Khoa Khoa học & Kỹ thuật máy tính

Trường ĐH Bách Khoa TP.HCM

Cryptography and Network Security
Lab 9
Snort Intrusion Detection Systems

INTRODUCTION

In this lab students will explore the Snort Intrusion Detection Systems. The students will study Snort IDS, a signature based intrusion detection system used to detect network attacks. Snort can also be used as a simple packet logger. For the purpose of this lab the students will use snort as a packet sniffer and write their own IDS rules.

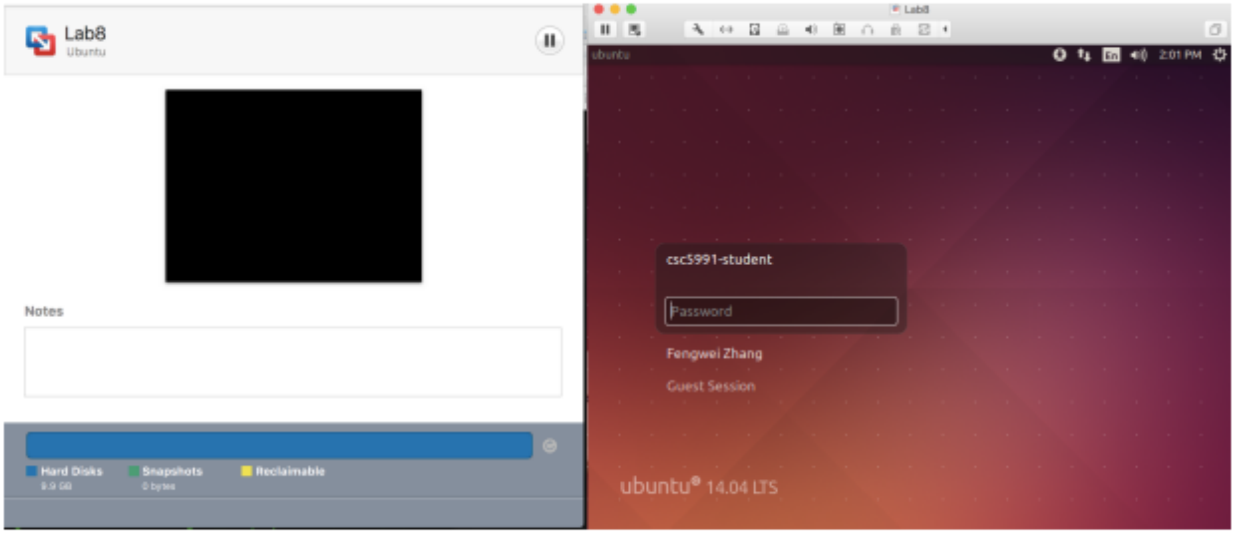
PRACTICE

Software Requirements

All required files are packed and configured in the provided virtual machine image.

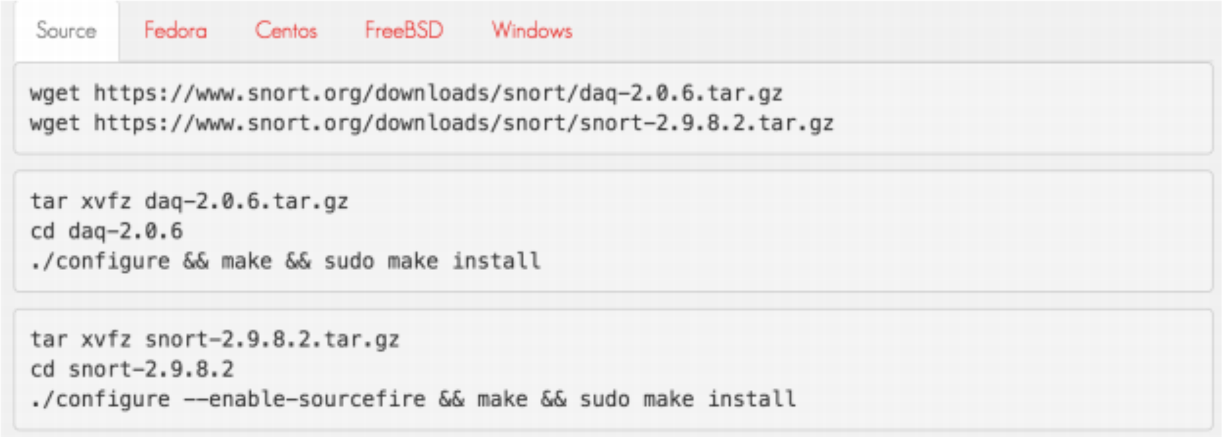
- The VMWare Software
  http://apps.eng.wayne.edu/MPStudents/Dreamspark.aspx
- The Ubuntu 14.04 Long Term Support (LTS) Version
  http://www.ubuntu.com/download/desktop
- Snort: A signature-based Intrusion Detection System
  https://www.snort.org/ - get-started

