Lab9 SNORT

We will be using SNORT NIDS in this assignment. Its easy to use it or install it within Linux based systems. If you have one, you dont need to use Deterlab for this assignment. If you dont have one, reserve one machine in Deterlaband use it for this assignment (Reserve one or two machines)

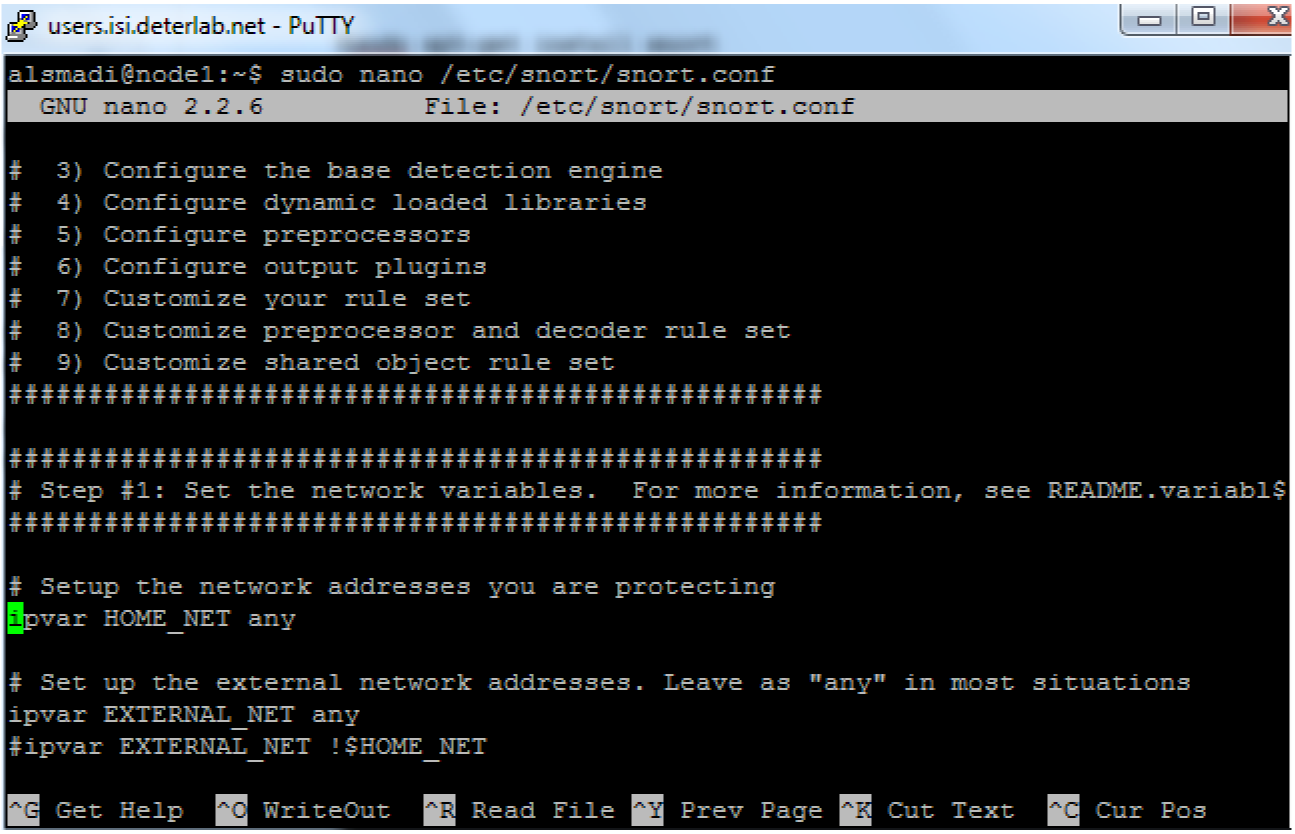
You can use the NS file below (to reserve 2 nodes)

```
------------------------------- SNORT.ns----------------------------------

set ns [new Simulator]

sourcetb_compat.tcl

set node1 [$ns node]

set node2 [$ns node]

tb-set-node-os $node1 Ubuntu1804-64-STD

tb-set-node-os $node2 Ubuntu1804-64-STD

#tb-set-node-memory-size $node1 512

#tb-set-node-memory-size $node2 512

#tb-set-node-startcmd node0 startupcmd

set link0 [$ns duplex-link $node1 $node2 100000.0kb 0.0ms DropTail]

$ns rtproto Static

$ns run

---------------------------------------------------------------------------------------------------------------
```

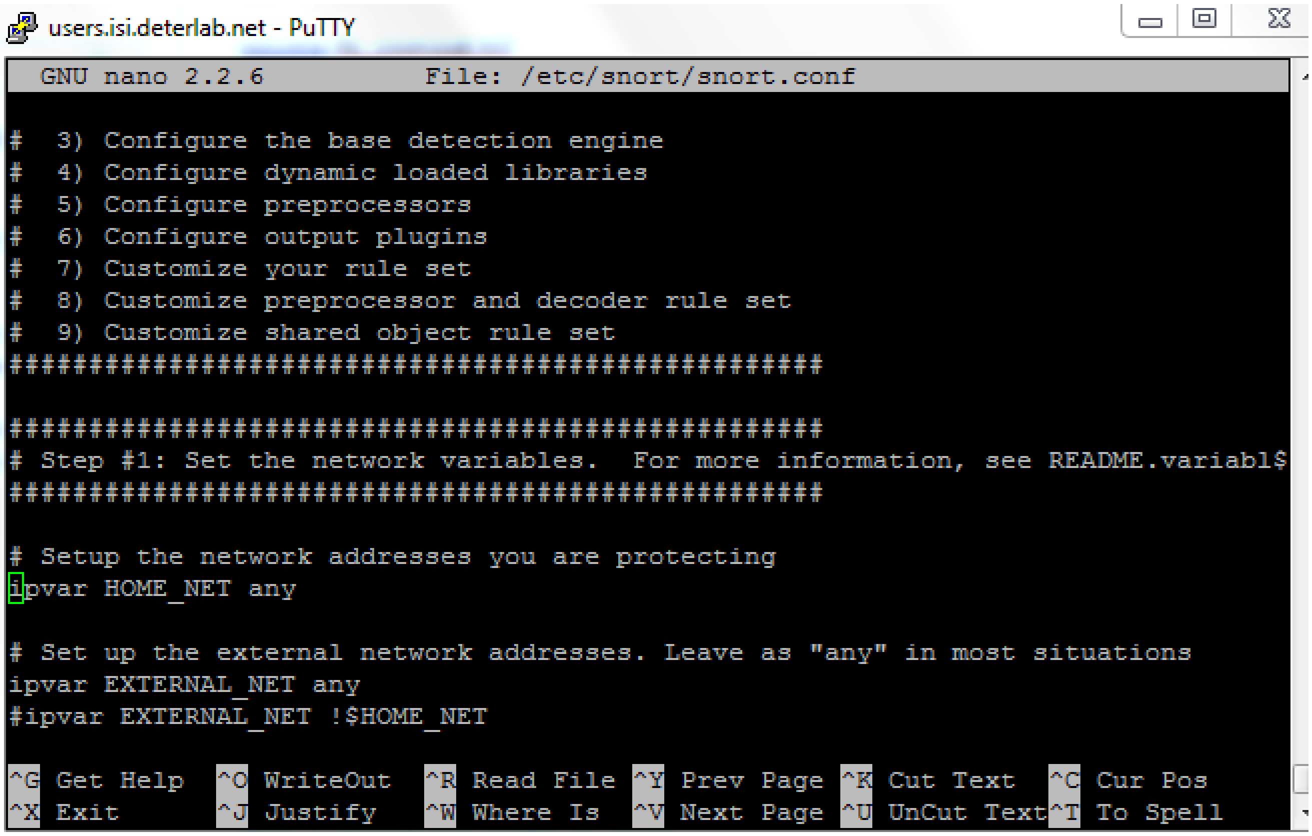
Step 1: Install and configure snort

```
$sudo apt-get install snort
```

