Dom Moore
Intrusion Detection and Prevention Systems

Overview

IDS/IPS administration and operation are essential duties in the modern security operations center (SOC)

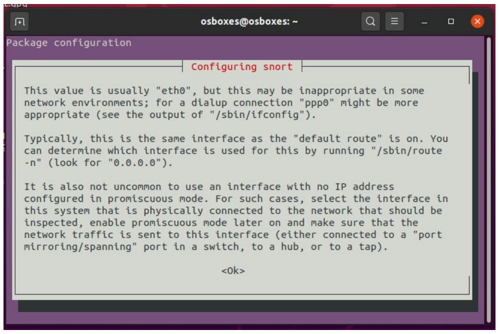
Problem area: Technical issues with the virtual machines being on the same network and providing alternate IP address

Part 1: Staging Snort

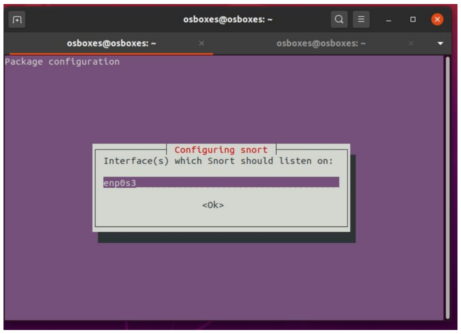
Deploy Snort to Ubuntu Linux 20.04, use Advanced Package Tool.

Step 1- Run Sudo apt update
Step 2- Run Sudo apt upgrade
Step 3- Run Sudo apt-get install snort
Step 4- Snort will prompt you for an adapter name and IP address. Provide these details to continue.
Step 5- Initialize Snort with Sudo snort -c snort.conf -A console -I [network interface name], note that you need to specify your network interface name accordingly.

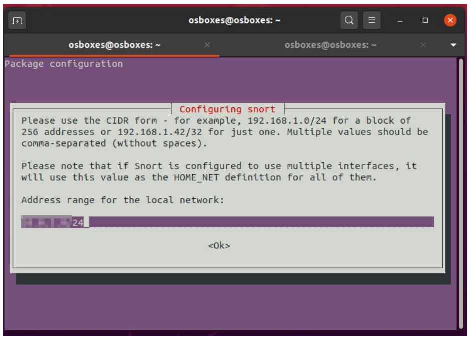
Snort Install



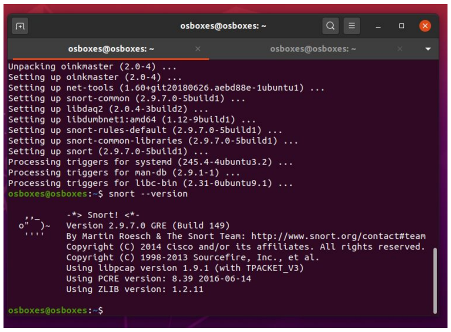
```
Package configuration

Configuring snort

This value is usually "eth0", but this may be inappropriate in some
network environments; for a dialup connection "ppp0" might be more
appropriate (see the output of "/sbin/ifconfig").

Typically, this is the same interface as the "default route" is on. You
can determine which interface is used for this by running "/sbin/route
-n" (look for "0.0.0.0").

It is also not uncommon to use an interface with no IP address
configured in promiscuous mode. For such cases, select the interface in
this system that is physically connected to the network that should be
inspected, enable promiscuous mode later on and make sure that the
network traffic is sent to this interface (either connected to a "port
mirroring/spanning" port in a switch, to a hub, or to a tap).

                              <Ok>
```



```
Package configuration

Configuring snort
Interface(s) which Snort should listen on:

enp0s3

<Ok>
```



```
Package configuration

Configuring snort
Please use the CIDR form - for example, 192.168.1.0/24 for a block of
256 addresses or 192.168.1.42/32 for just one. Multiple values should be
comma-separated (without spaces).

Please note that if Snort is configured to use multiple interfaces, it
will use this value as the HOME_NET definition for all of them.

Address range for the local network:

[redacted]24

                              <Ok>
```



```
Unpacking oinkmaster (2.0-4) ...
Setting up oinkmaster (2.0-4) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up snort-common (2.9.7.0-5build1) ...
Setting up libdaq2 (2.0.4-3build2) ...
Setting up libdumbnet1:amd64 (1.12-9build1) ...
Setting up snort-rules-default (2.9.7.0-5build1) ...
Setting up snort-common-libraries (2.9.7.0-5build1) ...
Setting up snort (2.9.7.0-5build1) ...
Processing triggers for systemd (245.4-4ubuntu3.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...
osboxes@osboxes:~$ snort --version

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

osboxes@osboxes:~$
```