In this lab, we use Ubuntu as our VM image.



Installing Snort into the Operating System

In our Lab 8 Ubuntu VM image, the snort has been installed and setup for you. If you want to use your own version of the image, you need to install snort into the operating system. To install the latest version of the snort, you can follow the installation instruction from the snort website. Note that installation instructions are vary from OSes. The instruction below shows how to install snort from its source code on Linux.



```
wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
wget https://www.snort.org/downloads/snort/snort-2.9.8.2.tar.gz
```

```
tar xvfz daq-2.0.6.tar.gz
cd daq-2.0.6
./configure && make && sudo make install
```

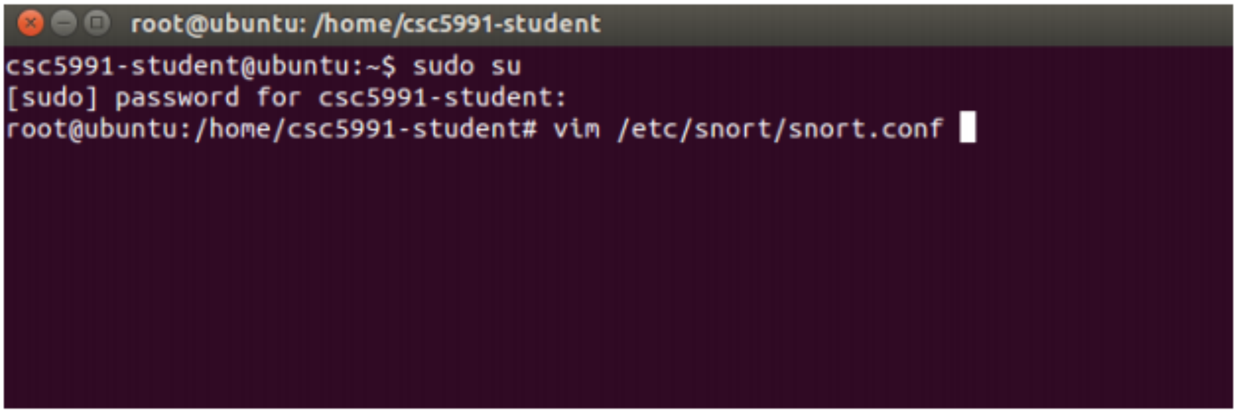
```
tar xvfz snort-2.9.8.2.tar.gz
cd snort-2.9.8.2
./configure --enable-sourcefire && make && sudo make install
```

You can find more information here:
https://www.snort.org/ - get-started

While you install the snort, you system may miss some libraries. You need to install the required libraries, too.

Configuring and Starting the Snort IDS

After installing the Snort, we need to configure it. The configuration file of snort is stored at /etc/snort/snort.conf. The screenshot below shows the commands to configure the Snort. You need to switch to root to gain the permission to read the snort configurations file.



