How to Evade a Network Intrusion Detection System (NIDS) Using Snort

Nearly every commercial enterprise worth hacking has an intrusion detection system (IDS). These network intrusion detection systems are designed to detect any malicious activity on the network. That means you!

As the name implies, a network intrusion detection system (NIDS) is intended to alert the system administrator of network-based intrusions. As a hacker, the better we understand how these NIDS work, the better we can evade them and stealthily enter and exit a network without detection. In an attempt to train you to evade these systems, I am beginning new series on how NIDS work.

Introducing Snort: Our NIDS of Choice

Snort is an open-source NIDS that is the most widely used NIDS in the world. Some estimate its market share at over 60%. It's used by such large organizations as Verizon, AT&T, the U.S. State Department, most U.S. military bases, and millions of medium to large businesses around the globe. Last month (July 2013), Cisco announced that they would be acquiring the parent company of Snort, Sourcefire Inc. of Columbia, MD. This insures that Snort will remain the dominant NIDS on the planet for some time to come, making it increasingly important that we understand Snort—so we can evade it.

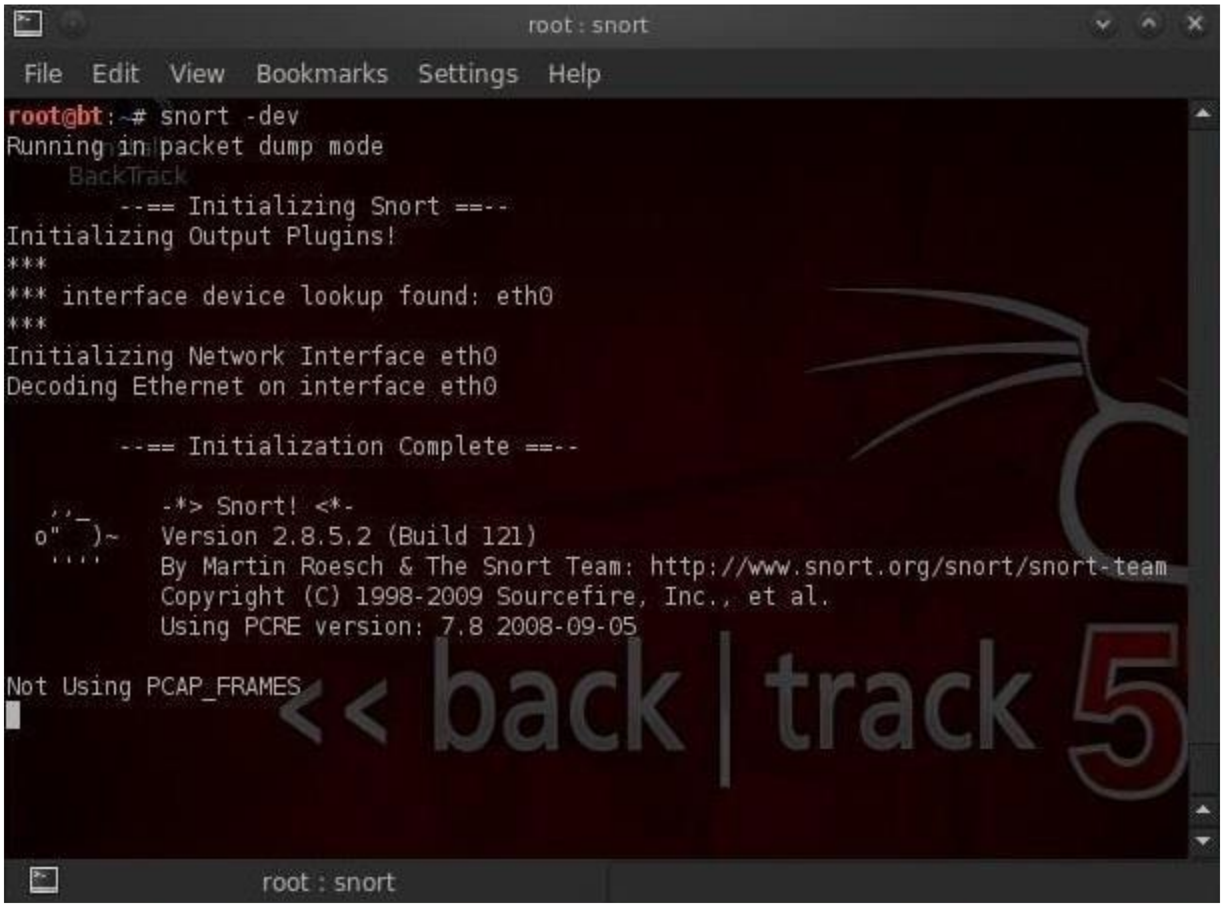
Image via wordpress.com

Fortunately, Snort is built into our BackTrack, so we don't need to install it. If you do need to download it, you can find it here.