Read the comments in **`/etc/snort/snort.conf`** carefully and pay attention to the definition of variable HOME_NET and EXTERNAL_NET.



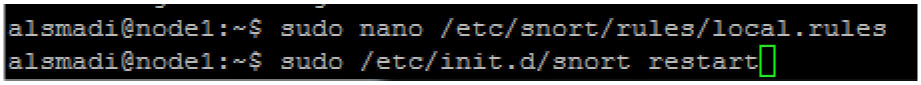
```
alsmadi@node1:~$ sudo nano /etc/snort/snort.conf
  GNU nano 2.2.6           File: /etc/snort/snort.conf

#  3) Configure the base detection engine
#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
###################################################

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

# Setup the network addresses you are protecting
ipvar HOME_NET any

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
#ipvar EXTERNAL_NET !$HOME_NET

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
```



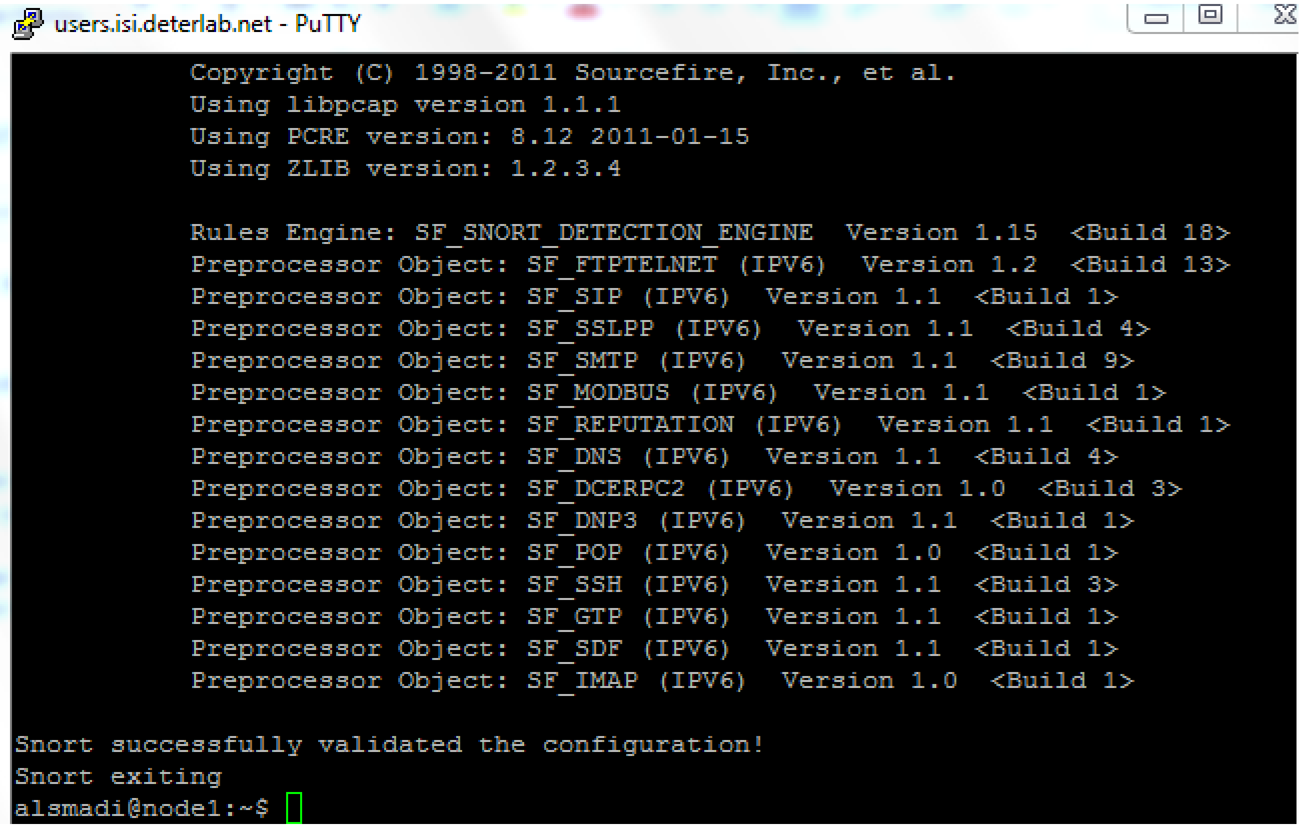
```
  GNU nano 2.2.6             File: /etc/snort/snort.conf

#   3) Configure the base detection engine
#   4) Configure dynamic loaded libraries
#   5) Configure preprocessors
#   6) Configure output plugins
#   7) Customize your rule set
#   8) Customize preprocessor and decoder rule set
#   9) Customize shared object rule set
###################################################

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

# Setup the network addresses you are protecting
ipvar HOME_NET any

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
#ipvar EXTERNAL_NET !$HOME_NET
```

