Michael Simmons

Cybrary Virtual Labs

Create Custom Snort Rules

Objective

- Take the following indicators and create snort rules based on these:
    - nc.exe (Executable being used to create connections to suspected attacker)
    - 172.16.200.2 (IP address of Suspected C2 controller)
    - Port 53 UDP traffic
    - TCP traffic on port 21 (FTP service being used for exfiltration)

Scenario

You are a first responder and have received intelligence that suspected attacker have set up a C2 controller (Server) on your network and are communicating with it. You want to create snort rules that alert you to traffic to 172.16.200.2 and DNS lookups on Port 53 to monitor what DNS servers are being queried as it is suspected that DNS is being used as a covert channel to create outbound connections.

You have also noticed that the suspected attackers download nc.exe from ftp servers and deploy them on suspected systems. You want to see which servers in your enterprise are downloading this tool to be away of which systems have been possibly compromised.

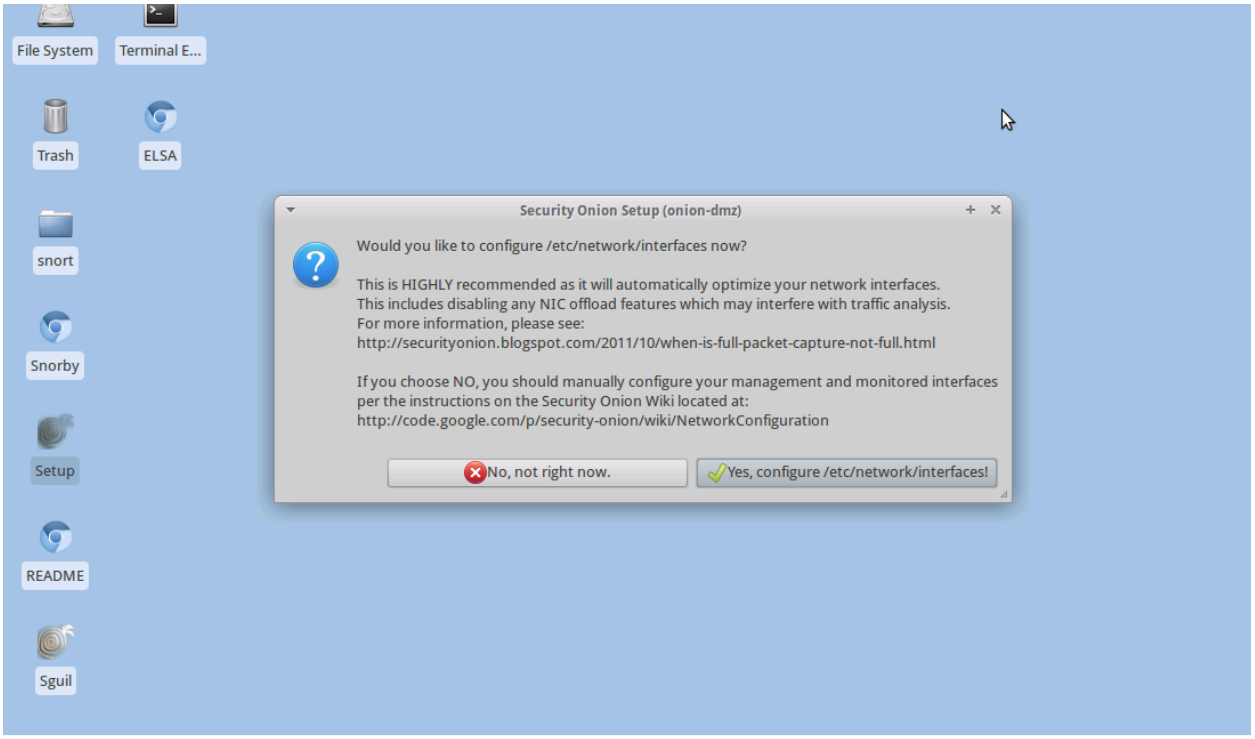
Configure Snort

Scenario

You may be called upon to confiure an IDS. In this lab we will cover a basic IDS configuration.

1. **Snort Installation**

   Locate the snort installation short cut on the desktop. It looks like an onion and says "Setup."
   Double click it.
   When prompted for a password type P@ssw0rd
   Click yes when asked to configure your /etc/network/interfaces.



Security Onion Setup (onion-dmz)

Would you like to configure /etc/network/interfaces now?

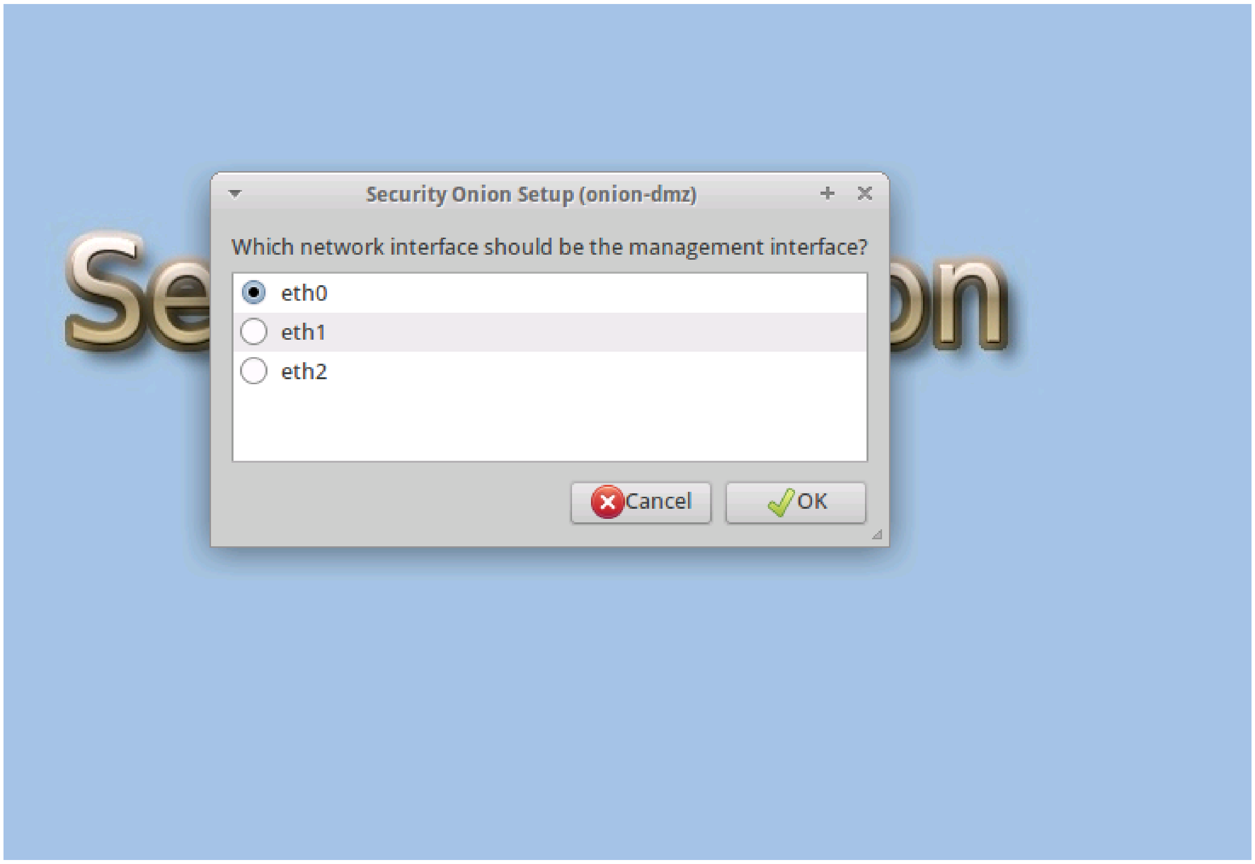
This is HIGHLY recommended as it will automatically optimize your network interfaces.
This includes disabling any NIC offload features which may interfere with traffic analysis.
For more information, please see:
http://securityonion.blogspot.com/2011/10/when-is-full-packet-capture-not-full.html