Step 1 Fire up Snort

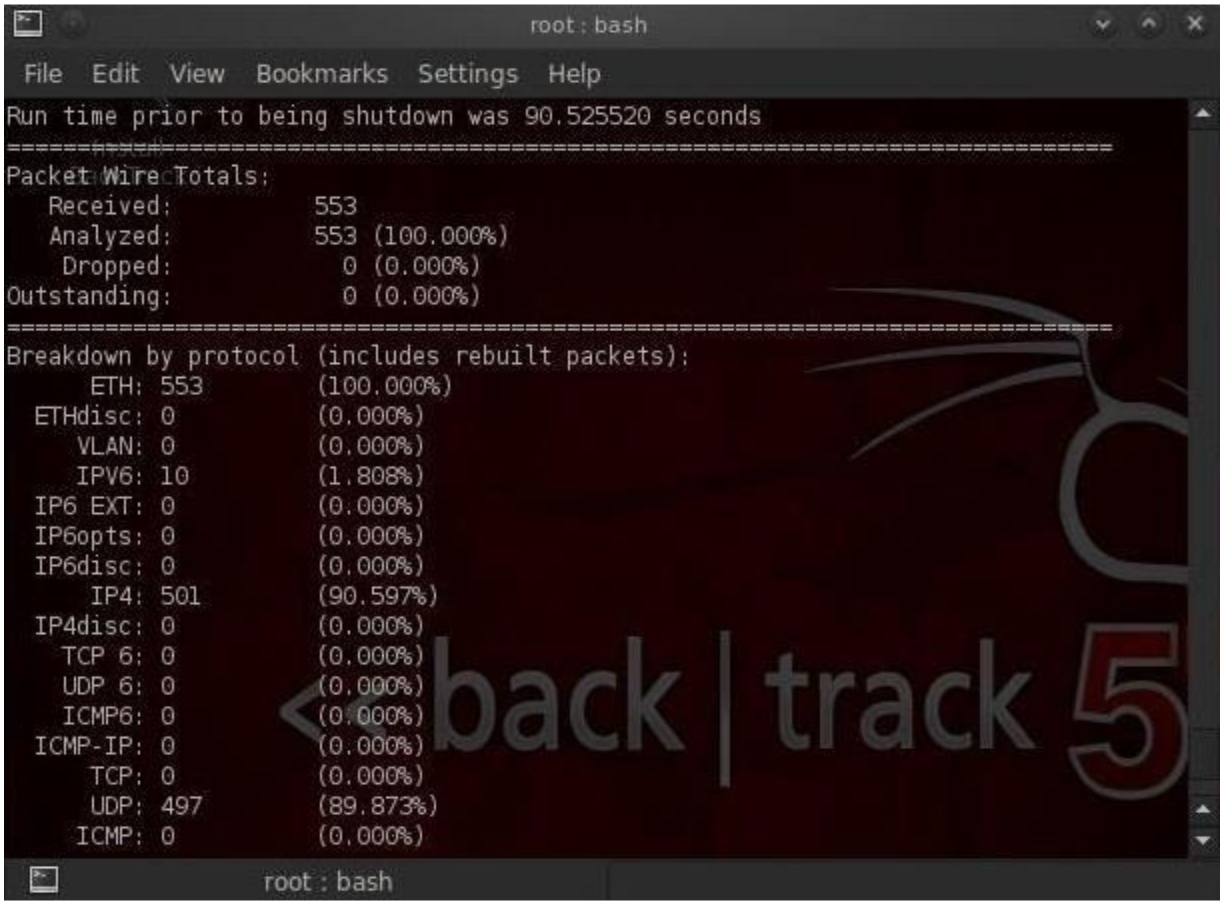
Snort is basically a network traffic sniffer that can apply rules to the traffic it sees to determine whether it contains malicious traffic. We can start Snort in sniffer mode by opening any terminal in BackTrack and typing:

- **snort -vde**



```
root@bt:~# snort -dev
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
***
*** interface device lookup found: eth0
***
Initializing Network Interface eth0
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.8.5.2 (Build 121)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2009 Sourcefire, Inc., et al.
           Using PCRE version: 7.8 2008-09-05

Not Using PCAP_FRAMES
```