You can test configuration by:

```
sudosnort -T -c /etc/snort/snort.conf
```

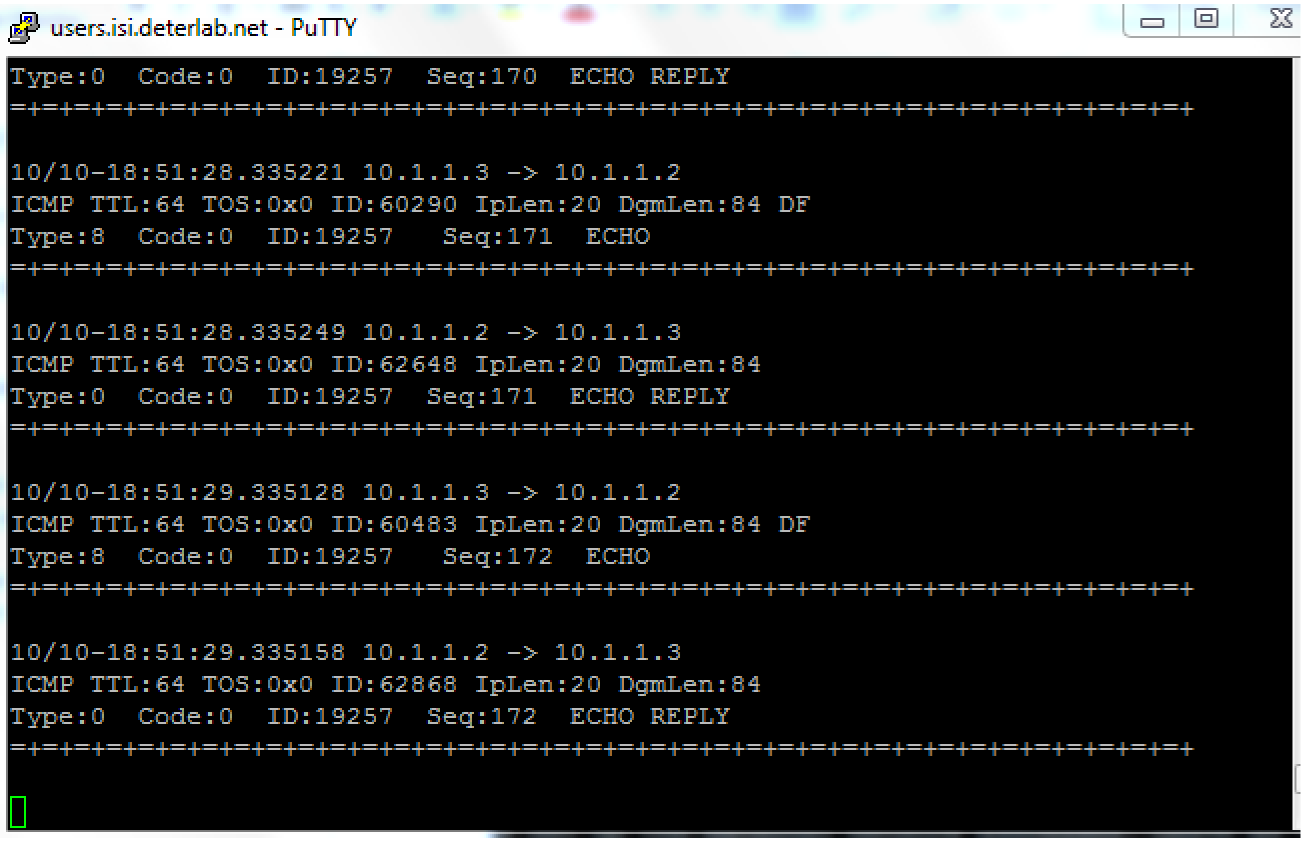


```
alsmadi@node1:~$ sudo nano /etc/snort/rules/local.rules
alsmadi@node1:~$ sudo /etc/init.d/snort restart
```



```
           Copyright (C) 1998-2011 Sourcefire, Inc., et al.
           Using libpcap version 1.1.1
           Using PCRE version: 8.12 2011-01-15
           Using ZLIB version: 1.2.3.4

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 1.15  <Build 18>
           Preprocessor Object: SF_FTPTELNET (IPV6)  Version 1.2  <Build 13>
           Preprocessor Object: SF_SIP (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP (IPV6)  Version 1.1  <Build 4>
           Preprocessor Object: SF_SMTP (IPV6)  Version 1.1  <Build 9>
           Preprocessor Object: SF_MODBUS (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS (IPV6)  Version 1.1  <Build 4>
           Preprocessor Object: SF_DCERPC2 (IPV6)  Version 1.0  <Build 3>
           Preprocessor Object: SF_DNP3 (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP (IPV6)  Version 1.0  <Build 1>
           Preprocessor Object: SF_SSH (IPV6)  Version 1.1  <Build 3>
           Preprocessor Object: SF_GTP (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP (IPV6)  Version 1.0  <Build 1>

Snort successfully validated the configuration!
Snort exiting
alsmadi@node1:~$
```

Then try running Snort as root: (basic command with no options sudo snort, monitoring mode). Image below after trying ping to the node.