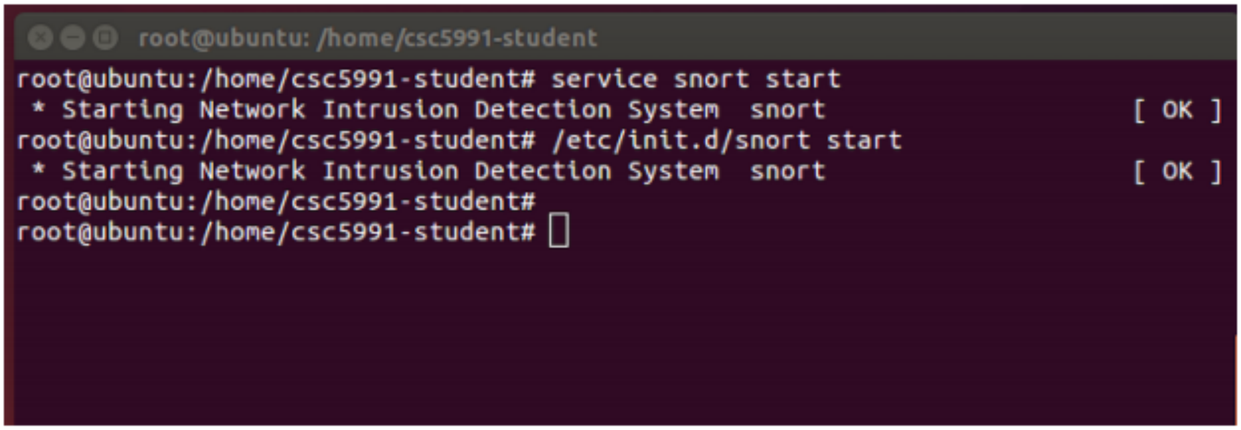
```
csc5991-student@ubuntu:~$ sudo su
[sudo] password for csc5991-student:
root@ubuntu:/home/csc5991-student# vim /etc/snort/snort.conf
```

After configuring the Snort, you need to start the Snort. You can simply type the following command to start the service.