Part 2: Detecting Network Activity with Custom Snort Rules

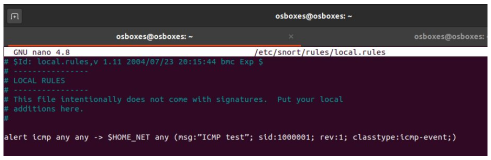
Step 1- Write and test a Snort rule that detects when ICMP packets are transmitted to its IP from the internet and raise an alert to the console.

Step 2- Write and test a Snort rule that detects when Kali Linux VM attempts an FTP connection to another local PC and raises an alert to the console.

Step 3- Write and test a Snort rule that detects when Kali Linux VM attempts an SSH connection to another local PC and raises an alert to the console.

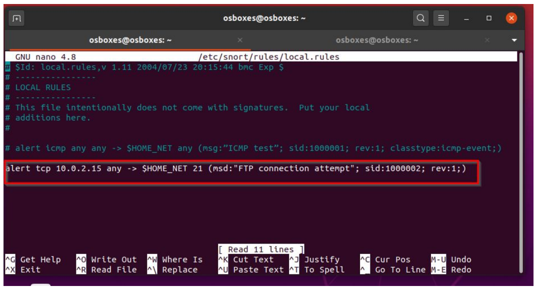
Step 4- Write and test a Snort rule that detects when Kali Linux VM attempts an HTTP connection to another local PC and raises an alert to the console.

Snort rules ICMP



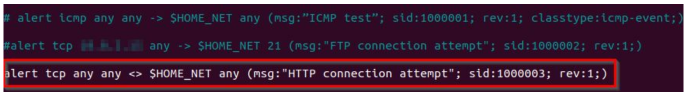
```
  GNU nano 4.8                                 /etc/snort/rules/local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
#

alert icmp any any -> $HOME_NET any (msg:”ICMP test”; sid:1000001; rev:1; classtype:icmp-event;)
```

FTP



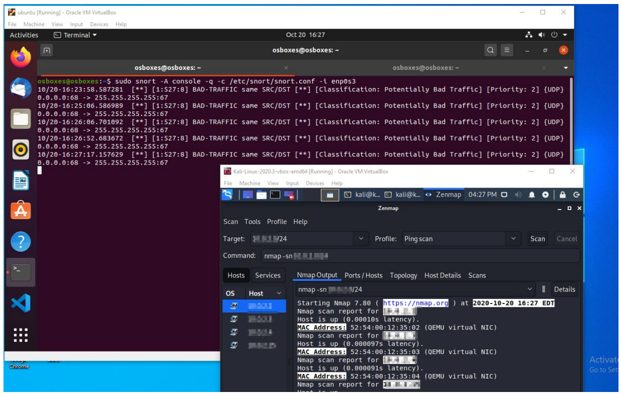
```
  GNU nano 4.8                                 /etc/snort/rules/local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
#

# alert icmp any any -> $HOME_NET any (msg:”ICMP test”; sid:1000001; rev:1; classtype:icmp-event;)

alert tcp 10.0.2.15 any -> $HOME_NET 21 (msd:"FTP connection attempt"; sid:1000002; rev:1;)



                                      [ Read 11 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo
```

HTTP



```
# alert icmp any any -> $HOME_NET any (msg:”ICMP test”; sid:1000001; rev:1; classtype:icmp-event;)

#alert tcp [redacted] any -> $HOME_NET 21 (msg:"FTP connection attempt"; sid:1000002; rev:1;)

alert tcp any any <> $HOME_NET any (msg:"HTTP connection attempt"; sid:1000003; rev:1;)
```

