

# Real-Time Threat Detection Using Snort IDS Integrated with Splunk

## System Requirements for Snort IDS Lab Setup

### Operating System:

Ubuntu (installed on either a physical machine or a virtual machine).

### Internet Connectivity:

A stable internet connection to install necessary packages and updates.

### Virtualization (if using a VM):

A preferred hypervisor like VirtualBox, VMware, or KVM should be installed and configured.

### Web Server:

Apache HTTP Server must be installed and running on Ubuntu for testing Snort rules related to HTTP traffic.

### 1. Remote Access Server:

OpenSSH Server must be installed and running on Ubuntu for testing Snort rules related to SSH traffic.

## Step 1: Get the Latest List of Packages

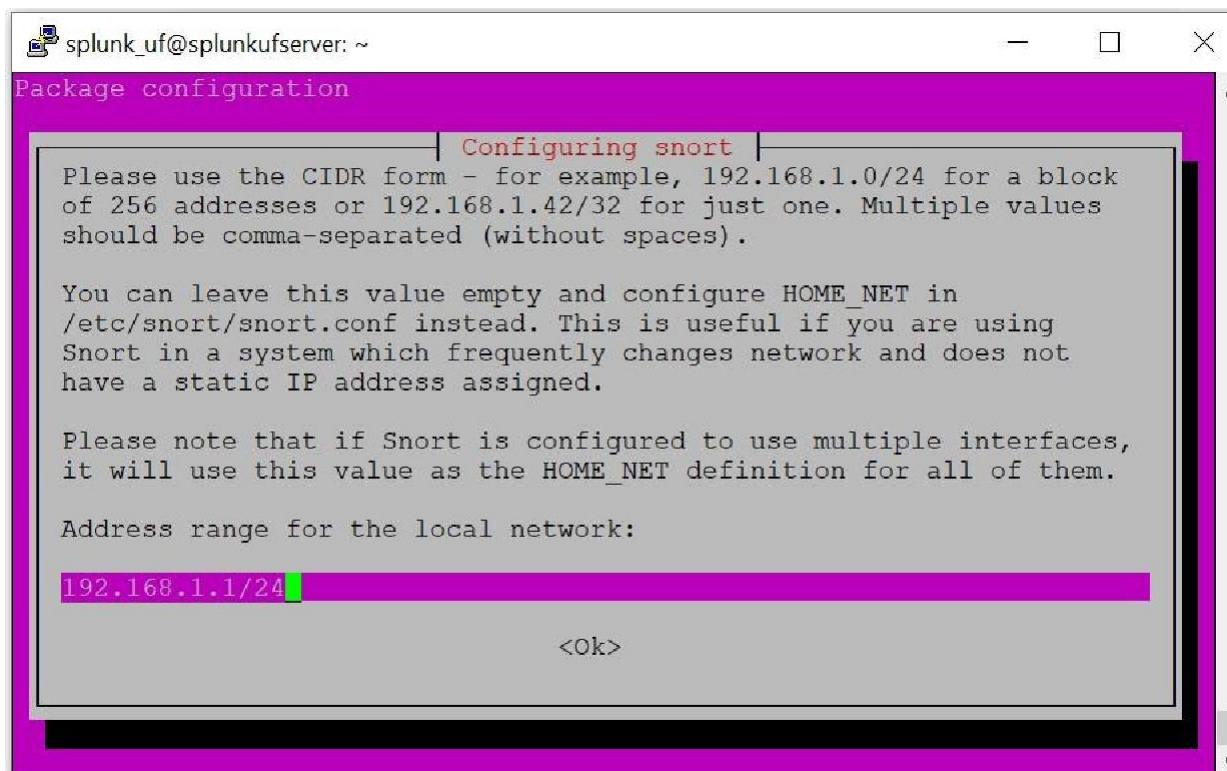
```
sudo apt-get  
update
```

## Step 2: Install Snort

Installing Snort on Ubuntu is effortless, simply use the following command:

```
sudo apt-get install snort -y
```

While installing Snort, we will be asked to determine which interface will be listened to by Snort.



192.168.1.1/24 then click ok.

## Step 3: Configuring Snort

Snort configuration file `snort.conf` will be seen under the `/etc/snort` directory. Snort will run according to this file.

### Configure Network Variables

We are using Snort as a Host Based IDS, so we should type the Ubuntu Machine's IP address as a **HOME\_NET** variable with an editor of your choice. We will be using VIM as an editor to configure `/etc/snort/snort.conf` file.

```
sudo nano /etc/snort/snort.conf
```



Ctrl + O, Enter Ctrl + x.