$ service snort start

or

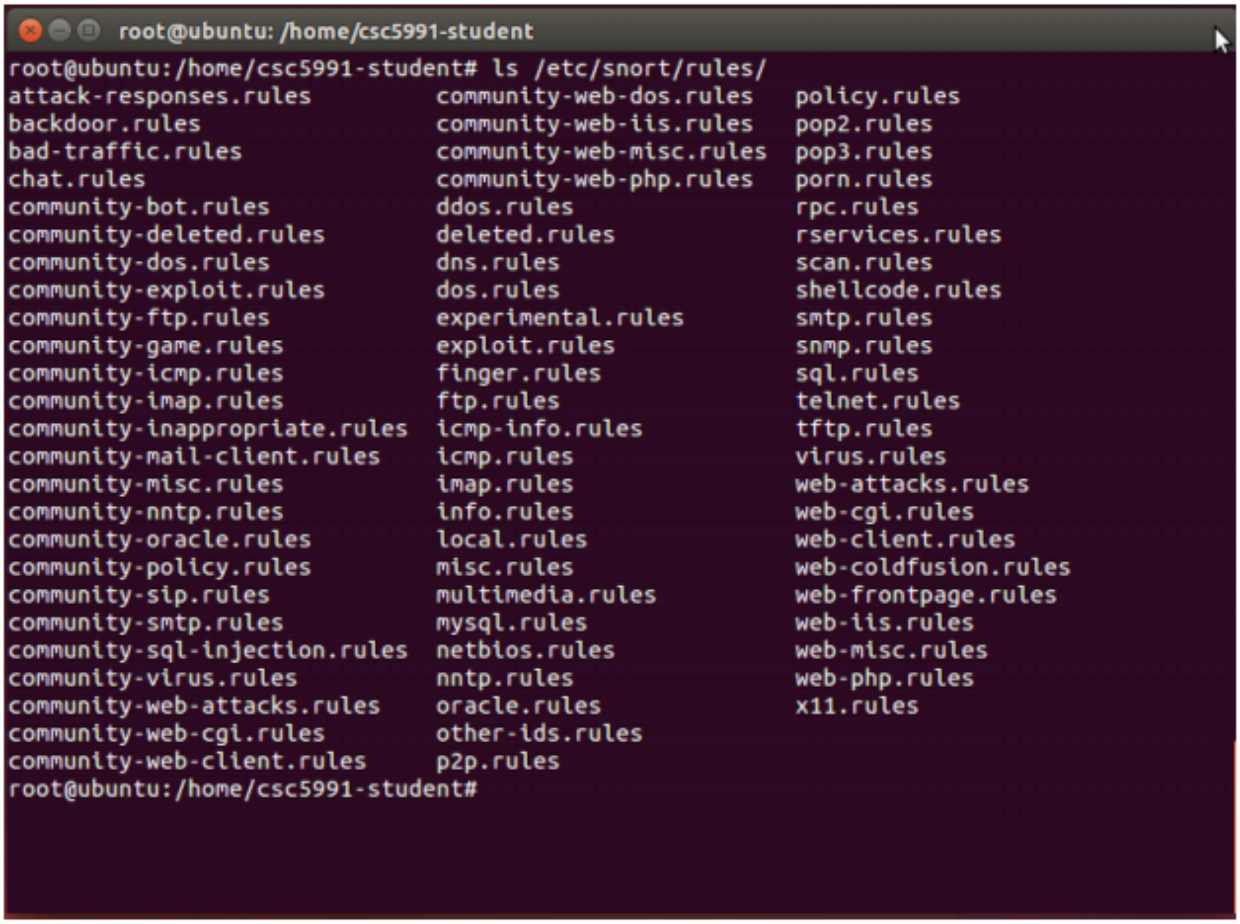
$ /etc/init.d/snort start



```
root@ubuntu:/home/csc5991-student# service snort start
 * Starting Network Intrusion Detection System  snort                    [ OK ]
root@ubuntu:/home/csc5991-student# /etc/init.d/snort start
 * Starting Network Intrusion Detection System  snort                    [ OK ]
root@ubuntu:/home/csc5991-student#
root@ubuntu:/home/csc5991-student#
```