After we hit enter, we begin to see packets going past the screen in rapid succession. Snort is simply sniffing packets from the wire and displaying them to us.



```
Run time prior to being shutdown was 90.525520 seconds
===============================================================================
Packet Wire Totals:
   Received:           553
   Analyzed:           553 (100.000%)
    Dropped:             0 (0.000%)
Outstanding:             0 (0.000%)
===============================================================================
Breakdown by protocol (includes rebuilt packets):
      ETH: 553        (100.000%)
  ETHdisc: 0          (0.000%)
     VLAN: 0          (0.000%)
     IPV6: 10         (1.808%)
  IP6 EXT: 0          (0.000%)
  IP6opts: 0          (0.000%)
  IP6disc: 0          (0.000%)
      IP4: 501        (90.597%)
  IP4disc: 0          (0.000%)
    TCP 6: 0          (0.000%)
    UDP 6: 0          (0.000%)
    ICMP6: 0          (0.000%)
  ICMP-IP: 0          (0.000%)
      TCP: 0          (0.000%)
      UDP: 497        (89.873%)
     ICMP: 0          (0.000%)
```

To stop Snort, hit the **Control C**. When we stop Snort, it displays our statistics on the packet capture.

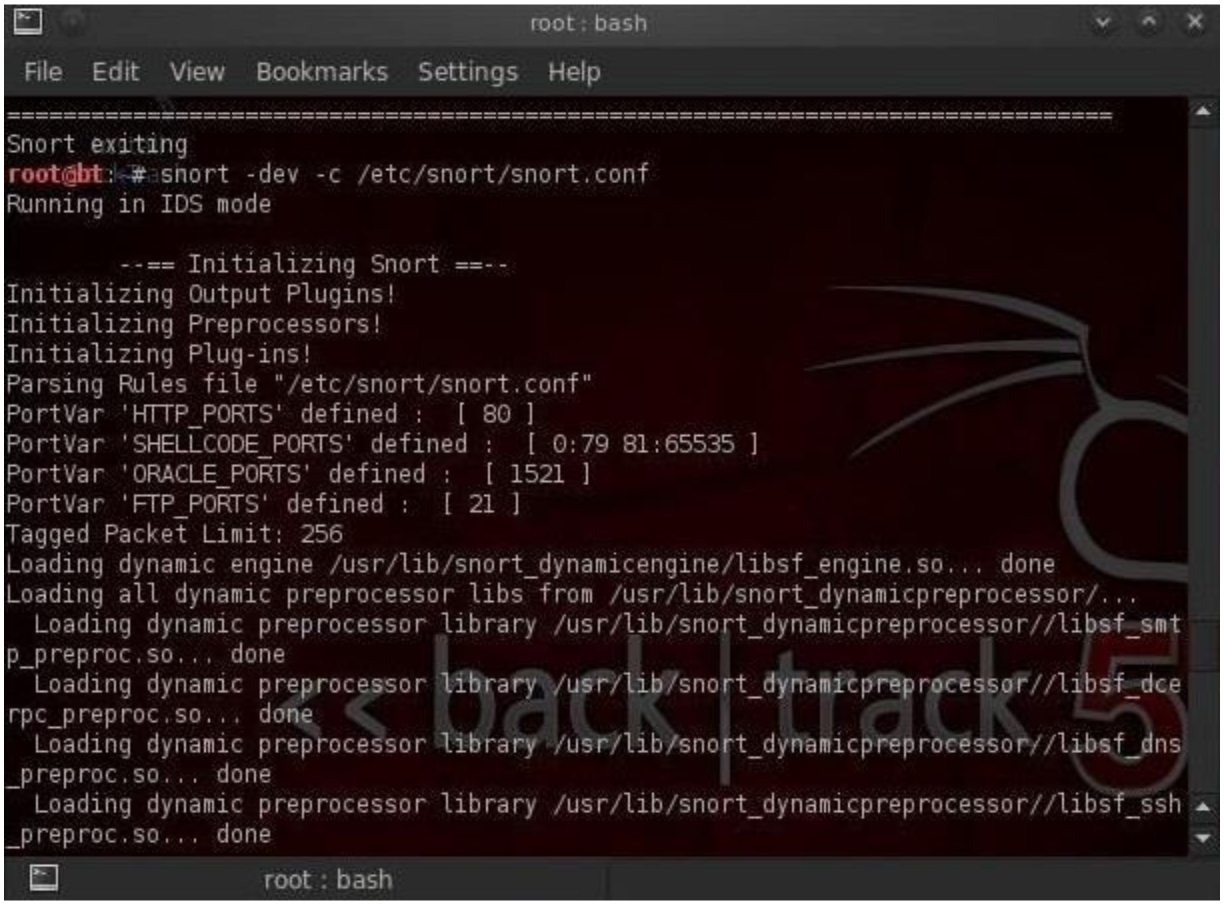
Step 2 Intrusion Detection Mode

To get Snort to operate in Intrusion Detection (IDS) mode, we need to get Snort to use its configuration file. Nearly all applications in Linux are controlled by a configuration file that is a simple text file. This same applies to Snort. Snort's configuration file is named **snort.conf** and is usually found at **/etc/snort/snort.conf**. So, to get Snort to use its configuration file, we need to start it with:

