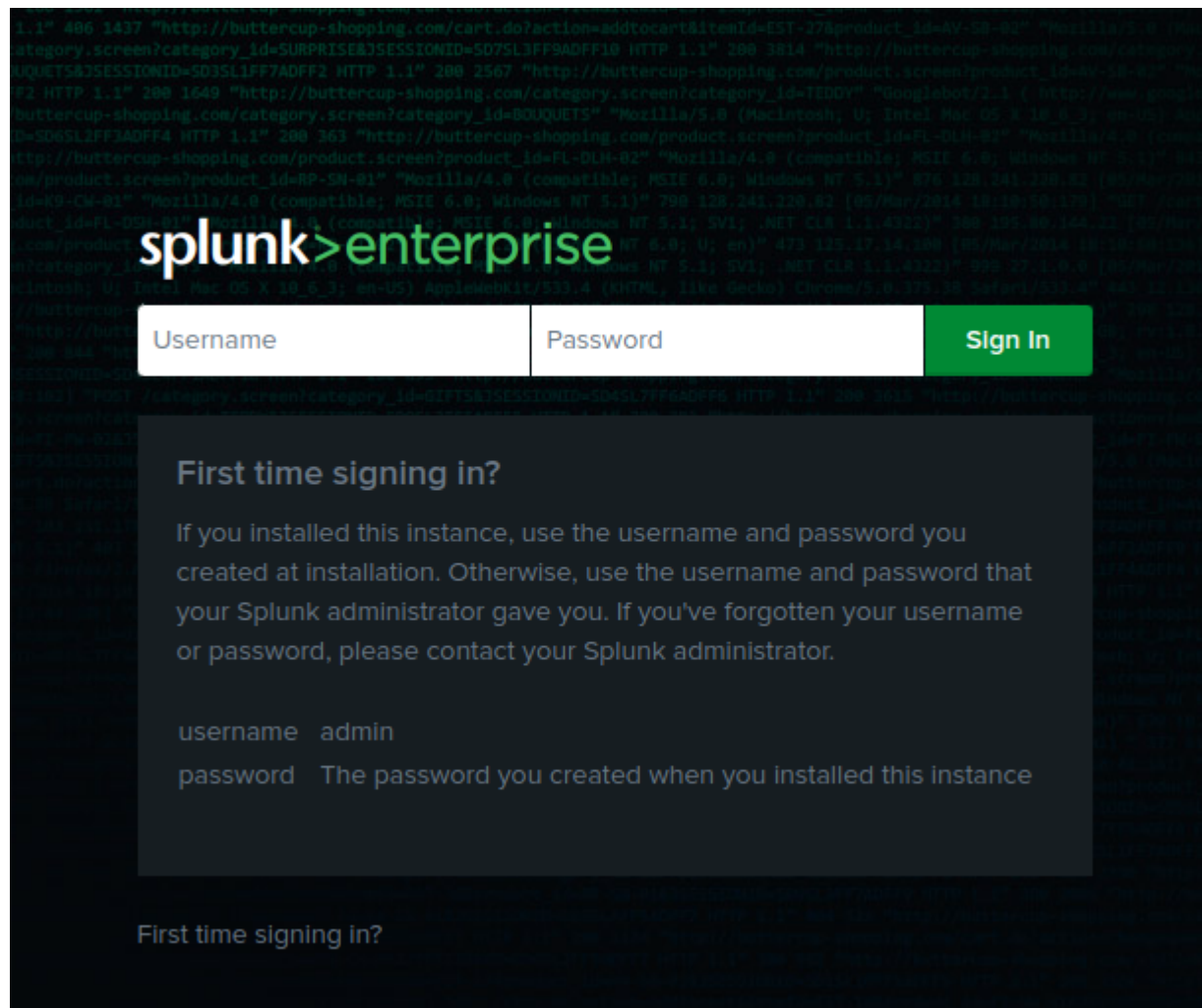
We can choose to start or to enable splunk every time the system starts. I will choose to start it manually every time the system starts.

After the installation, the files were placed into /opt/splunk.