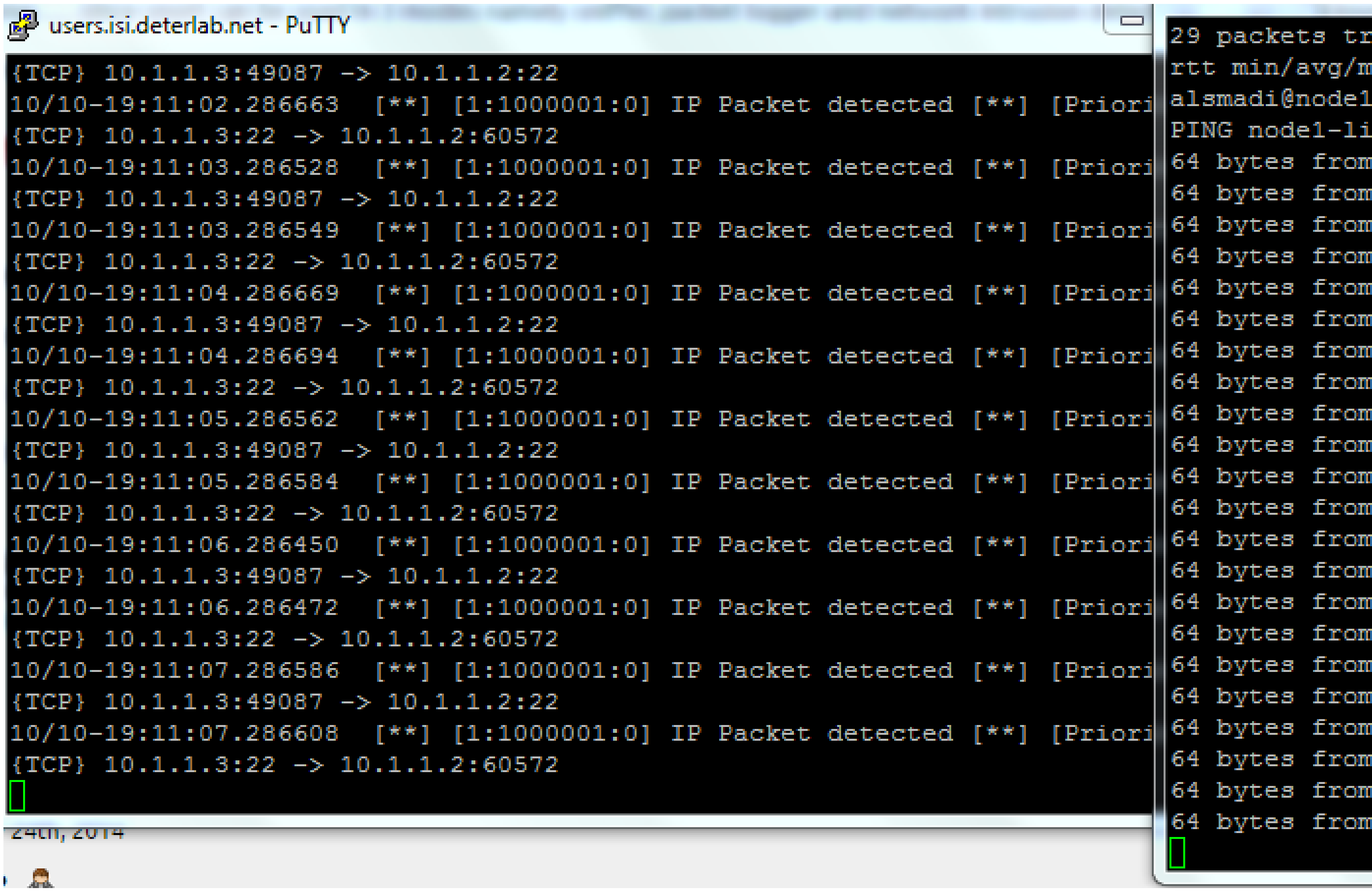
```
Type:0  Code:0  ID:19257  Seq:170  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/10-18:51:28.335221 10.1.1.3 -> 10.1.1.2
ICMP TTL:64 TOS:0x0 ID:60290 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:19257   Seq:171  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/10-18:51:28.335249 10.1.1.2 -> 10.1.1.3
ICMP TTL:64 TOS:0x0 ID:62648 IpLen:20 DgmLen:84
Type:0  Code:0  ID:19257  Seq:171  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/10-18:51:29.335128 10.1.1.3 -> 10.1.1.2
ICMP TTL:64 TOS:0x0 ID:60483 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:19257   Seq:172  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/10-18:51:29.335158 10.1.1.2 -> 10.1.1.3
ICMP TTL:64 TOS:0x0 ID:62868 IpLen:20 DgmLen:84
Type:0  Code:0  ID:19257  Seq:172  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

$sudo snort -c /etc/snort/snort.conf

Watch the output carefully, and address any errors in your config file. (Hint: some default .rules files contain deprecated format, try to comment those files in the config file). Continue re-running snort until you get it working correctly.

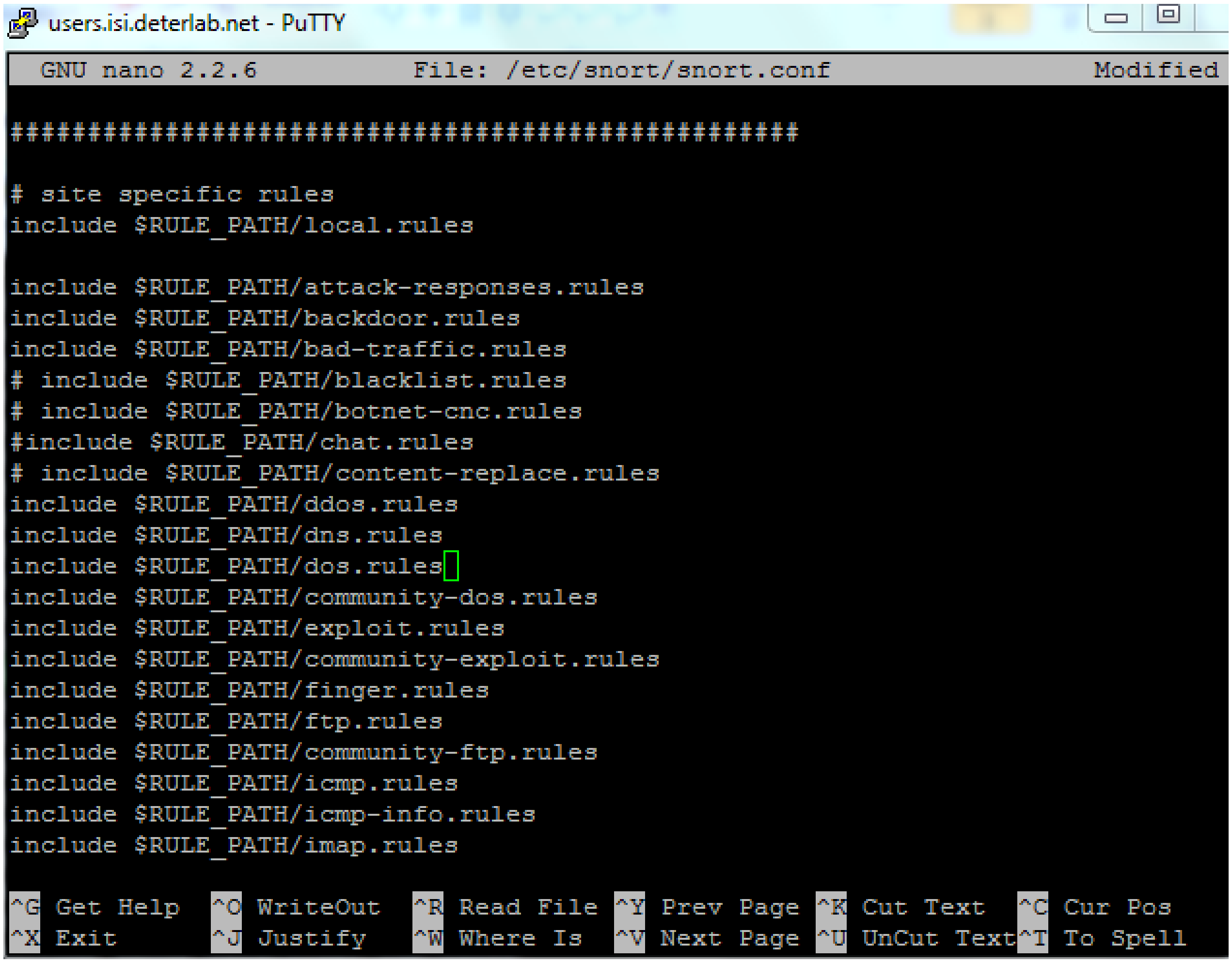
The command to run SNORT in IDS mode:

```
sudo snort -q -A console -c /etc/snort/snort.conf
```

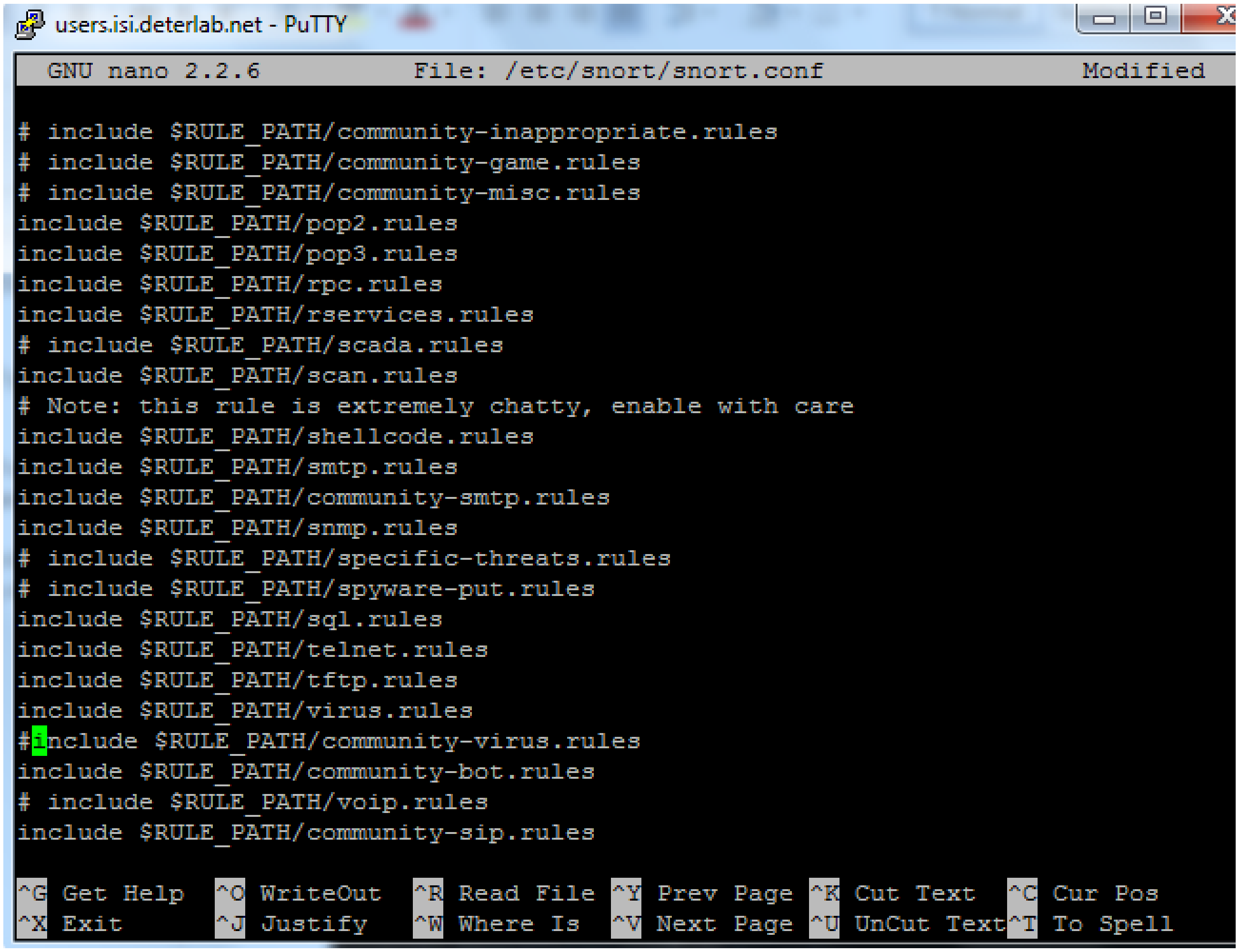