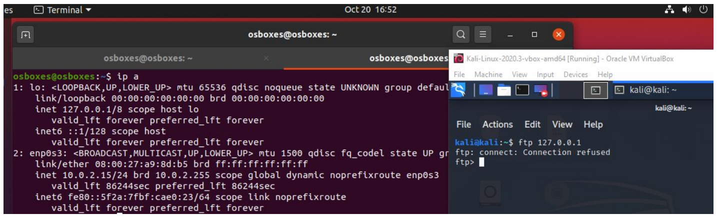
Alerts



```
osboxes@osboxes:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3
10/20-16:23:58.587281  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP}
0.0.0.0:68 -> 255.255.255.255:67
10/20-16:25:06.586989  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP}
0.0.0.0:68 -> 255.255.255.255:67
10/20-16:26:06.701092  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP}
0.0.0.0:68 -> 255.255.255.255:67
10/20-16:26:52.683672  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP}
0.0.0.0:68 -> 255.255.255.255:67
10/20-16:27:17.157629  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP}
0.0.0.0:68 -> 255.255.255.255:67
```

Target: [redacted]/24
Profile: Ping scan
Command: nmap -sn [redacted]

```
nmap -sn [redacted]/24

Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-20 16:27 EDT
Nmap scan report for [redacted]
Host is up (0.00010s latency).
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
Nmap scan report for [redacted]
Host is up (0.000097s latency).
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)
Nmap scan report for [redacted]
Host is up (0.000091s latency).
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)
Nmap scan report for [redacted]
```



```
osboxes@osboxes:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
    link/ether 08:00:27:a9:8d:b5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 86244sec preferred_lft 86244sec
    inet6 fe80::5f2a:7fbf:cae0:23/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
kali@kali:~$ ftp 127.0.0.1
ftp: connect: Connection refused
ftp>
```

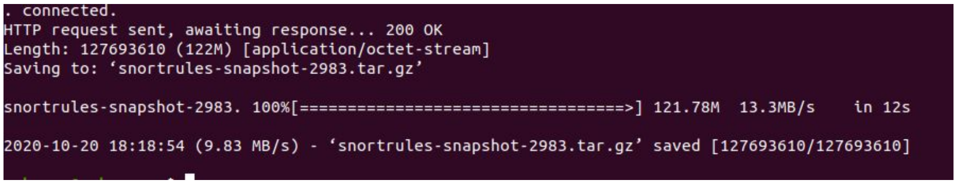
Terminal
Oct 20 16:52
osboxes@osboxes: ~
Kali-Linux-2020.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help

Part 3: Detecting Network Activity with Premade Snort Rules

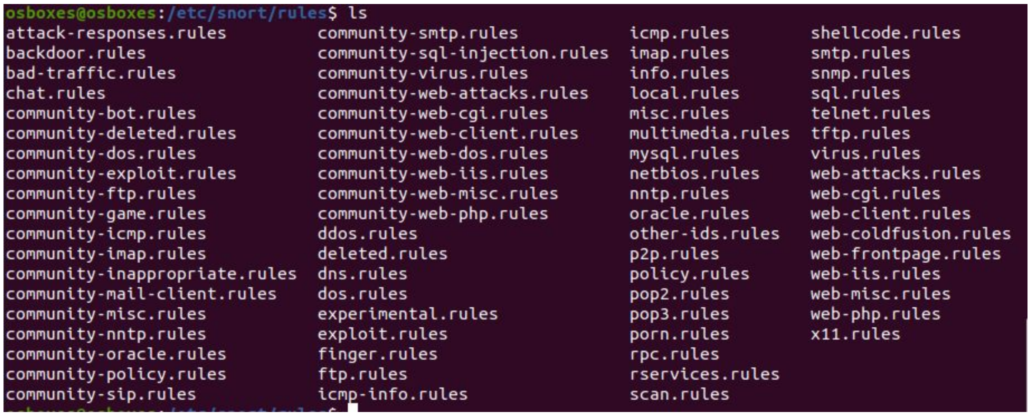
Step 1- Register at snort.org, then download the “Registered” rules package at Snort downloads.

Step 2- Load the rules pack into Snort.