If you choose NO, you should manually configure your management and monitored interfaces per the instructions on the Security Onion Wiki located at:
http://code.google.com/p/security-onion/wiki/NetworkConfiguration

2. Configure Management Interface

Now we configure our management interface on eth0. Choose static and 192.168.0.77 for the IP and 255.255.255.0 for the subnet mask.

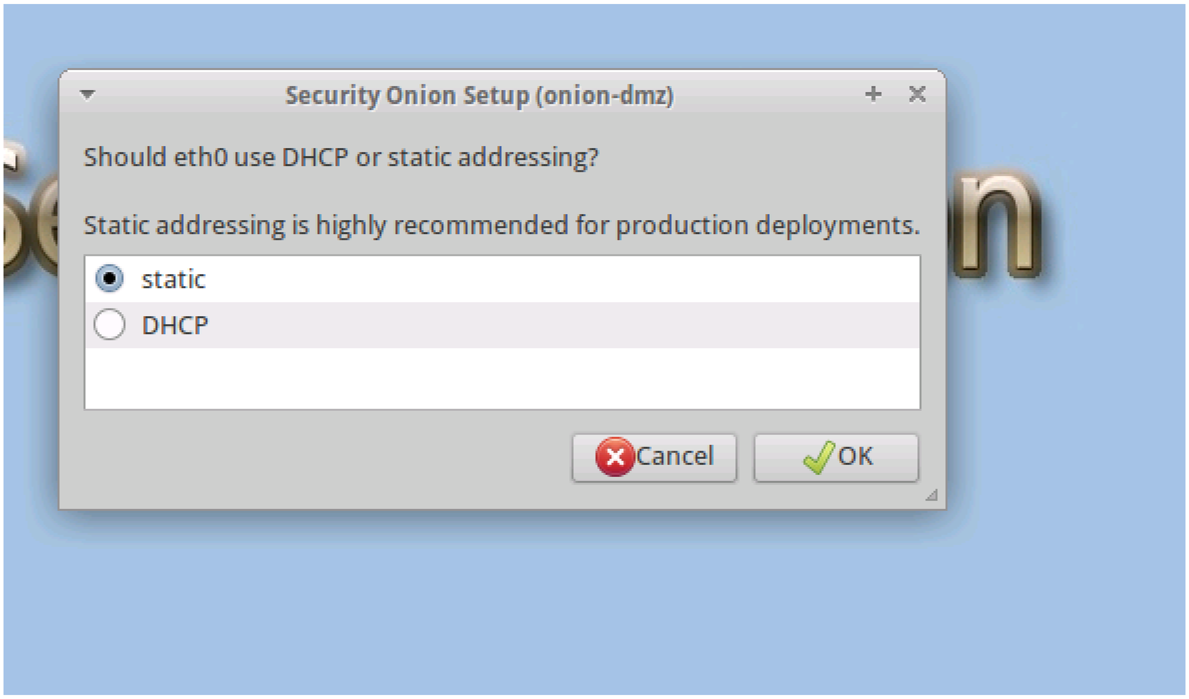
In a productin environment you would use the static IP assigned to the host you're security onion is on. Here we are choosing an IP that is not the gateway or the broadcast and that isn't already in use by other hosts on the network.



Security Onion Setup (onion-dmz)

Which network interface should be the management interface?

● eth0
○ eth1
○ eth2

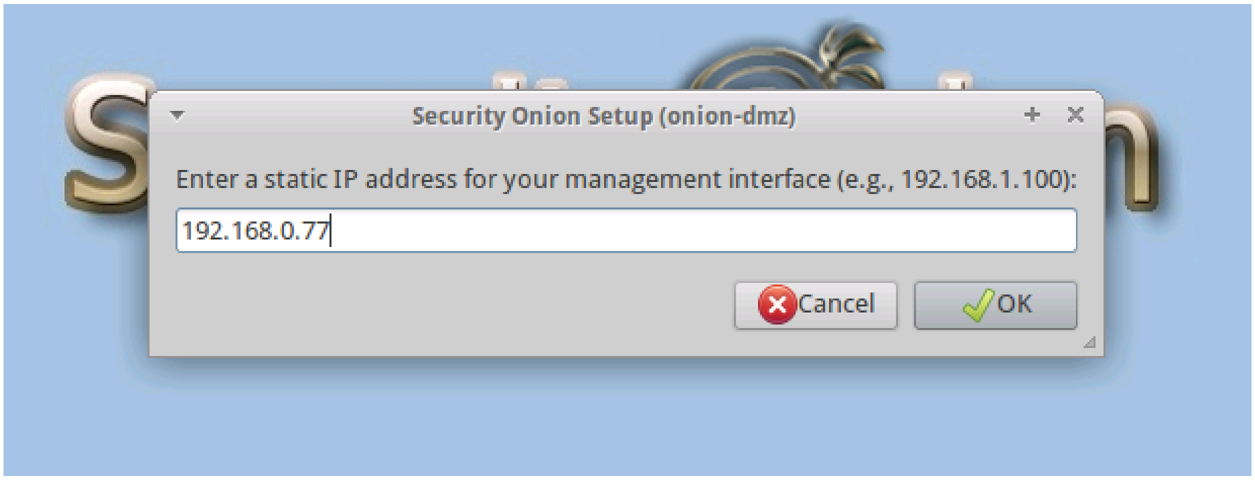


Security Onion Setup (onion-dmz)

Should eth0 use DHCP or static addressing?