- **snort -vde -c /etc/snort/snort.conf**

Where **-c** says use the configuration file, and **/etc/snort/snort.conf** is the location of the configuration file.

When Snort starts in IDS mode, we begin to see a screen similar to that below. Eventually, the screen will stop scrolling and Snort will begin to watch your network traffic.



```
===============================================================================
Snort exiting
root@bt:~# snort -dev -c /etc/snort/snort.conf
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1521 ]
PortVar 'FTP_PORTS' defined :  [ 21 ]
Tagged Packet Limit: 256
Loading dynamic engine /usr/lib/snort_dynamicengine/libsf_engine.so... done
Loading all dynamic preprocessor libs from /usr/lib/snort_dynamicpreprocessor/...
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_smt
p_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_dce
rpc_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_dns
_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf_ssh
_preproc.so... done
```

Now Snort is sniffing our wire and will alert when something malicious appears!

Step 3 Configuring Snort

Snort comes with a default configuration file that, for the most part, will work with little editing. The configuration has plenty of comments to explain what each line and section does, so you can figure it out with little outside assistance.

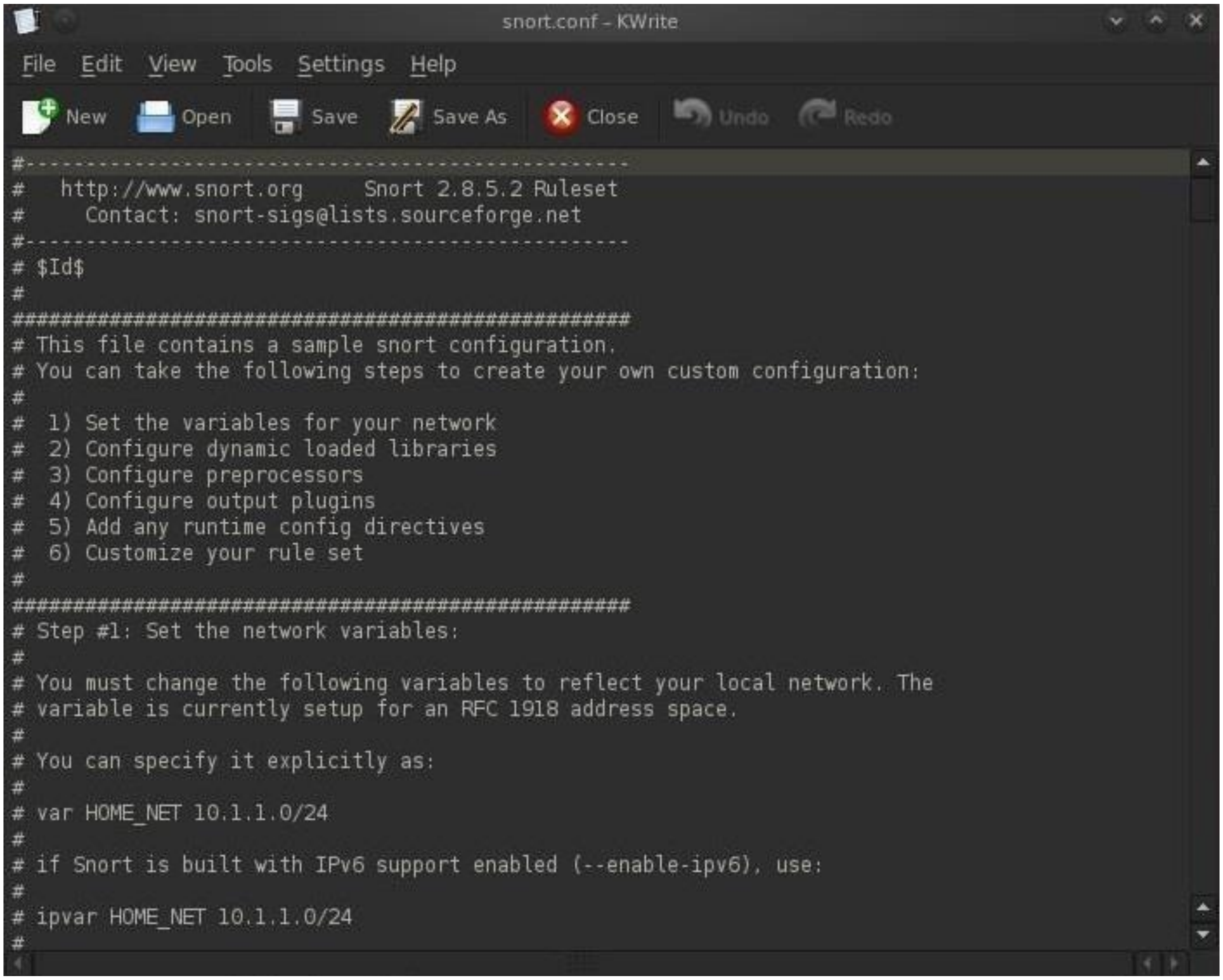
There are at least 3 areas, though, that need some attention and configuring...