### Examining Rule Path and Rule Files

The **RULE\_PATH** variable in the `snort.conf` file determines the location of the snort rule files.

`/etc/snort/rules/local.rules` files contain our snort rules. When writing Snort rules, this file will be used.

Snort installation comes with some default community rules with classification. For example, if we want to examine backdoor rules we can examine the `backdoor.rules` file.

## Configure Output Files

To generate log files to examine alerts for the rule matching traffic pattern, we can use several methods. In this article we will write the logs to CSV files and PCAP files.

To generate logs in CSV files, the following line in the snort.conf file should be added to the “Configure output plugins” section.

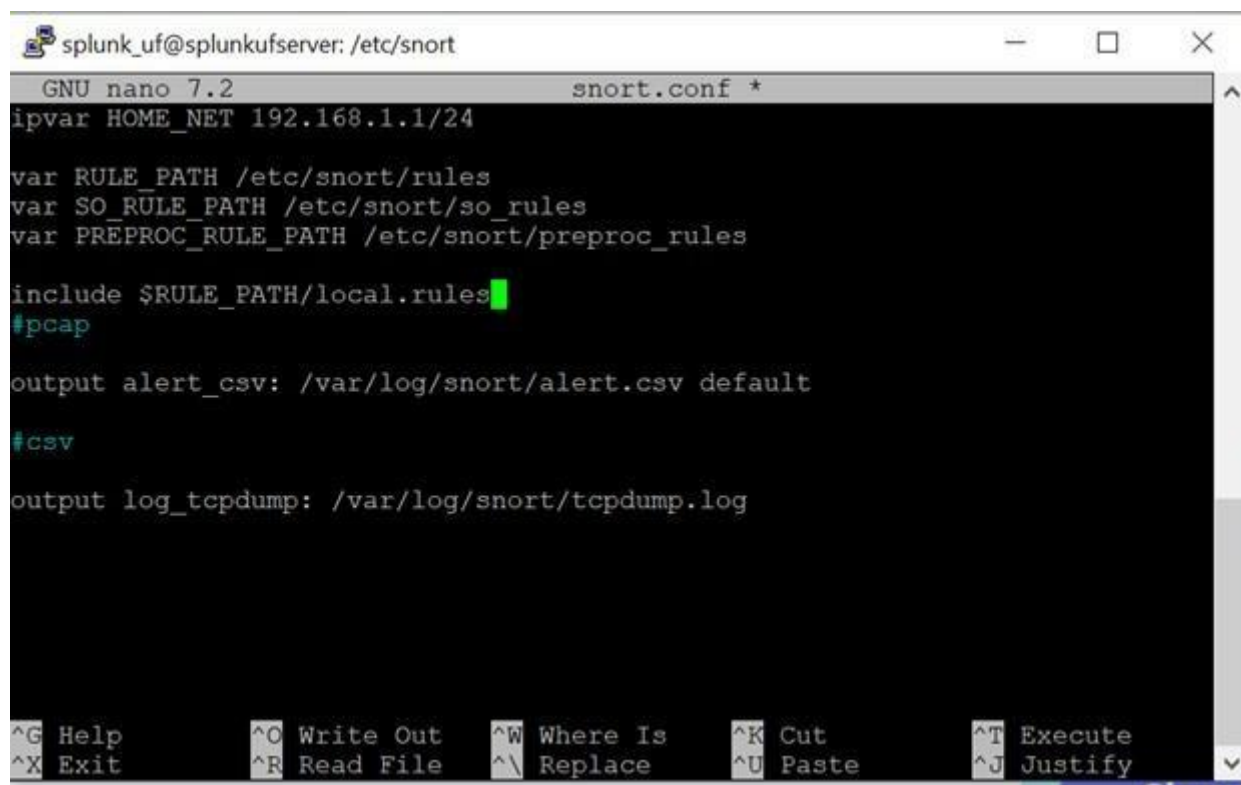
```
#pcap
```

```
output alert_csv: /var/log/snort/alert.csv default
```

To generate logs in PCAP files, the following line in the snort.conf file should be added to the “Configure output plugins” section.

```
#csv
```

```
output log_tcpdump: /var/log/snort/tcpdump.log
```