Static addressing is highly recommended for production deployments.

● static
○ DHCP

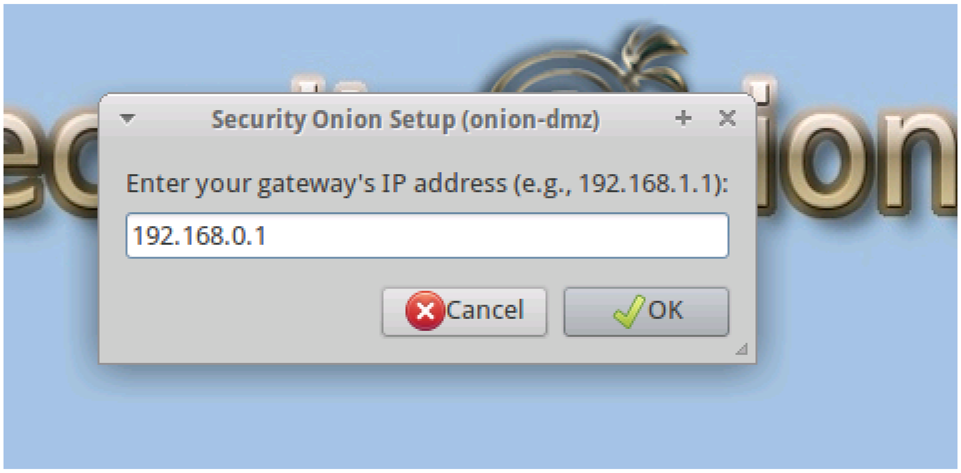

Security Onion Setup (onion-dmz)

Enter a static IP address for your management interface (e.g., 192.168.1.100):

192.168.0.77

3. **Gateway IP Address**

Enter 192.168.0.1 for the gateway IP.


Security Onion Setup (onion-dmz)

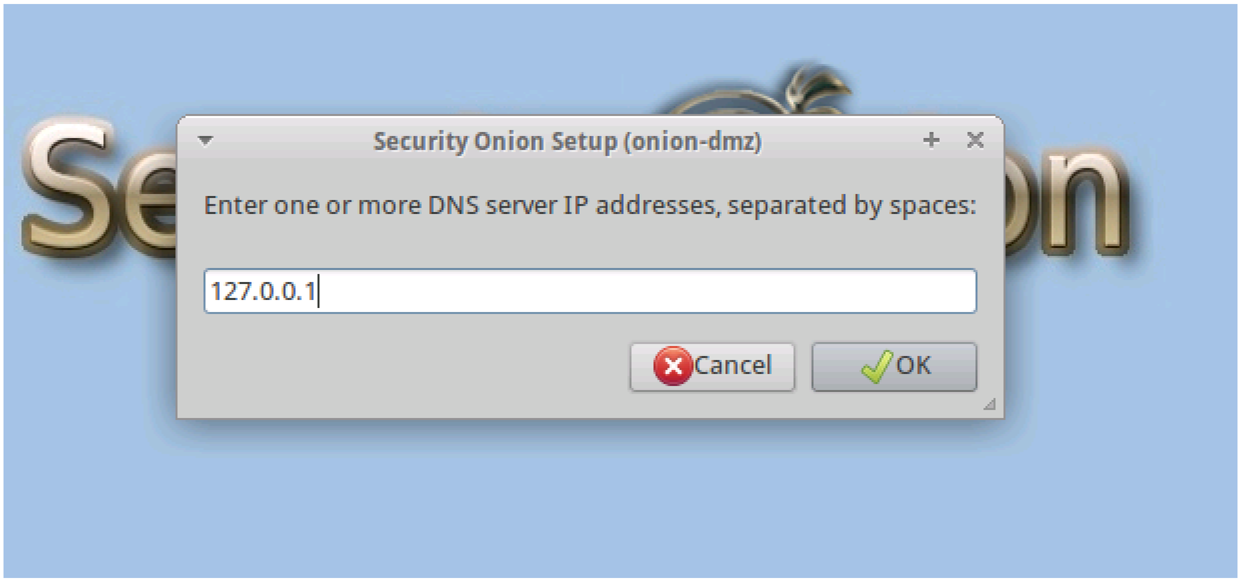
Enter your gateway's IP address (e.g., 192.168.1.1):

192.168.0.1

4.DNS Server IP

Enter 127.0.0.1 in the DNS server IP.

IN a production environment you'd use your own DNS server IP if you run one or could enter the IP of another known DNS server such as 8.8.8.8


Security Onion Setup (onion-dmz)

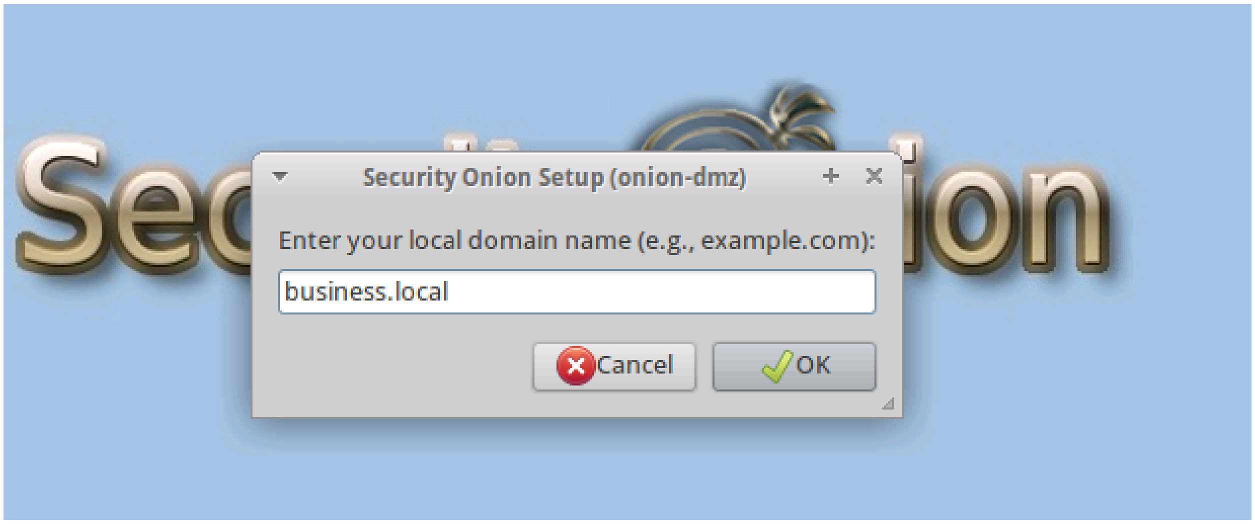
Enter one or more DNS server IP addresses, separated by spaces:

127.0.0.1

5. Local Domain

For the local domain put business.local

Choose No when asked to if you want to configure monitor interfaces. We only want to configure our management interface right now.