```
{TCP} 10.1.1.3:49087 -> 10.1.1.2:22
10/10-19:11:02.286663  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:22 -> 10.1.1.2:60572
10/10-19:11:03.286528  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:49087 -> 10.1.1.2:22
10/10-19:11:03.286549  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:22 -> 10.1.1.2:60572
10/10-19:11:04.286669  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:49087 -> 10.1.1.2:22
10/10-19:11:04.286694  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:22 -> 10.1.1.2:60572
10/10-19:11:05.286562  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:49087 -> 10.1.1.2:22
10/10-19:11:05.286584  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:22 -> 10.1.1.2:60572
10/10-19:11:06.286450  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:49087 -> 10.1.1.2:22
10/10-19:11:06.286472  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:22 -> 10.1.1.2:60572
10/10-19:11:07.286586  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:49087 -> 10.1.1.2:22
10/10-19:11:07.286608  [**] [1:1000001:0] IP Packet detected [**] [Priori
{TCP} 10.1.1.3:22 -> 10.1.1.2:60572
```

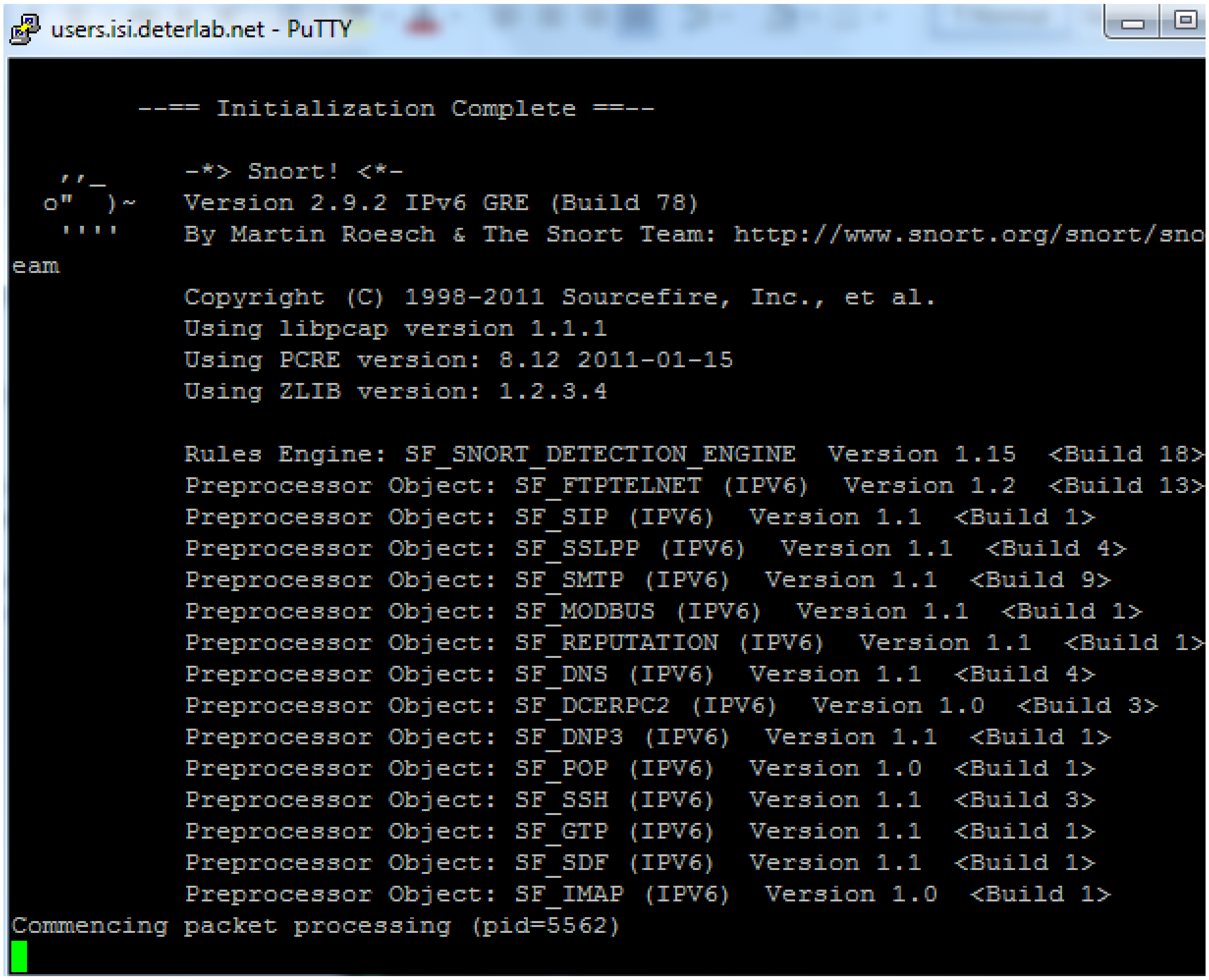
```
29 packets tr
rtt min/avg/m
alsmadi@node1
PING node1-li
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
64 bytes from
```



```
  GNU nano 2.2.6             File: /etc/snort/snort.conf                 Modified

###################################################

# site specific rules
include $RULE_PATH/local.rules

include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
# include $RULE_PATH/blacklist.rules
# include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/chat.rules
# include $RULE_PATH/content-replace.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/community-dos.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/community-exploit.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/community-ftp.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/imap.rules

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```



```
  GNU nano 2.2.6               File: /etc/snort/snort.conf                 Modified

# include $RULE_PATH/community-inappropriate.rules
# include $RULE_PATH/community-game.rules
# include $RULE_PATH/community-misc.rules
include $RULE_PATH/pop2.rules
include $RULE_PATH/pop3.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
# include $RULE_PATH/scada.rules
include $RULE_PATH/scan.rules
# Note: this rule is extremely chatty, enable with care
include $RULE_PATH/shellcode.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/community-smtp.rules
include $RULE_PATH/snmp.rules
# include $RULE_PATH/specific-threats.rules
# include $RULE_PATH/spyware-put.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/virus.rules
#include $RULE_PATH/community-virus.rules
include $RULE_PATH/community-bot.rules
# include $RULE_PATH/voip.rules
include $RULE_PATH/community-sip.rules

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text^T To Spell
```