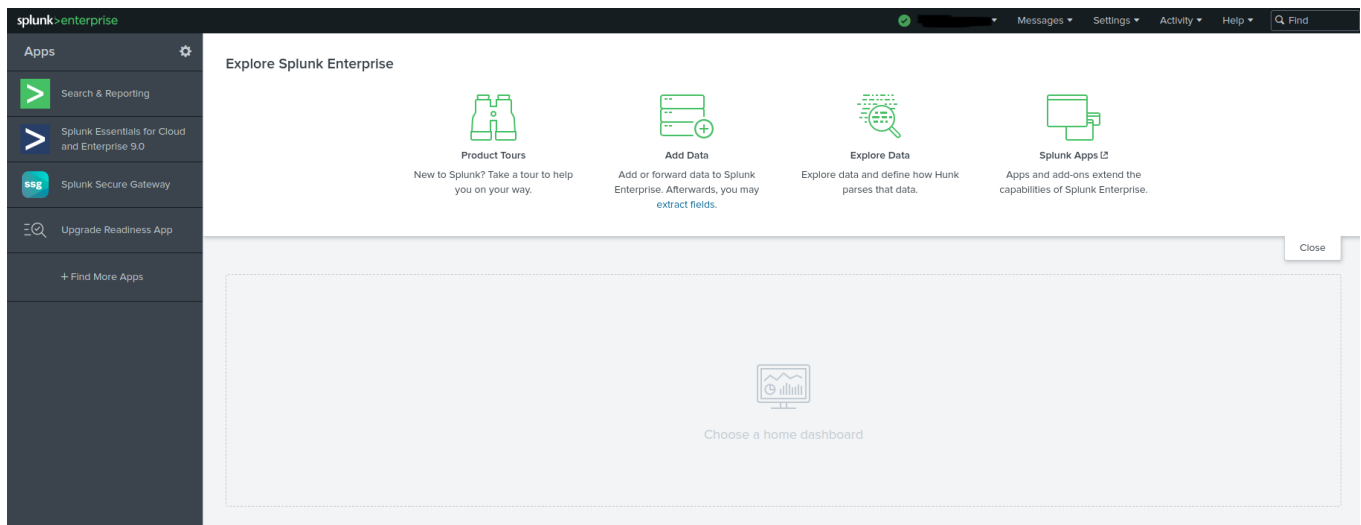
```
sudo /opt/splunk/bin/splunk start
```

We will get some output and the port in which splunk is listening. So we can access it through visiting the localhost address in that port from the web browser.

So visiting the `http://localhost:port` from our web browser, we can see the splunk panel.