Security Onion Setup (onion-dmz)

Enter your local domain name (e.g., example.com):

business.local

6. Reboot

Review and accept the changes, then reboot.

View Existing Snort Rules

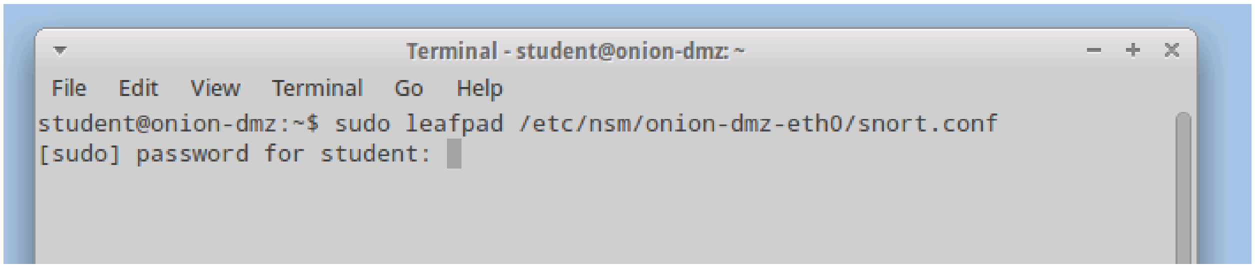
Scenario

In this exercise we will take a look at some of the existing rules that come pre-written with snort.

1. **Review Snort Config File**

   Let's view the snort .conf file located in the **/etc/nsm/onion-dmz-eth0** directory.

   We are reviewing the config file to learn the location of existing rulesets. Don't make any changes to the snort.conf file at this time.

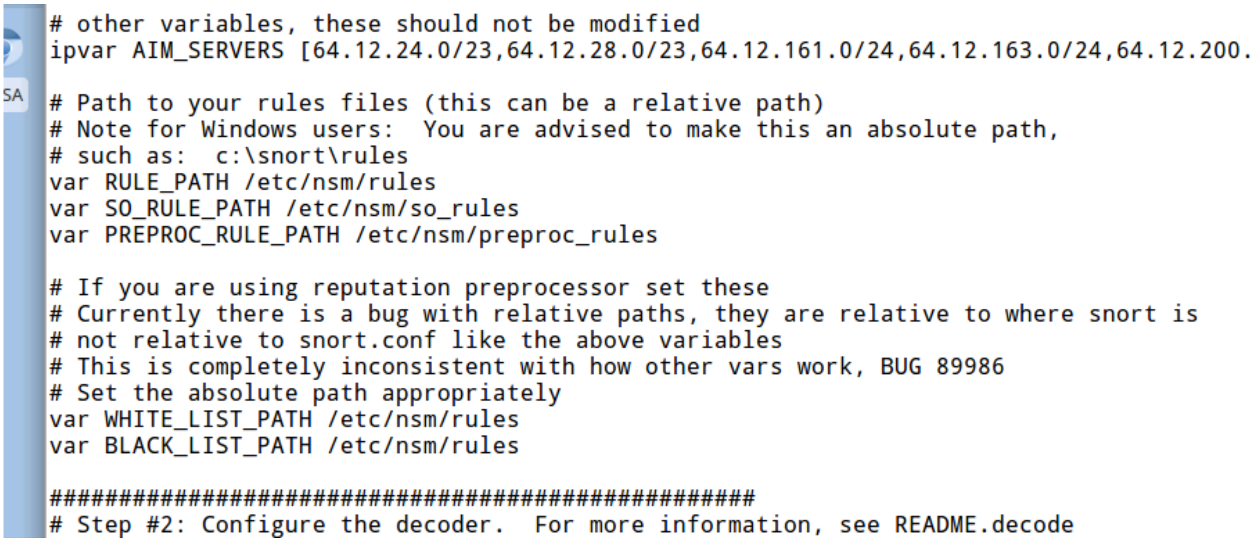


```
student@onion-dmz:~$ sudo leafpad /etc/nsm/onion-dmz-eth0/snort.conf
[sudo] password for student:
```

2. Review RULE_PATH Variable

Search for the RULE_PATH variable in the snort.conf file you opened in the previous step.

We see the var is set to /etc/nsm/rules.

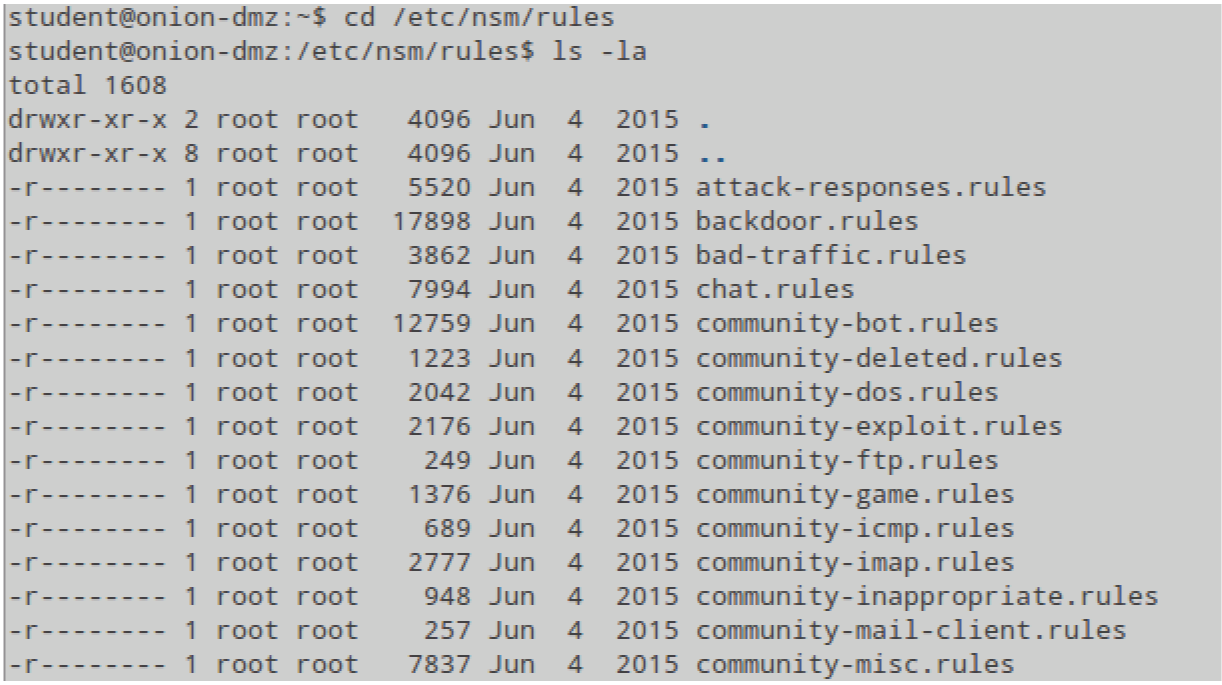


```
# other variables, these should not be modified
ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.

# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH /etc/nsm/rules
var SO_RULE_PATH /etc/nsm/so_rules
var PREPROC_RULE_PATH /etc/nsm/preproc_rules

# If you are using reputation preprocessor set these
# Currently there is a bug with relative paths, they are relative to where snort is
# not relative to snort.conf like the above variables
# This is completely inconsistent with how other vars work, BUG 89986
# Set the absolute path appropriately
var WHITE_LIST_PATH /etc/nsm/rules
var BLACK_LIST_PATH /etc/nsm/rules

###################################################
# Step #2: Configure the decoder.  For more information, see README.decode
```