A screenshot of a terminal window titled 'splunk\_uf@splunkufserver: /etc/snort'. The window shows the 'snort.conf' file being edited with 'GNU nano 7.2'. The configuration includes: 'ipvar HOME\_NET 192.168.1.1/24', 'var RULE\_PATH /etc/snort/rules', 'var SO\_RULE\_PATH /etc/snort/so\_rules', 'var PREPROC\_RULE\_PATH /etc/snort/preproc\_rules', 'include \$RULE\_PATH/local.rules', '#pcap', 'output alert\_csv: /var/log/snort/alert.csv default', '#csv', and 'output log\_tcpdump: /var/log/snort/tcpdump.log'. The bottom status bar shows various keyboard shortcuts like '^G Help', '^O Write Out', '^W Where Is', '^K Cut', '^T Execute', '^X Exit', '^R Read File', '^\_ Replace', '^U Paste', and '^J Justify'.

```
splunk_uf@splunkufserver: /etc/snort
GNU nano 7.2      snort.conf *
ipvar HOME_NET 192.168.1.1/24

var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

include $RULE_PATH/local.rules
#pcap

output alert_csv: /var/log/snort/alert.csv default

#csv

output log_tcpdump: /var/log/snort/tcpdump.log

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify
```

## Test the Configuration File

We can test our configuration with the following command. The relevant command will produce an error if we made any mistakes in the configuration file.

```
sudo snort -T -i enp0s3 -c /etc/snort/snort.conf
```

To determine which interface Snort should listen on, type the interface on the Snort configuration area after typing the following command. # ip a or # ifconfig

If everything is configured correctly, you will see a “Snort successfully validated the configuration!” message at the bottom of the screen as shown below.

```
root@splunkufserver: /etc/snort
Verifying Preprocessor Configurations!

MaxRss at the end of rules:29440
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".

--== Initialization Complete ==--

o''_~  -*> Snort! <*-
o''_~  Version 2.9.20 GRE (Build 82)
o''_~  By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
o''_~  Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
o''_~  Copyright (C) 1998-2013 Sourcefire, Inc., et al.
o''_~  Using libpcap version 1.10.4 (with TPACKET_V3)
o''_~  Using PCRE version: 8.39 2016-06-14
o''_~  Using ZLIB version: 1.3

Total snort Fixed Memory Cost - MaxRss:29568
Snort successfully validated the configuration!
Snort exiting
root@splunkufserver: /etc/snort#
```

## Step 4: Understanding Snort Rule Structure

Snort rules consist of two parts which are Rule Header and Rule Options.

Rule header contains the **Rule's Action, Protocol, Source IP Address, Source Port, Direction, Destination IP Address and Destination Port** information.

**Rule options form the heart of Snort's intrusion detection engine combining ease of use with power and flexibility.** All Snort Rule Options are separated from each other using semicolon(;). Rule option keywords are separated from their arguments with a colon(:).

Some general rule options are: message, SID, REV.

Some general detection options are: content, distance, within, PCRE Here

is the snort rule structure:

```
action protocol sourceIP sourceport -> destinationIP destinationport ([Rule
options])
```

## ICMP Detection Rule

Open **/etc/snort/rules/local.rules** file to write custom ICMP rule with the editor of your choice and add the following rule to detect incoming ICMP packets.

```
[ alert icmp any any -> $HOME_NET any (msg:"ICMP Packet Monitor"; sid:100786; rev:1); ]
```

```
root@splunkufserver: /etc/snort/rules
GNU nano 7.2 local.rules
alert icmp any any -> $HOME NET any (msg:"ICMP Packet Monitor"; sid:100786; rev:1;)
```

Ctrl + O, Enter Ctrl + x.

## SSH Connection Attempts Detection Rule

As you know SSH protocol is running on TCP 22 by default. There is an SSH Server running on the Ubuntu machine. To detect incoming SSH connection attempts with the snort add the second rule to the /etc/snort/rules/local.rules file as following:

```
[ alert tcp any any -> $HOME_NET 22 (msg:"SSH Brute Force Attempt"; sid:107806; rev:1;) ]
```

```
root@splunkufserver: /etc/snort/rules
```

```
GNU nano 7.2 local.rules *
alert icmp any any -> $HOME_NET any (msg:"ICMP Packet Monitor"; sid:100786; rev:1;)
alert tcp any any -> $HOME_NET 22 (msg:"SSH Brute Force Attempt"; sid:107806; rev:1;)
```

Ctrl + O, Enter Ctrl + x.

## Step 5: Testing Snort Rules

We will test ICMP, SSH, and HTTP (curl) based Snort alerts using an attacker machine and your Ubuntu Snort machine.

## Before Testing — Run Snort in Alert Mode

On your **Ubuntu machine (Snort installed)**, run the following command to monitor and see alerts in real time:

```
# sudo snort -q -l /var/log/snort -i ens33 -A console -c /etc/snort/snort.conf
```

Replace `eth0` with your **actual network interface**. Use `ip a` to check it (e.g., `ens33`, `enp0s3`, etc.).