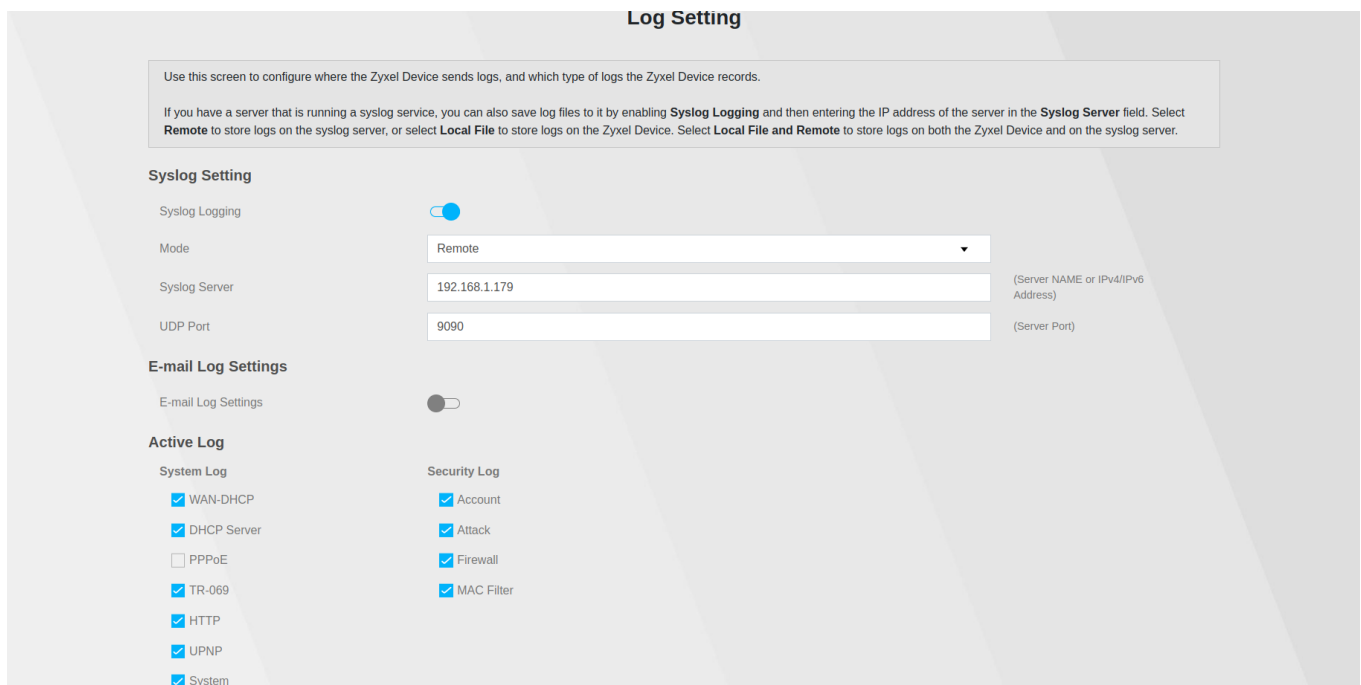


Log in with your credentials.



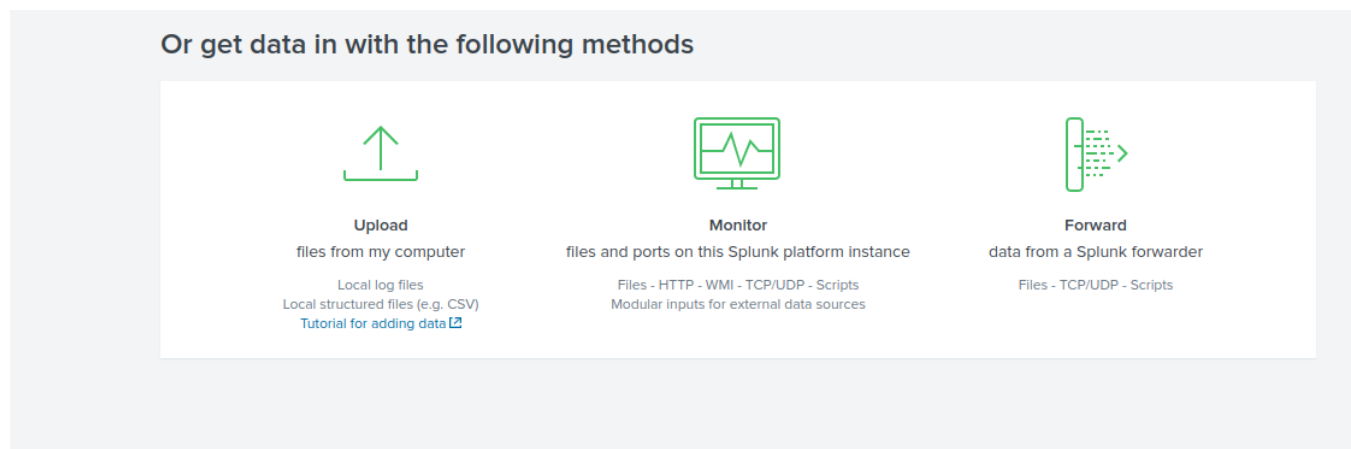
3. Forwarding the logs from my router to splunk

To forward the logs from the router to Splunk you need to enter into the admin panel of your router. It is important to note that some routers do not have that much logging functionalities, so maybe you are not able to forward the logs to Splunk.



In my case, I was using a Zyxel router that had some logging functionalities. I set the syslog server address and the UDP port to send the logging. This address is where my Splunk software is installed. Now we need to configure Splunk to listen on that 9090 UDP port!

So in Splunk, we go to the **Add Data** option.



Then **Monitor**. And we choose **TCP/UDP**. Set the UDP port.

Files & Directories
Upload a file, index a local file, or monitor an entire directory.

HTTP Event Collector
Configure tokens that clients can use to send data over HTTP or HTTPS.

TCP / UDP
Configure the Splunk platform to listen on a network port.

Scripts
Get data from any API, service, or database with a script.

Splunk Assist Instance Identifier
Assigns a random identifier to each Beam node

Systemd Journal Input for Splunk

Configure this instance to listen on any TCP or UDP port to capture data sent over the network (such as syslog). [Learn More](#)

☐ TCP ☒ UDP

Port ?
Example: 514

Source name override ?
host:port

Only accept connection from ?
example: 10.1.2.3, lbadhost.splunk.com, *.splunk.com

Then I set some options like the source type, Host, and App context.