3. Browse to Rules Directory

Now that we know where the rules are stored let's browse to that directory and take a look at some of the rules that came with snort.

Type:
cd /etc/nsm/rules
ls -la



```
student@onion-dmz:~$ cd /etc/nsm/rules
student@onion-dmz:/etc/nsm/rules$ ls -la
total 1608
drwxr-xr-x 2 root root   4096 Jun  4  2015 .
drwxr-xr-x 8 root root   4096 Jun  4  2015 ..
-r-------- 1 root root   5520 Jun  4  2015 attack-responses.rules
-r-------- 1 root root  17898 Jun  4  2015 backdoor.rules
-r-------- 1 root root   3862 Jun  4  2015 bad-traffic.rules
-r-------- 1 root root   7994 Jun  4  2015 chat.rules
-r-------- 1 root root  12759 Jun  4  2015 community-bot.rules
-r-------- 1 root root   1223 Jun  4  2015 community-deleted.rules
-r-------- 1 root root   2042 Jun  4  2015 community-dos.rules
-r-------- 1 root root   2176 Jun  4  2015 community-exploit.rules
-r-------- 1 root root    249 Jun  4  2015 community-ftp.rules
-r-------- 1 root root   1376 Jun  4  2015 community-game.rules
-r-------- 1 root root    689 Jun  4  2015 community-icmp.rules
-r-------- 1 root root   2777 Jun  4  2015 community-imap.rules
-r-------- 1 root root    948 Jun  4  2015 community-inappropriate.rules
-r-------- 1 root root    257 Jun  4  2015 community-mail-client.rules
-r-------- 1 root root   7837 Jun  4  2015 community-misc.rules
```

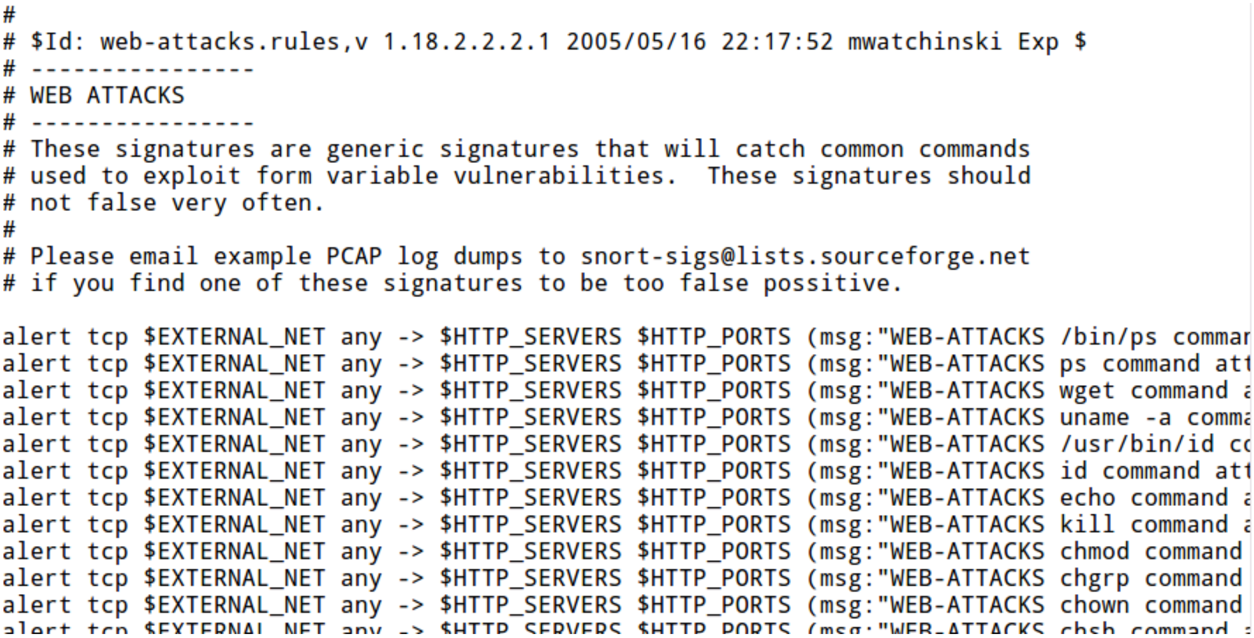
4.View Rules

View the rules file named using Leafpad:

web-attacks.rules

backdoor.rules

This portion of the lab is to expose you to additional snort rules that come preconfigured when you install snort. They can be very useful as a reference when creating your own snort rules.



```
#
# $Id: web-attacks.rules,v 1.18.2.2.2.1 2005/05/16 22:17:52 mwatchinski Exp $
# ----------------
# WEB ATTACKS
# ----------------
# These signatures are generic signatures that will catch common commands
# used to exploit form variable vulnerabilities.  These signatures should
# not false very often.
#
# Please email example PCAP log dumps to snort-sigs@lists.sourceforge.net
# if you find one of these signatures to be too false possitive.

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS /bin/ps comman
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS ps command att
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS wget command a
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS uname -a comma
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS /usr/bin/id co
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS id command att
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS echo command a
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS kill command a
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS chmod command
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS chgrp command
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS chown command
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS chsh command a
```