🔗 **Now Perform These 2 Attacks from Attacker Machine (e.g., Kali Linux)!**

1 - ICMP Ping Test (Triggers ICMP Rule)

On the **attacker machine**, run:

```
ping <target_ip>
```

2 - SSH Connection Attempt (Triggers SSH Rule)

On the **attacker machine**, run:

```
ssh <target-IP>
```

**ADD Forward server to monitor logs in splunk enterprise server**

**# sudo add forward-server 192.168.100.3:9997**

**Splunk username: admin**

**Password: admin@123**

```
splunk_uf@splunkufserver:/$ cd /opt/splunkforwarder/bin
splunk_uf@splunkufserver:/opt/splunkforwarder/bin$ sudo ./splunk add forward-server 192.168.100.3:9997
Warning: Attempting to revert the SPLUNK_HOME ownership
Warning: Executing "chown -R splunkfwd:splunkfwd /opt/splunkforwarder"
Your session is invalid. Please login.
Splunk username: admin
Password:
```

**Add file/directory to be monitored**

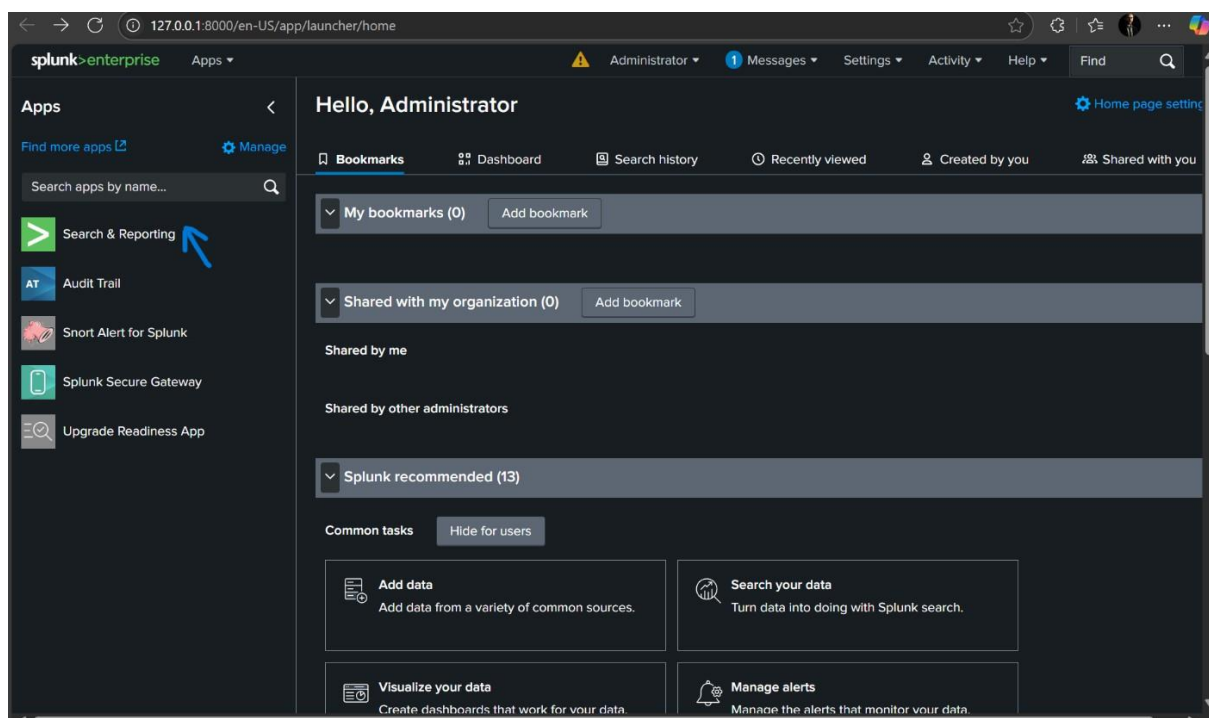
**# sudo ./splunk add monitor /var/log/snort -auth admin:admin@123**

```
splunk_uf@splunkufserver:/opt/splunkforwarder/bin$ sudo ./splunk add monitor /var/log/snort/alert -auth admin:admin@123
Warning: Attempting to revert the SPLUNK_HOME ownership
Warning: Executing "chown -R splunkfwd:splunkfwd /opt/splunkforwarder"
Added monitor of '/var/log/snort/alert'.
splunk_uf@splunkufserver:/opt/splunkforwarder/bin$
```

**Then restart splunk**

**# sudo ./splunk restart**

**Login to splunk enterprise server and follow below instruction to monitor Snort integrated logs.**





splunk>enterpriseApps

Administrator1 MessagesSettingsActivityHelpFind

SearchAnalyticsDatasetsReportsAlertsDashboards

Search & Reporting

Search

enter search here...Last 24 hours

No Event SamplingFast Mode

> Search History

How to Search

If you are not familiar with the search features, or want to learn more, or see your available data, see one of the following resources.