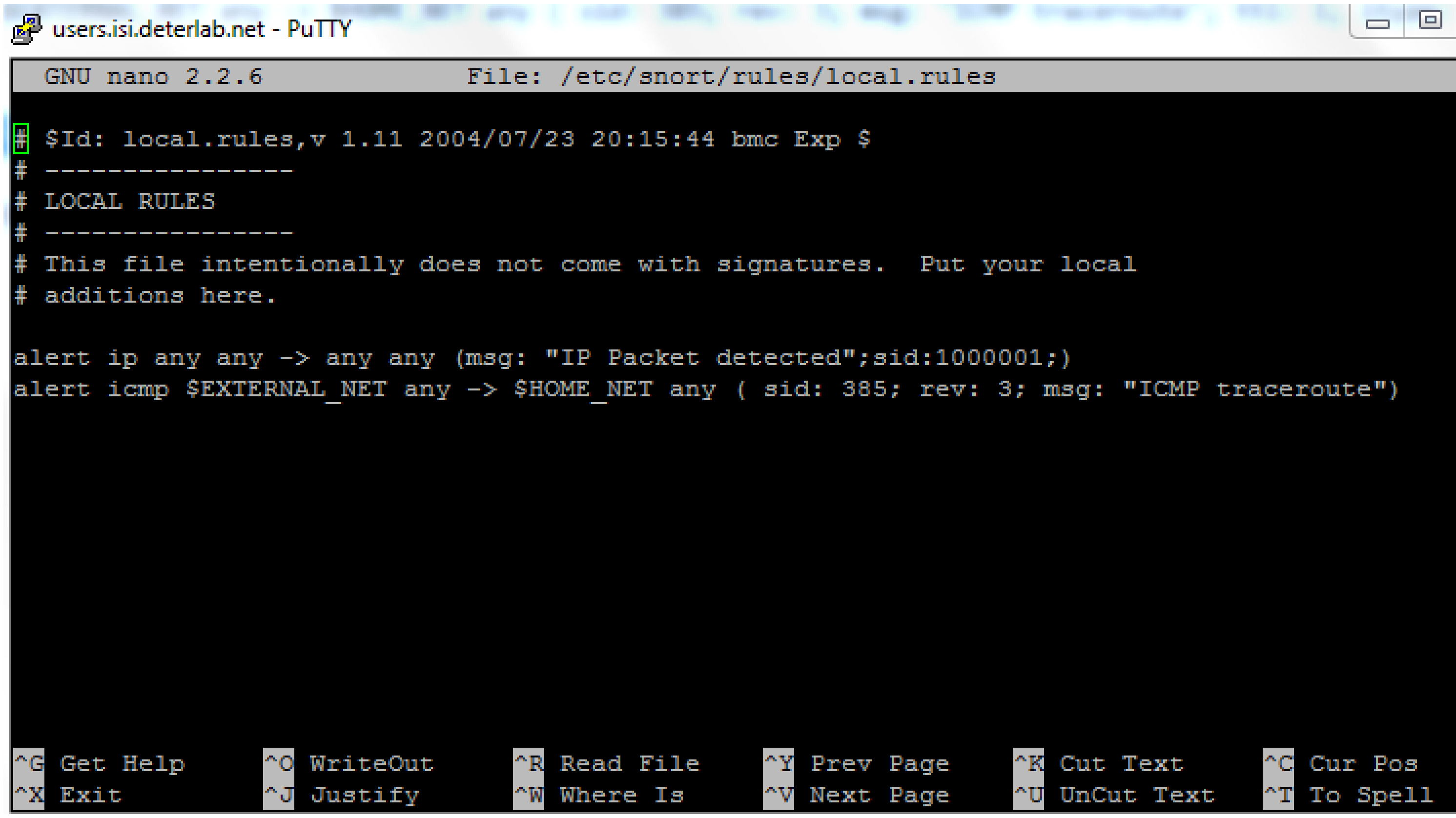


```
        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.2 IPv6 GRE (Build 78)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/sno
eam
           Copyright (C) 1998-2011 Sourcefire, Inc., et al.
           Using libpcap version 1.1.1
           Using PCRE version: 8.12 2011-01-15
           Using ZLIB version: 1.2.3.4

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 1.15  <Build 18>
           Preprocessor Object: SF_FTPTELNET (IPV6)  Version 1.2  <Build 13>
           Preprocessor Object: SF_SIP (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP (IPV6)  Version 1.1  <Build 4>
           Preprocessor Object: SF_SMTP (IPV6)  Version 1.1  <Build 9>
           Preprocessor Object: SF_MODBUS (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS (IPV6)  Version 1.1  <Build 4>
           Preprocessor Object: SF_DCERPC2 (IPV6)  Version 1.0  <Build 3>
           Preprocessor Object: SF_DNP3 (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP (IPV6)  Version 1.0  <Build 1>
           Preprocessor Object: SF_SSH (IPV6)  Version 1.1  <Build 3>
           Preprocessor Object: SF_GTP (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP (IPV6)  Version 1.0  <Build 1>
Commencing packet processing (pid=5562)
```

Step 2: Read about Snort's signature syntax in the Snort User's Manual which is located on the class wiki. In particular, be sure to review the meta-data options reference and sid. Once you are somewhat familiar with the rule language, read through some of the web attacks rules files. These are files named in the form web-*.rules under **`/etc/snort/rules/`**. Follow the references listed in a few of the rules and read about the type of attack the specific signatures are designed to detect.

The file (local.rule) is created for users to add their own rules (as its empty by default). We will add our experimental rules to this one

Image below shows an example of 2 rules added to local.rules

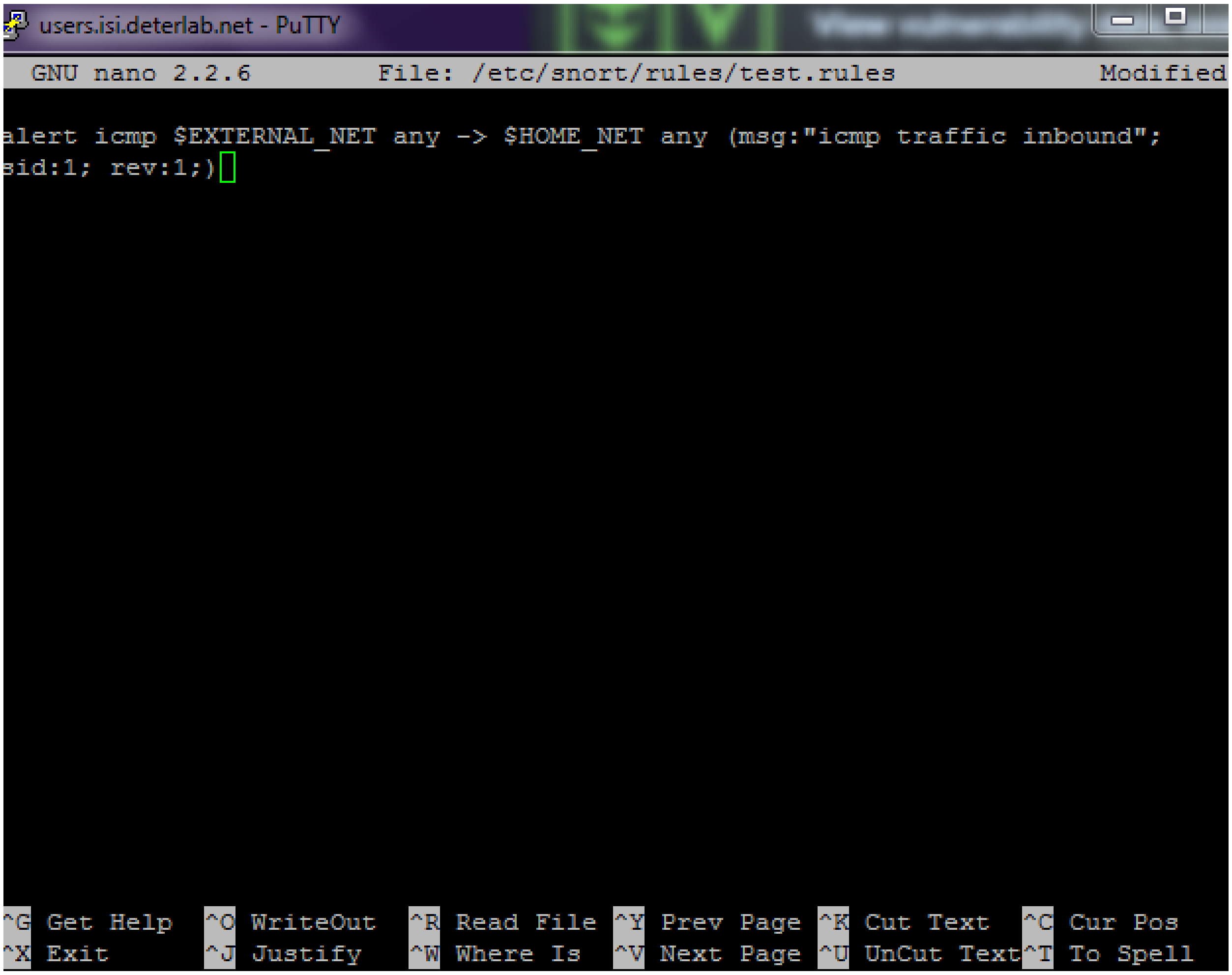


users.isi.deterlab.net - PuTTY

```
  GNU nano 2.2.6                File: /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 ip any any -> any any (msg: "IP Packet detected";sid:1000001;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any ( sid: 385; rev: 3; msg: "ICMP traceroute")
```