Create Custom Snort Rules

Scenario

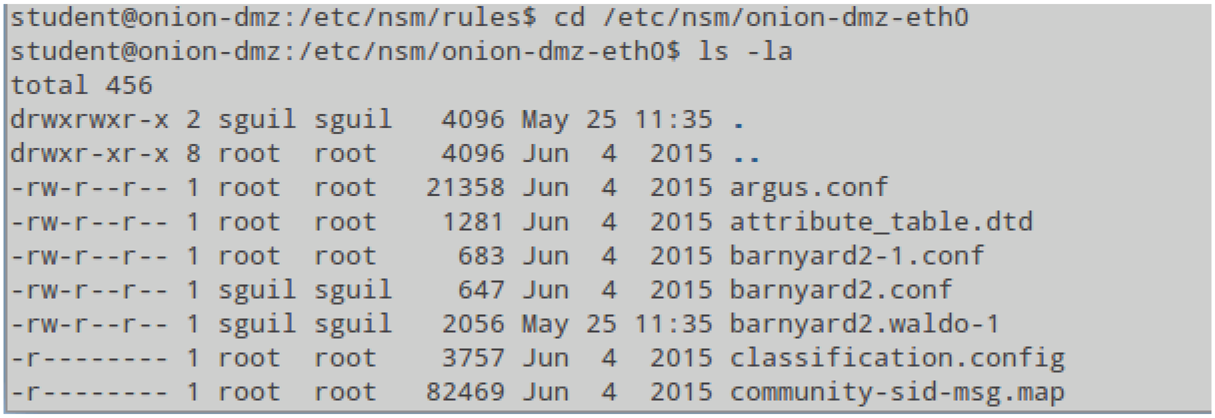
In this lab we will configure snort and create custom snort signatures based on indicators of known malicious activity.

1. **Snort Configuration Files**

   Let's begin by revisting the various files associated with snort:

   Type: cd /etc/nsm/onion-dmz-eth0
   Type: ls -la

   Keep the snort.conf file open because we will soon add our "custom rules" rules file to it. View the snort.conf file but, don't make any changes to it yet.



```
student@onion-dmz:/etc/nsm/rules$ cd /etc/nsm/onion-dmz-eth0
student@onion-dmz:/etc/nsm/onion-dmz-eth0$ ls -la
total 456
drwxrwxr-x 2 sguil sguil   4096 May 25 11:35 .
drwxr-xr-x 8 root  root    4096 Jun  4  2015 ..
-rw-r--r-- 1 root  root   21358 Jun  4  2015 argus.conf
-rw-r--r-- 1 root  root    1281 Jun  4  2015 attribute_table.dtd
-rw-r--r-- 1 root  root     683 Jun  4  2015 barnyard2-1.conf
-rw-r--r-- 1 sguil sguil    647 Jun  4  2015 barnyard2.conf
-rw-r--r-- 1 sguil sguil   2056 May 25 11:35 barnyard2.waldo-1
-r-------- 1 root  root    3757 Jun  4  2015 classification.config
-r-------- 1 root  root   82469 Jun  4  2015 community-sid-msg.map
```

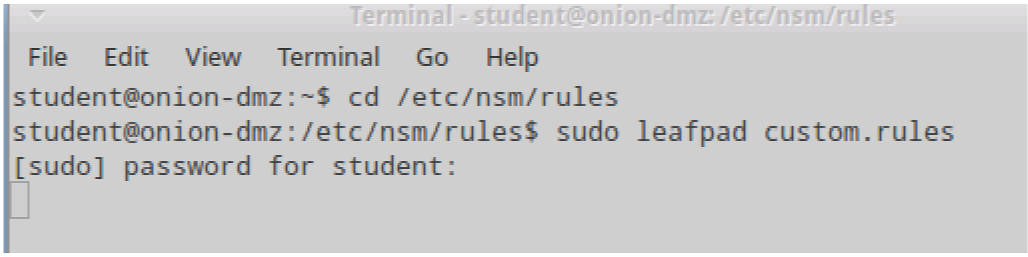
2.Create Custom Rules File

Open a terminal and type the following:

cd /etc/nsm/rules

sudo leafpad custom.rules

Here we use the application leafpad to create a file that will contain our custom rules.



```
student@onion-dmz:~$ cd /etc/nsm/rules
student@onion-dmz:/etc/nsm/rules$ sudo leafpad custom.rules
[sudo] password for student:
```

3.Add Custom Rules to "customrules" File

Let's add some rules to the "customrules" file we created in the previous step:

ADD THE INFORMATION BELOW TO THE "customrules" file

# BEGINNING OF CUSTOMRULES FILES

# These are the rules in response to todays incident

alert tcp any any -> any 21 (msg: "Establshed FTP connections "; flags:S; sid:10000;)
alert ip any any <> 172.16.200.2 any (msg:" KNOWN C2 Server "; sid:10001;)
alert tcp any any <> any any (msg: "Looking for nc.exe "; content:"nc.exe"; nocase; sid:10002;)
alert udp any any <> any any (msg: "DNS Requests "; sid:10004;)

# END OF CUSTOMRULES FILE

***Save the file before closing.

Make sure you save the file before closing it.

Be sure to check the "Idea" icon at the bottom of your screen for more information on the snort signature format.

Snort rules consist of the following;

alert tcp any any -> any 21 (msg: "Establshed FTP connections "; flags:S; sid:10000;)

alert - what we want this rule to do (options include log, activate,reject)

tcp - the protocol we are monitoring (options include tcp,udp,icmp,ip)

any - the source ip address(s) or network (CIDR allowed)

any - The source port(s) we wish to monitor

-> the direction we wish to monitor -> outbound <- inbound <> both directions

any - The desination IP address (CIDR allowed)

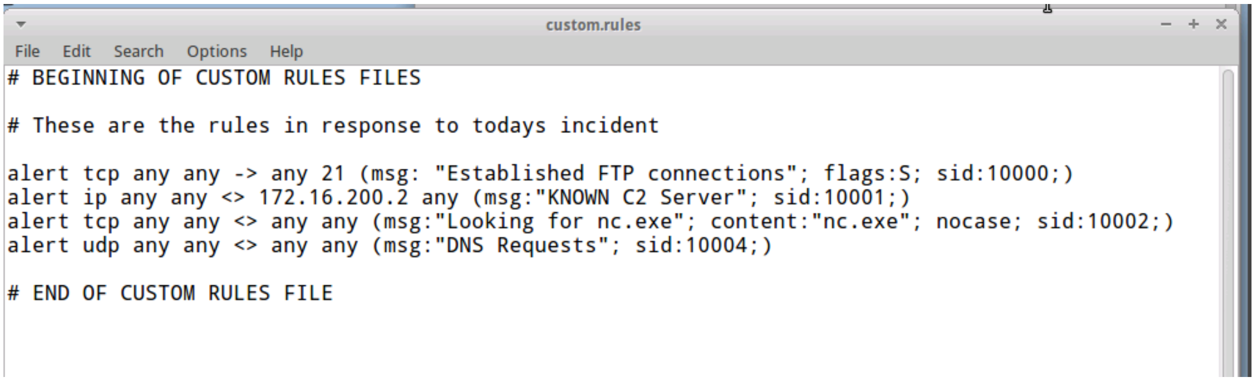
21 - The destination port we wish to monitor

msg: - What message we wish to be included in the message

flags:S - This is the sync flag in a tcp connection.

sid:10001 - This is the security ID . Newer versions of snort required a security identifier with each log and will complain with they are duplicates. These are user created and must contain only numbers.