Source type

The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you've got, so that the Splunk platform can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

Select New

log2metrics_keyvalue ▼

App context

Application contexts are folders within a Splunk platform instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. The Splunk platform loads all app contexts based on precedence rules. [Learn More](#)

App Context

Search & Reporting (search) ▼

Host

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

Method ?

IP DNS Custom

Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. [Learn More](#)

Index

Default ▼

[Create a new index](#)

FAQ

- > [How do indexes work?](#)
- > [How do I know when to create or use multiple indexes?](#)

After that we click review, we check everything is ok and then we are done! Let's see the logs!

4. Searching

We click the option **Search & Reporting** at the home page, and we can start to perform some searches on the logs that we have!

The first thing I did was to log in into the router once again to see how the events look like

New Search

host="192.168.1.1" login

✓ 2 events (22/01/2023 11:00:00.000 to 23/01/2023 11:28:17.000)

No Event Sampling ▼

Events (2)

Patterns

Statistics

Visualization

Format Timeline ▼

— Zoom Out

+ Zoom to Selection

× Deselect

List ▼

Format

50 Per Page ▼

< Hide Fields

≡ All Fields

SELECTED FIELDS

a host 1

a source 1

a sourcetype 1

INTERESTING FIELDS

date_hour 1

date_mday 1

date_minute 1

a date_month 1

date_second 1

a date_wday 1

date_year 1

a date_zone 1

a index 1

linecount 1

a punct 1

a splunk_server 1

timeendpos 1

timestartpos 1

+ Extract New Fields

i	Time	Event
>	23/01/2023 11:27:12.000	Jan 23 11:27:12 user.info zHttpd: Account: User [REDACTED] login from the host(192.168.1.179). host = 192.168.1.1 source = udp:9090 sourcetype = log2metrics_keyvalue
>	23/01/2023 11:27:12.000	Jan 23 11:27:12 user.info zHttpd: Account: User [REDACTED] login from the host(192.168.1.179). host = 192.168.1.1 source = udp:9090 sourcetype = log2metrics_keyvalue

Nice! So, we will receive many events form the router. Now we can make searches and start to gain some hands on experience with Splunk! The next step is going to be to add more logs from other sources to Splunk.

5. Forwarding the logs from my computer to splunk

In this case I will send the syslog from a macOS to Splunk. For this purpose we need to modify the **syslog.conf** file that lives in /etc. We need to add the following line: ***.* @ip:port**

The IP address is the IP address of the computer where you have splunk installed. And the port is the port that you will listen on Splunk to receive the logs.

```
File: syslog.conf

# Note that flat file logs are now configured in /etc/asl.conf

install.* @127.0.0.1:32376
.* @192.168.1.179:9191
```

Then we need to execute these commands to restart the syslogd daemon

```
$ sudo launchctl stop com.apple.syslogd
$ sudo launchctl start com.apple.syslogd
```

Finally, we need to configure Splunk to listen on the port that we specified on the **syslog.conf** to listen. Like we did when forwarding the router logs.

Now we can see the logs from our mac computer!

New Search

host*192.168.1.98*

123 events (1/23/23 9:00:00.000 AM to 1/24/23 9:07:20.000 AM) No Event Sampling

Events (123) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect

1 hour per column

List Format 50 Per Page

< Hide Fields All Fields

SELECTED FIELDS

- # host 1
- # source 1
- # sourcetype 1

INTERESTING FIELDS

- # date_hour 2
- # date_mday 1
- # date_minute 7
- # date_month 1
- # date_serverid 14

Time	Event
1/24/23 9:06:38.000 AM	Jan 24 09:06:38 192.168.1.98 Jan 24 09:06:38 macOS diagnostics_agent[775]: host = 192.168.1.98 source = udp:9191 sourcetype = log2metrics_keyvalue
1/24/23 9:06:38.000 AM	Jan 24 09:06:38 192.168.1.98 Jan 24 09:06:38 macOS diagnostics_agent[775]: host = 192.168.1.98 source = udp:9191 sourcetype = log2metrics_keyvalue
1/24/23 9:06:38.000 AM	Jan 24 09:06:38 192.168.1.98 Jan 24 09:06:38 macOS softwareupdated[349]: SUOSIPowerEventObserver: System will sleep host = 192.168.1.98 source = udp:9191 sourcetype = log2metrics_keyvalue
1/24/23 9:06:38.000 AM	Jan 24 09:06:38 192.168.1.98 Jan 24 09:06:38 macOS SoftwareUpdateNotificationManager[845]: SUOSIPowerEventObserver: System will sleep host = 192.168.1.98 source = udp:9191 sourcetype = log2metrics_keyvalue

6. Installing snort and adding logs to Splunk

Snort is an IPS/IDS tool. So with snort we can monitor and drop/reject traffic based on rules.