Rules download / Install

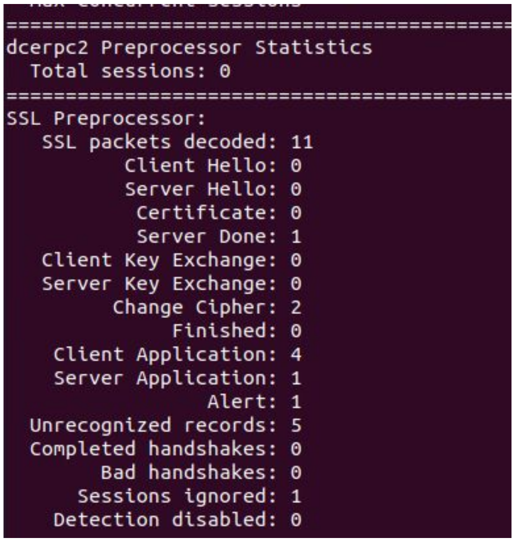


```
. connected.
HTTP request sent, awaiting response... 200 OK
Length: 127693610 (122M) [application/octet-stream]
Saving to: ‘snortrules-snapshot-2983.tar.gz’

snortrules-snapshot-2983. 100%[=================================>] 121.78M  13.3MB/s    in 12s

2020-10-20 18:18:54 (9.83 MB/s) - ‘snortrules-snapshot-2983.tar.gz’ saved [127693610/127693610]
```



```
osboxes@osboxes:/etc/snort/rules$ ls
attack-responses.rules           community-smtp.rules            icmp.rules         shellcode.rules
backdoor.rules                   community-sql-injection.rules   imap.rules         smtp.rules
bad-traffic.rules                community-virus.rules           info.rules         snmp.rules
chat.rules                       community-web-attacks.rules     local.rules        sql.rules
community-bot.rules              community-web-cgi.rules         misc.rules         telnet.rules
community-deleted.rules          community-web-client.rules      multimedia.rules   tftp.rules
community-dos.rules              community-web-dos.rules         mysql.rules        virus.rules
community-exploit.rules          community-web-iis.rules         netbios.rules      web-attacks.rules
community-ftp.rules              community-web-misc.rules        nntp.rules         web-cgi.rules
community-game.rules             community-web-php.rules         oracle.rules       web-client.rules
community-icmp.rules             ddos.rules                      other-ids.rules    web-coldfusion.rules
community-imap.rules             deleted.rules                   p2p.rules          web-frontpage.rules
community-inappropriate.rules    dns.rules                       policy.rules       web-iis.rules
community-mail-client.rules      dos.rules                       pop2.rules         web-misc.rules
community-misc.rules             experimental.rules              pop3.rules         web-php.rules
community-nntp.rules             exploit.rules                   porn.rules         x11.rules
community-oracle.rules           finger.rules                    rpc.rules
community-policy.rules           ftp.rules                       rservices.rules
community-sip.rules              icmp-info.rules                 scan.rules
```

Rule scan detection



```
==========================================
dcerpc2 Preprocessor Statistics
  Total sessions: 0
==========================================
SSL Preprocessor:
   SSL packets decoded: 11
          Client Hello: 0
          Server Hello: 0
           Certificate: 0
           Server Done: 1
   Client Key Exchange: 0
   Server Key Exchange: 0
         Change Cipher: 2
              Finished: 0
    Client Application: 4
    Server Application: 1
                 Alert: 1
  Unrecognized records: 5
  Completed handshakes: 0
        Bad handshakes: 0
      Sessions ignored: 1
    Detection disabled: 0
```

Part 4: Reporting

- How does Snort NIDS differ from a LAN firewall appliance?
  - The snort does not prevent any action from taking place only alerts you of detection, this particular tool also needs more technical savvy in order to customize settings due to the fact that there is no GUI to interact with.
- Why would security teams deploy a NIDS solution?
  - To be able to monitor traffic on the network in an attempt to prevent attacks, also a way of identifying patterns that can be used to think of more advanced defensive approaches, as well as the ability to customize the rules to fit a specific need.
- What are some limitations/shortcomings of a NIDS solution? In other words, what malicious activity would a NIDS not detect?
  - A major limitation is the absence of a GUI this tool has massive capabilities and therefore a number of things can go wrong and without a good amount of technical skills to get this tool to work as someone might want it to could have a steep learning curve.