1. The EXTERNAL_NET variable
2. The HOME_NET variable
3. The path to the Snort rules

Without the Snort rules, Snort is just a sniffer/packet logger, far from the powerful IDS it can be. That being said, let's get inside that Snort configuration file and make the minimum changes to get Snort to run as an effective IDS.

Let's open the snort configuration file with **KWrite**.

- **kwrite /etc/snort/snort.conf**


snort.conf - KWrite
File Edit View Tools Settings Help
New Open Save Save As Close Undo Redo

```
#--------------------------------------------------
#   http://www.snort.org     Snort 2.8.5.2 Ruleset
#     Contact: snort-sigs@lists.sourceforge.net
#--------------------------------------------------
# $Id$
#
###################################################
# This file contains a sample snort configuration.
# You can take the following steps to create your own custom configuration:
#
#  1) Set the variables for your network
#  2) Configure dynamic loaded libraries
#  3) Configure preprocessors
#  4) Configure output plugins
#  5) Add any runtime config directives
#  6) Customize your rule set
#
###################################################
# Step #1: Set the network variables:
#
# You must change the following variables to reflect your local network. The
# variable is currently setup for an RFC 1918 address space.
#
# You can specify it explicitly as:
#
# var HOME_NET 10.1.1.0/24
#
# if Snort is built with IPv6 support enabled (--enable-ipv6), use:
#
# ipvar HOME_NET 10.1.1.0/24
#
```

As you can see in the screenshot above, the configuration file is comprised of six (6) sections.

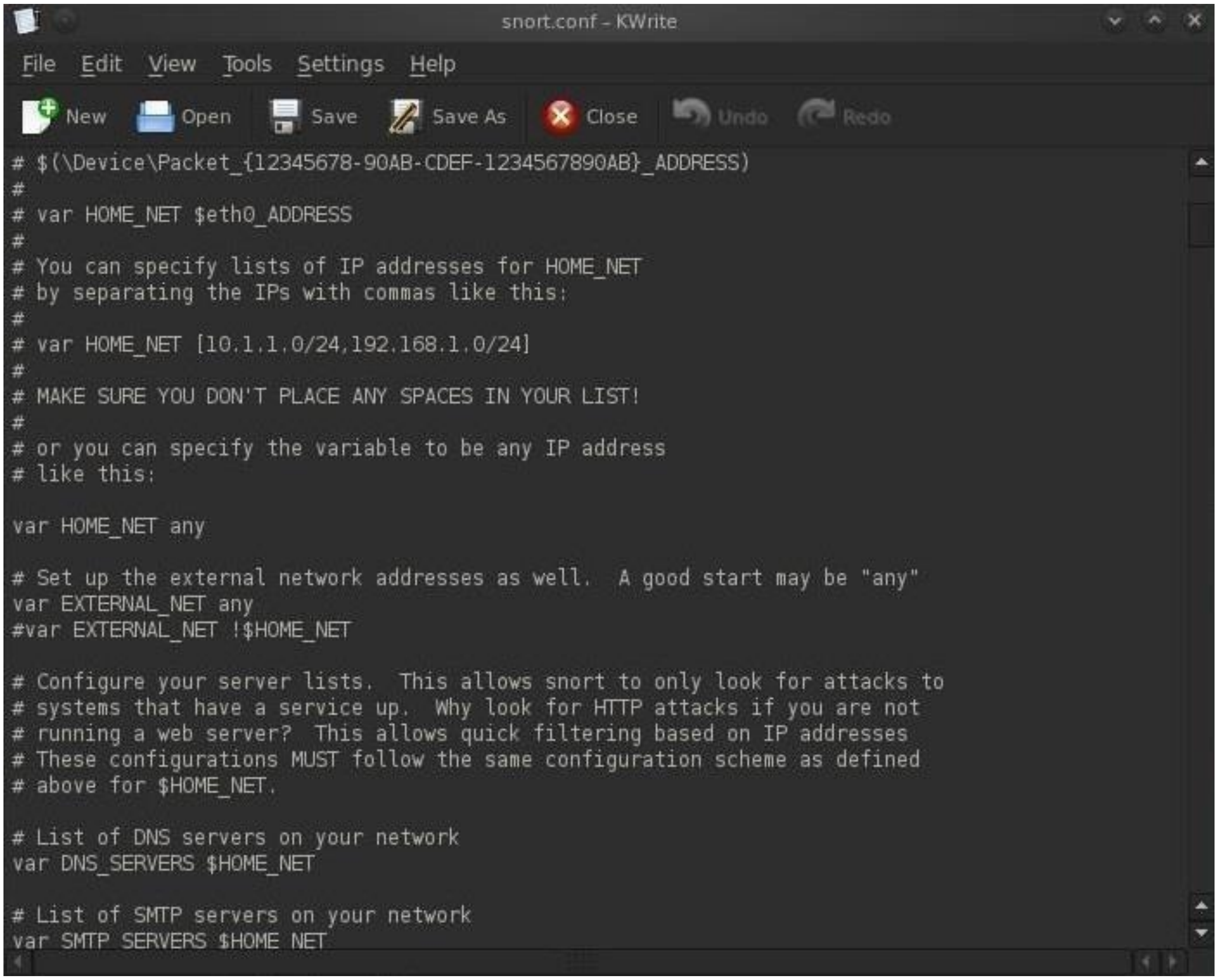
1. Set the variables on your network
2. Configure dynamic loaded libraries
3. Configure preprocessors
4. Configure output plugins
5. Add any runtime config directives
6. Customize your rule set