...less



```
# BEGINNING OF CUSTOM RULES FILES

# These are the rules in response to todays incident

alert tcp any any -> any 21 (msg: "Established FTP connections"; flags:S; sid:10000;)
alert ip any any <> 172.16.200.2 any (msg:"KNOWN C2 Server"; sid:10001;)
alert tcp any any <> any any (msg:"Looking for nc.exe"; content:"nc.exe"; nocase; sid:10002;)
alert udp any any <> any any (msg:"DNS Requests"; sid:10004;)

# END OF CUSTOM RULES FILE
```

4. Add "customrules" to Snort Config File

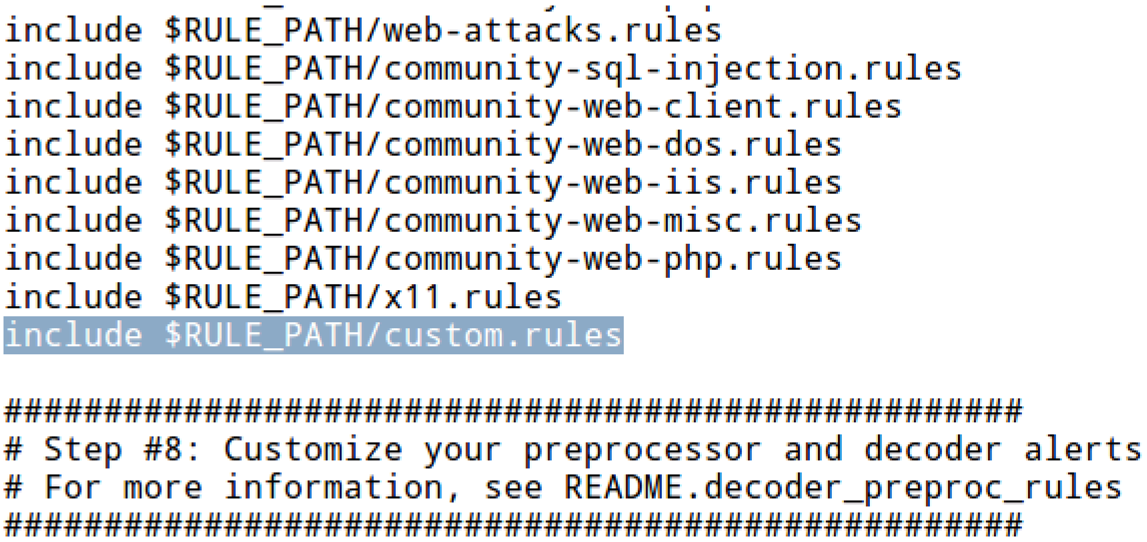
Open and edit your snort.conf file and make the following changes:

sudo leafpad /etc/nsm/onion-dmz-eth0/snort.conf &

Go to Step # 7 near the bottom of the file and add:

include $RULE_PATH/custom.rules

Then save file.



```
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/community-sql-injection.rules
include $RULE_PATH/community-web-client.rules
include $RULE_PATH/community-web-dos.rules
include $RULE_PATH/community-web-iis.rules
include $RULE_PATH/community-web-misc.rules
include $RULE_PATH/community-web-php.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/custom.rules

###################################################
# Step #8: Customize your preprocessor and decoder alerts
# For more information, see README.decoder_preproc_rules
###################################################
```

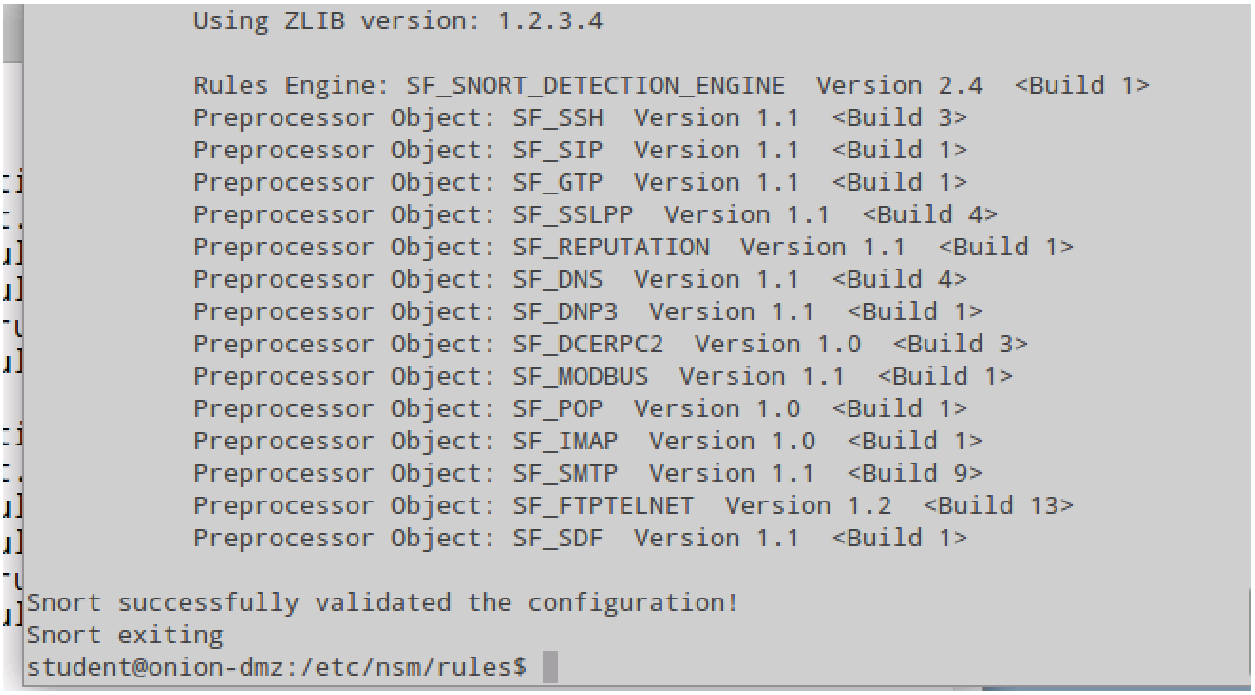
Test your Snort configuration after making changes by typing in:

snort -T -c /etc/nsm/onion-dmz-eth0/snort.conf

-T - means to test the configuration

-c - use the following configuration file

*Note: Use sudo



```
           Using ZLIB version: 1.2.3.4

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.4  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>

Snort successfully validated the configuration!
Snort exiting
student@onion-dmz:/etc/nsm/rules$
```

5.Testing Custom Rules

Test your the custom rules by running it against a packet capture that contains data your alerts SHOULD fire on. You should always test your rules AGAINST a sample data set to ensure desired results.