DocumentationTutorialData Summary

Analyze Your Data with Table Views

Table Views let you prepare data without using SPL. First, use a point-and-click interface to select data. Then, clean and transform it for analysis in Analytics Workspace, Search, or Pivot!

Learn more about Table Views, or view and manage your Table Views with the Datasets listing page.

Create Table View

Data Summary

Hosts (1)Sources (44)Sourcetypes (25)

filter

Host		Count	Last Update
splunkufw		167,235	6/18/25 4:52:55.000 PM

Data Summary

Hosts (1)Sources (44)Sourcetypes (25)

snort

Source		Count	Last Update
/var/log/snort/alert		726	6/13/25 7:39:23.000 PM
/var/log/snort/alert.csv		9,946	6/18/25 4:48:57.000 PM
/var/log/snort/backup-log.tar./alert.csv		13,364	6/13/25 7:20:43.000 PM
/var/log/snort/backup-log.tar./snort.alert.fast		38	6/13/25 7:20:43.000 PM
/var/log/snort/backup-log.tar./snort.alert.fast1.gz		15	6/13/25 7:20:43.000 PM
/var/log/snort/backup-log.tar./snort.alert.fast.2.gz		7,480	6/13/25 7:20:43.000 PM
/var/log/snort/backup-log.tar./snort.alert.fast.3.gz		5,688	6/13/25 7:20:43.000 PM
/var/log/snort/backup-log.tar./snort.alert.fast.4.gz		164	6/13/25 7:20:43.000 PM
/var/log/snort/snort.alert.fast		8,720	6/18/25 4:48:57.000 PM
/var/log/snort/snort.alert.fast1.gz		15	6/13/25 6:30:22.000 PM

Search

Analytics

Datasets

Reports

Alerts

Dashboards

Search & Reporting

New Search

Save As

Create Table View

Close

host=splunkufw

Last 24 hours

Q

6,645 events (6/18/25 1:30:00.000 PM to 6/19/25 1:57:04.000 PM)

No Event Sampling

Job

II

Fast Mode

Events (6,645)

Patterns

Statistics

Visualization

Timeline format

Zoom Out

+ Zoom to Selection

x Deselect

1 hour per column

Format

Show: 20 Per Page

View: List

Prev

1

2

3

4

5

6

7

8

...

Next

Hide Fields

All Fields

SELECTED FIELDS

a host 1

a source 11

a sourcetype 11

INTERESTING FIELDS

a index 1

# linecount 1

a splunk\_server 1

+ Extract New Fields

Time

Event

6/18/25 4:51:38.821 PM

2025-06-18T11:21:38.821166+00:00 splunkufw kernel: hrtimer: interrupt took 4677166 ns

6/18/25 4:51:38.821 PM

2025-06-18T11:21:38.821166+00:00 splunkufw kernel: hrtimer: interrupt took 4677166 ns

6/18/25 4:51:32.362 PM

2025-06-18T11:21:32.362964+00:00 splunkufw systemd[1]: apt-daily-upgrade.service: Consumed 5.243s CPU time.

6/18/25 4:51:32.362 PM

2025-06-18T11:21:32.362886+00:00 splunkufw systemd[1]: Finished apt-daily-upgrade.service - Daily apt upgrade and clean activities.

6/18/25 4:51:32.361 PM

2025-06-18T11:21:32.361403+00:00 splunkufw systemd[1]: apt-daily-upgrade.service: Deactivated successfully.

6/18/25

2025-06-18 11:21:31 997 INFO No packages found that can be upgraded unattended and no pending auto-removals

Summary:

- Installed Snort IDS on Ubuntu
- Configured directories, interfaces, and `snort.conf`
- Wrote and tested rules for ICMP ping, SSH login attempts, and HTTP requests
- Verified real-time alerts in console mode

This practical implementation strengthened my skills in network security and intrusion detection.



Project Objective

This project aims to simulate and detect real-time network threats using **Snort IDS** installed on Ubuntu, and send those alerts to **Splunk Enterprise** for centralized log analysis and real-time monitoring — just like a real SOC environment.



Created By:

Afroz Shaikh

afrozshaikh8086@gmail.com

Role: SOC Analyst in Training



Focus: Threat Detection | Splunk | IDS/IPS | Real-Time Monitoring