Snort is very easy to install and configure on ubuntu.

```
sudo apt install snort
```

We will be asked to enter with CIDR notation the address of our local network.

After that we can start to use snort!

The snort logs are stored on **/var/log/snort** and the rules are stored on **/etc/snort/rules**. Let's create a rule to detect ping scans.

```
sudo vi /etc/snort/rules/local.rules
```

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your local
# additions here.

alert icmp any any -> $HOME_NET any (msg:"Ping scan detected"; sid:1000001;)
```

So the rule sets the action to alert, when using the ICMP protocol from any source to the LAN, and will display the message "Ping scan detected". The sid is a rule number identifier. We are using the number 1.000.001 because the numbers below 1.000.001 are reserved for rules from different vendors. So if we install some more rules we do not have an sid conflict between the rules. We need to restart the snort service for the changes to be applied.

Now we can use a tool like nmap to perform a ping scan to the whole network or we can create an easy script for this purpose

```
#!/bin/bash

function ctrl_c {
    exit 1
}

trap ctrl_c INT

for host in {1..254}
do
    ping -c 1 -W 1 192.168.1.$host &>/dev/null && echo -e "Host up at --> 192.168.1.$host" &
done
wait
```

Let's run it and see how the logs look like!

```
01/25-13:11:02.267541  [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] {ICMP} 192.168.1.179 -> 192.168.1.1
01/25-13:11:02.293797  [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] {ICMP} 192.168.1.179 -> 192.168.1.98
01/25-13:11:02.602050  [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] {ICMP} 192.168.1.98 -> 192.168.1.179
01/25-13:11:02.646174  [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] {ICMP} 192.168.1.179 -> 192.168.1.251
01/25-13:11:02.696763  [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] {ICMP} 192.168.1.251 -> 192.168.1.179
```

Nice, now lets add this data to our Splunk SIEM!

In this case I will add the logs to splunk in a different way. I am going to monitor this file **snort.alert.fast** with splunk.

Configure this instance to monitor files and directories for data. To monitor all objects in a directory, select the directory. The Splunk platform monitors and assigns a single source type to all objects within the directory. This might cause problems if there are different object types or data sources in the directory. To assign multiple source types to objects in the same directory, configure individual data inputs for those objects. [Learn More](#)

File or Directory [?]

On Windows: c:\apache\apache.error.log or \\hostname\apache\apache.error.log. On Unix: /var/log or /mnt/www01/var/log.

Includelist [?]

Excludelist [?]

Now we have the snort logs on Splunk!

New Search

Save As Create Table View Close

source="/var/log/snort/snort.alert.fast" NOT "[IPv6-ICMP]" NOT Traffic | regex !~ IPv6|

Last 24 hours

2 events (1/24/23 12:00:00.000 PM to 1/25/23 12:56:43.000 PM) No Event Sampling

Job

Format Timeline Zoom Out Zoom to Selection Deselect

1 hour per column

	Time	Event
>	1/25/23 12:56:40.420 PM	01/25-12:56:40.420940 [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] (ICMP) 192.168.1.179 -> 192.168.1.1 01/25-12:56:40.448835 [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] (ICMP) 192.168.1.179 -> 192.168.1.98 01/25-12:56:40.519355 [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] (ICMP) 192.168.1.98 -> 192.168.1.179 host = 192.168.1.179 source = /var/log/snort/snort.alert.fast sourcetype = snort
>	1/25/23 12:43:50.373 PM	01/25-12:43:50.373584 [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] (ICMP) 192.168.1.179 -> 192.168.1.1 01/25-12:43:50.389063 [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] (ICMP) 192.168.1.179 -> 192.168.1.98 01/25-12:43:50.542843 [**] [1:1000001:0] Ping scan detected [**] [Priority: 0] (ICMP) 192.168.1.98 -> 192.168.1.179 host = 192.168.1.179 source = /var/log/snort/snort.alert.fast sourcetype = snort

< Hide Fields

All Fields

SELECTED FIELDS

host 1

source 1

sourcetype 1

INTERESTING FIELDS

date_hour 1

date_mday 1

date_minute 2