From a terminal emulator type the following:

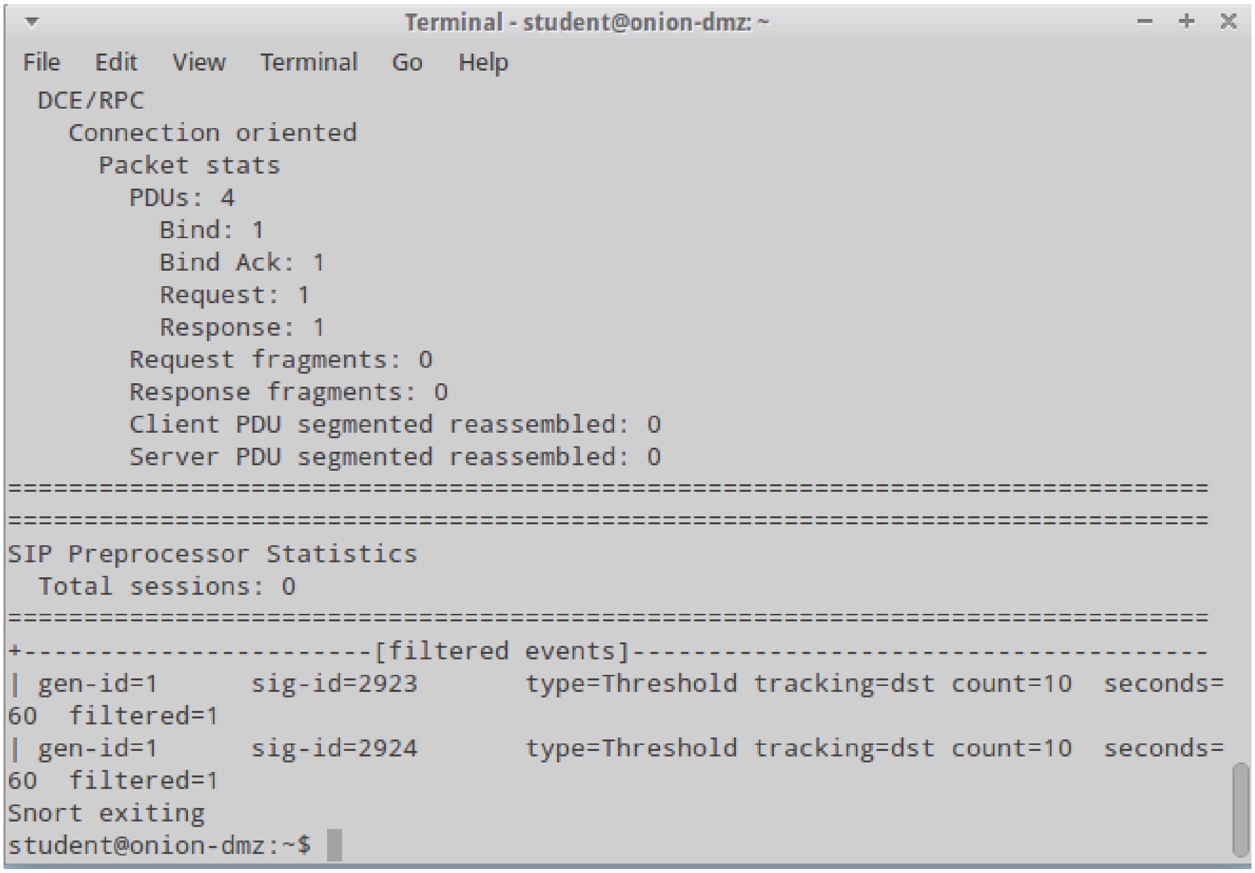
sudo snort -l . -c /etc/nsm/onion-dmz-eth0/snort.conf -r /home/student/capture131.pcap

It should have run without error.

Explanation -

sudo snort -l . -c /etc/nsm/onion-dmz-eth0/snort.con -r capture131.pcap

-l - Where to log the alerts to (here we use "." to signify current working directory)
-c - Which configuration file to use
-r - Which capture file to read into snort



```
  DCE/RPC
    Connection oriented
      Packet stats
        PDUs: 4
          Bind: 1
          Bind Ack: 1
          Request: 1
          Response: 1
        Request fragments: 0
        Response fragments: 0
        Client PDU segmented reassembled: 0
        Server PDU segmented reassembled: 0
===============================================================================
===============================================================================
SIP Preprocessor Statistics
  Total sessions: 0
===============================================================================
+-----------------------[filtered events]--------------------------------------
| gen-id=1      sig-id=2923       type=Threshold tracking=dst count=10  seconds=
60  filtered=1
| gen-id=1      sig-id=2924       type=Threshold tracking=dst count=10  seconds=
60  filtered=1
Snort exiting
student@onion-dmz:~$ 
```

6.Examine alerts created by custom rules

Examine the alerts created by your custom rules:

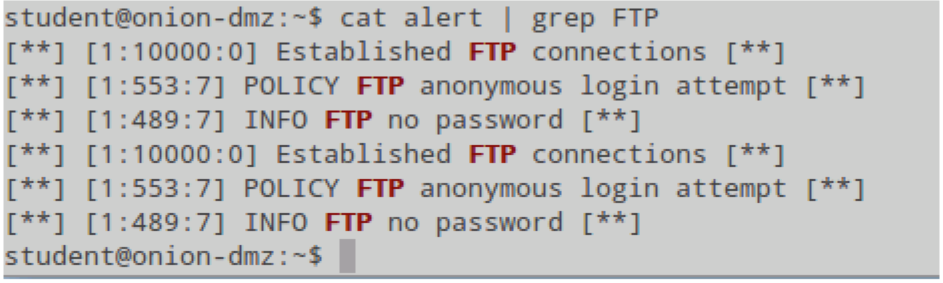
Type:
cat alert | grep FTP

View the full alert file by typing:

sudo leafpad alert

Can you figure out how to find other alerts based on the additional custom rules you wrote?

Do you see alerts created by your custom rules?



```
student@onion-dmz:~$ cat alert | grep FTP
[**] [1:10000:0] Established FTP connections [**]
[**] [1:553:7] POLICY FTP anonymous login attempt [**]
[**] [1:489:7] INFO FTP no password [**]
[**] [1:10000:0] Established FTP connections [**]
[**] [1:553:7] POLICY FTP anonymous login attempt [**]
[**] [1:489:7] INFO FTP no password [**]
student@onion-dmz:~$
```

Congratulations you have configured your IDS to use a custom rule set.