We need to first set the variables for our internal and external network. These are defined by the lines:

- **var HOME_NET**
- **var EXTERNAL_NET**

We can define our **HOME_NET** as the IP address or subnet we're trying to protect. You see in the screenshot that it's set as "any." This will work, but it's not optimal for detecting malicious activity. We should set the **HOME_NET** to our internal IP address, such as 192.168.1.1, or our internal subnet, such as 192.168.1.0/24.

In most cases, security admins will define their **EXTERNAL_NET** as everything that is NOT their **HOME_NET**. To accomplish this, we can simply negate (**!**) the **HOME_NET** or **! HOME_NET**.

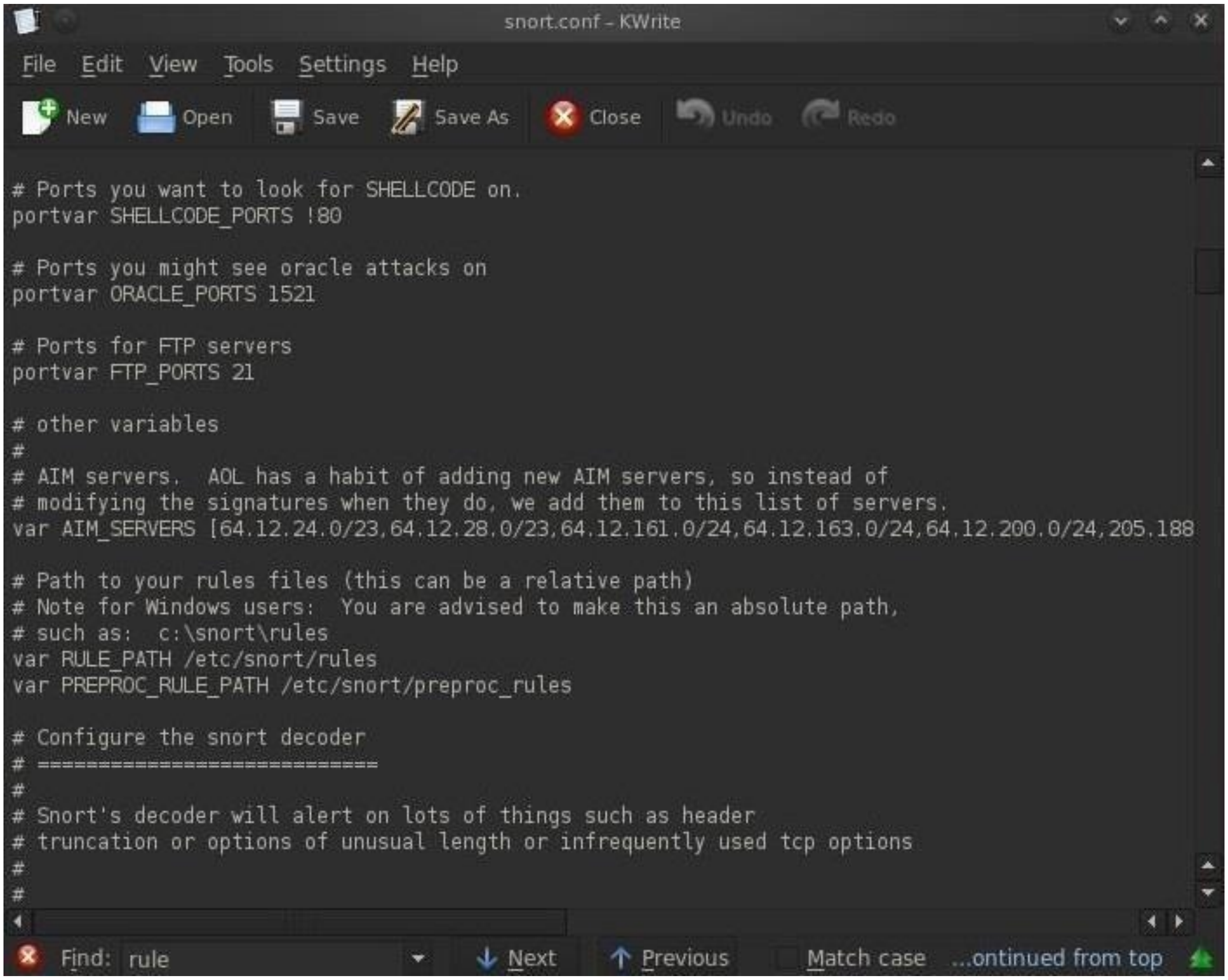

snort.conf - KWrite
File Edit View Tools Settings Help
New Open Save Save As Close Undo Redo

```
# $(\Device\Packet_{12345678-90AB-CDEF-1234567890AB}_ADDRESS)
#
# var HOME_NET $eth0_ADDRESS
#
# You can specify lists of IP addresses for HOME_NET
# by separating the IPs with commas like this:
#
# var HOME_NET [10.1.1.0/24,192.168.1.0/24]
#
# MAKE SURE YOU DON'T PLACE ANY SPACES IN YOUR LIST!
#
# or you can specify the variable to be any IP address
# like this:

var HOME_NET any

# Set up the external network addresses as well.  A good start may be "any"
var EXTERNAL_NET any
#var EXTERNAL_NET !$HOME_NET

# Configure your server lists.  This allows snort to only look for attacks to
# systems that have a service up.  Why look for HTTP attacks if you are not
# running a web server?  This allows quick filtering based on IP addresses
# These configurations MUST follow the same configuration scheme as defined
# above for $HOME_NET.

# List of DNS servers on your network
var DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
var SMTP_SERVERS $HOME_NET
```

Next, we need to set our path to our rules. As we can see in the screenshot below, about two-thirds of the way down, there is:

- **var RULE_PATH /etc/snort/rules**

In most installations, this path will be correct (but does vary with different installations) and we can simply leave it as is, but make certain that your rules are in this path before assuming so. When you are done, simply save the snort.conf file.


snort.conf - KWrite
File Edit View Tools Settings Help
New Open Save Save As Close Undo Redo
Find: rule
Next Previous Match case ...ontinued from top