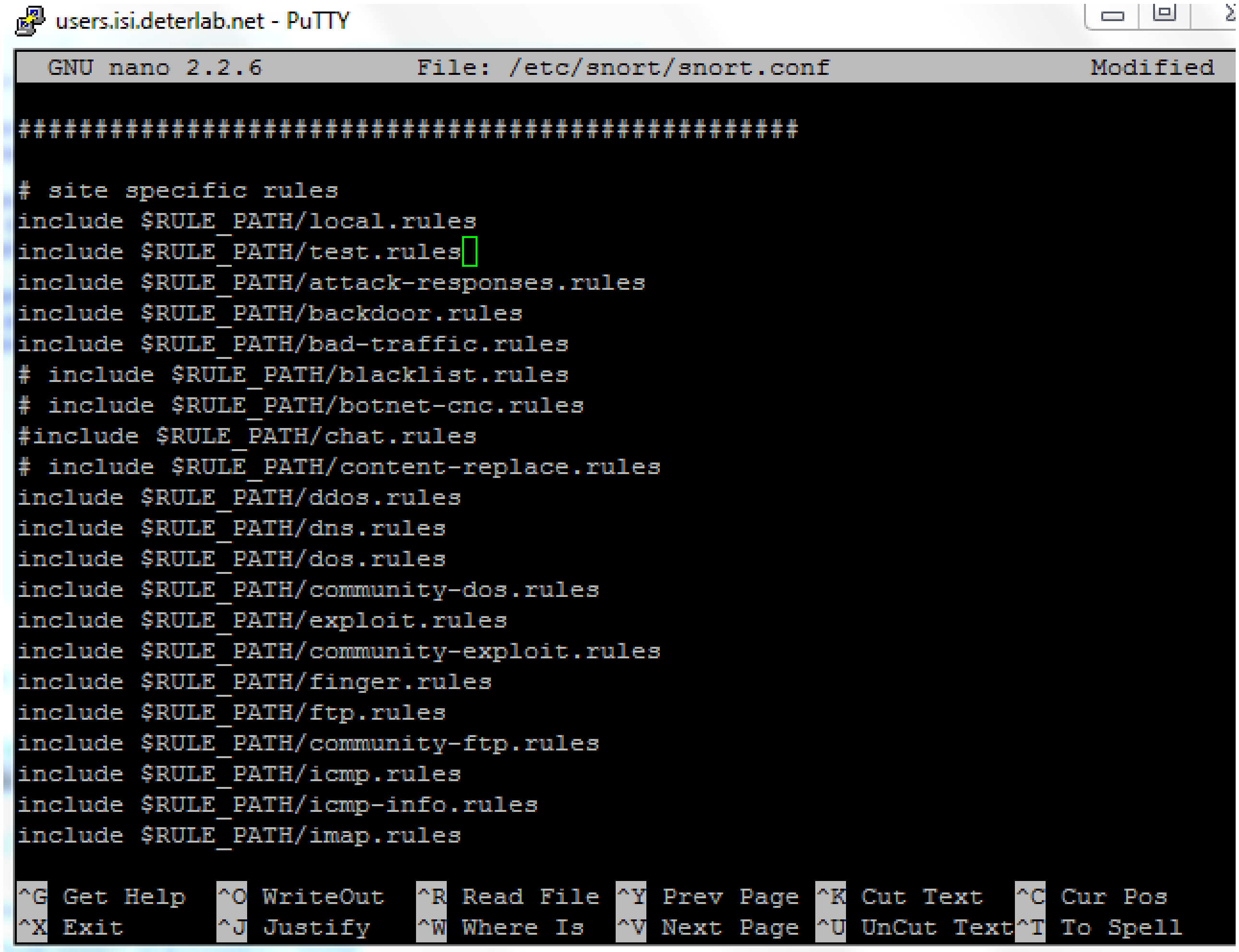
Optional : You can create a new rule file (for our testing)

Try to add rules to your new rule file



```
  GNU nano 2.2.6          File: /etc/snort/rules/test.rules                 Modified

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"icmp traffic inbound";
sid:1; rev:1;)
```

Make sure you add your rule file to config (you can do that in command line)



```
  GNU nano 2.2.6            File: /etc/snort/snort.conf                  Modified

###################################################

# site specific rules
include $RULE_PATH/local.rules
include $RULE_PATH/test.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
# include $RULE_PATH/blacklist.rules
# include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/chat.rules
# include $RULE_PATH/content-replace.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/community-dos.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/community-exploit.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/community-ftp.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/imap.rules

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Now, select two web attack signatures that seem straight-forward to understand. It would be simpler if you select a signature that looks for "evil" data in an HTTP URL string. Log into your Windows server and open a browser. Based on the documentation provided with the signature you have selected, attempt to trigger the Snort signature by making a HTTP request to which contains an attack string which should be detected.

Next, verify in your Snort logs that your attack triggered an alert based on that. (Hint:/var/log/snort/)

Step 3: Snort also allows us to write custom rules. Open the file /etc/snort/rules/local.rules and add one rule that detects each visit to www.google.com that is made by the virtual machine. The rule should look for any outbound TCP traffic that is going to port 80 and contains the pattern "www.google.com" in the URL and trigger an alert when it gets a match. Give the rule an SID of 1000000 or higher. Then visit Google with a web browser and check if your rule triggered an alert.

Record your screenshots as steps to write and test your (google) created rule

Questions:

1. In step 1, how did you modify the config file to make it work?

2. In step 2, describe the two attack signatures you chose and explain the corresponding rules against them. How did you attempt to trigger the alert? How did snort process your requests?

3. In step 3, copy/paste your new rule here. How did you confirm that your rule was enforced by snort?