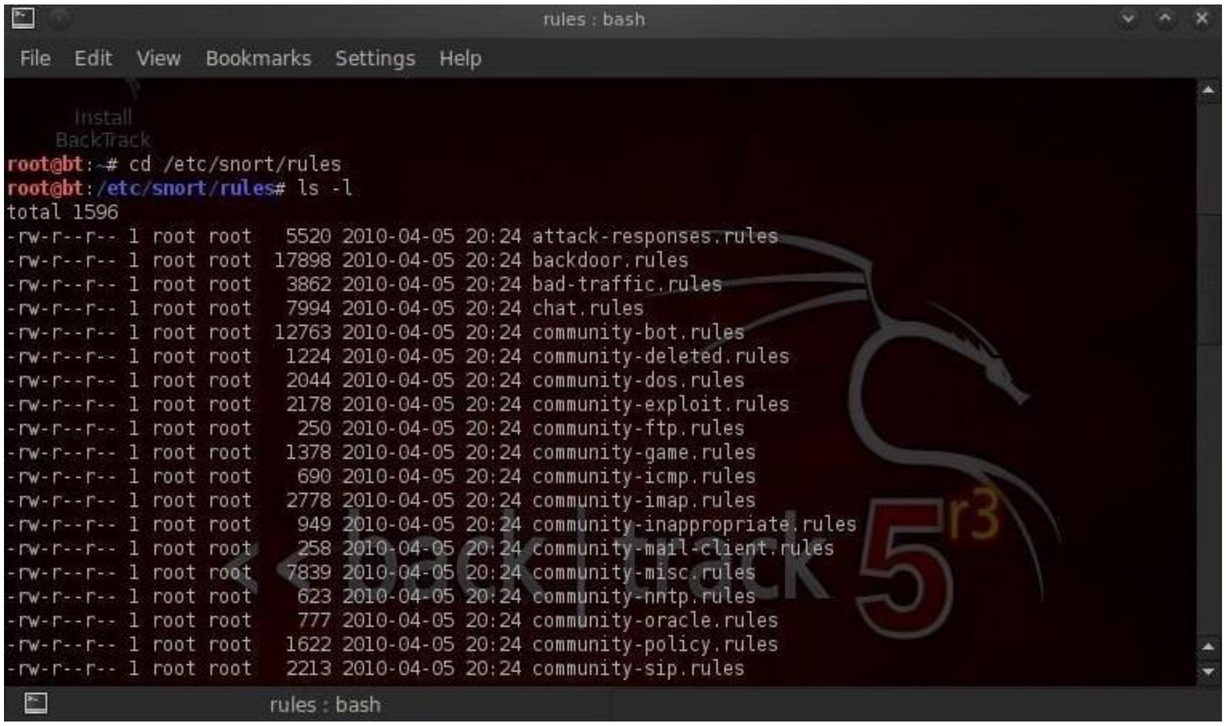
```
# Ports you want to look for SHELLCODE on.
portvar SHELLCODE_PORTS !80

# Ports you might see oracle attacks on
portvar ORACLE_PORTS 1521

# Ports for FTP servers
portvar FTP_PORTS 21

# other variables
#
# AIM servers.  AOL has a habit of adding new AIM servers, so instead of
# modifying the signatures when they do, we add them to this list of servers.
var AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188

# 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/snort/rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# Configure the snort decoder
# ============================
#
# Snort's decoder will alert on lots of things such as header
# truncation or options of unusual length or infrequently used tcp options
#
#
```

Step 4 Checking the Snort Rules

We can navigate to the rules directory by typing these two commands:

- **cd /etc/snort/rules**
- **ls -l**

In this way, we can see all of the files that comprise our Snort rules. It's these Snort rules that are designed to catch intrusions and alert the security admin.



```
root@bt:~# cd /etc/snort/rules
root@bt:/etc/snort/rules# ls -l
total 1596
-rw-r--r-- 1 root root   5520 2010-04-05 20:24 attack-responses.rules
-rw-r--r-- 1 root root  17898 2010-04-05 20:24 backdoor.rules
-rw-r--r-- 1 root root   3862 2010-04-05 20:24 bad-traffic.rules
-rw-r--r-- 1 root root   7994 2010-04-05 20:24 chat.rules
-rw-r--r-- 1 root root  12763 2010-04-05 20:24 community-bot.rules
-rw-r--r-- 1 root root   1224 2010-04-05 20:24 community-deleted.rules
-rw-r--r-- 1 root root   2044 2010-04-05 20:24 community-dos.rules
-rw-r--r-- 1 root root   2178 2010-04-05 20:24 community-exploit.rules
-rw-r--r-- 1 root root    250 2010-04-05 20:24 community-ftp.rules
-rw-r--r-- 1 root root   1378 2010-04-05 20:24 community-game.rules
-rw-r--r-- 1 root root    690 2010-04-05 20:24 community-icmp.rules
-rw-r--r-- 1 root root   2778 2010-04-05 20:24 community-imap.rules
-rw-r--r-- 1 root root    949 2010-04-05 20:24 community-inappropriate.rules
-rw-r--r-- 1 root root    258 2010-04-05 20:24 community-mail-client.rules
-rw-r--r-- 1 root root   7839 2010-04-05 20:24 community-misc.rules
-rw-r--r-- 1 root root    623 2010-04-05 20:24 community-nntp.rules
-rw-r--r-- 1 root root    777 2010-04-05 20:24 community-oracle.rules
-rw-r--r-- 1 root root   1622 2010-04-05 20:24 community-policy.rules
-rw-r--r-- 1 root root   2213 2010-04-05 20:24 community-sip.rules
```