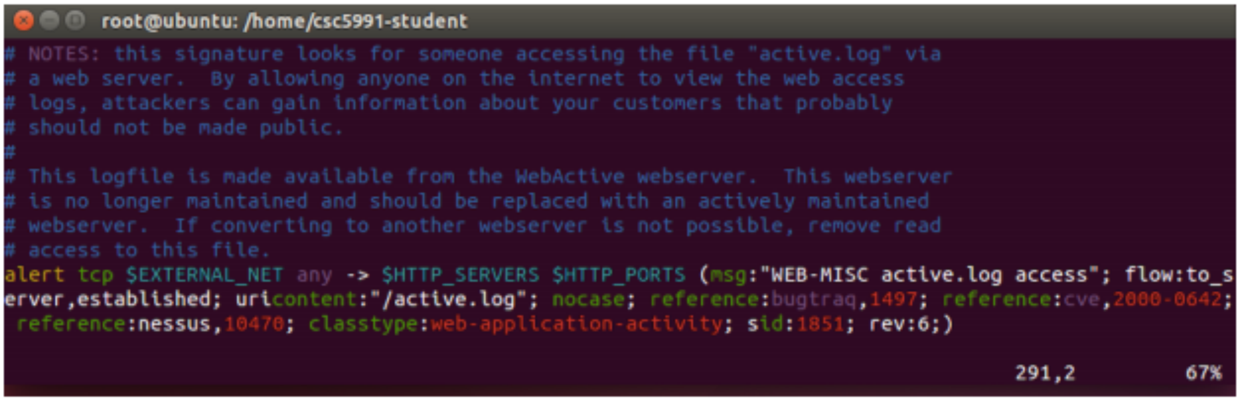
Snort Rules

Snort is a signature-based IDS, and it defines rules to detect the intrusions. All rules of Snort are stored under /etc/snort/rules directory. The screenshot below shows the files that contain rules of Snort.



```
root@ubuntu:/home/csc5991-student# ls /etc/snort/rules/
attack-responses.rules         community-web-dos.rules   policy.rules
backdoor.rules                 community-web-iis.rules   pop2.rules
bad-traffic.rules              community-web-misc.rules  pop3.rules
chat.rules                     community-web-php.rules   porn.rules
community-bot.rules            ddos.rules                rpc.rules
community-deleted.rules        deleted.rules             rservices.rules
community-dos.rules            dns.rules                 scan.rules
community-exploit.rules        dos.rules                 shellcode.rules
community-ftp.rules            experimental.rules        smtp.rules
community-game.rules           exploit.rules             snmp.rules
community-icmp.rules           finger.rules              sql.rules
community-imap.rules           ftp.rules                 telnet.rules
community-inappropriate.rules  icmp-info.rules           tftp.rules
community-mail-client.rules    icmp.rules                virus.rules
community-misc.rules           imap.rules                web-attacks.rules
community-nntp.rules           info.rules                web-cgi.rules
community-oracle.rules         local.rules               web-client.rules
community-policy.rules         misc.rules                web-coldfusion.rules
community-sip.rules            multimedia.rules          web-frontpage.rules
community-smtp.rules           mysql.rules               web-iis.rules
community-sql-injection.rules  netbios.rules             web-misc.rules
community-virus.rules          nntp.rules                web-php.rules
community-web-attacks.rules    oracle.rules              x11.rules
community-web-cgi.rules        other-ids.rules
community-web-client.rules     p2p.rules
root@ubuntu:/home/csc5991-student#
```

The screenshot below shows a real rule in the /etc/snort/rules/web-misc.rules. The slides of Lab 8 has more information about Snort rules including syntax and format.

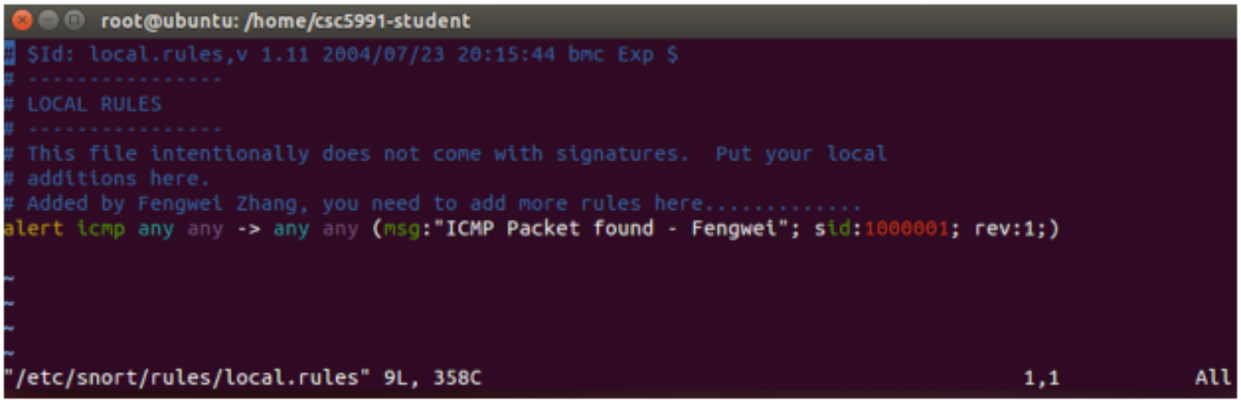


```
# NOTES: this signature looks for someone accessing the file "active.log" via
# a web server.  By allowing anyone on the internet to view the web access
# logs, attackers can gain information about your customers that probably
# should not be made public.
#
# This logfile is made available from the WebActive webserver.  This webserver
# is no longer maintained and should be replaced with an actively maintained
# webserver.  If converting to another webserver is not possible, remove read
# access to this file.
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC active.log access"; flow:to_s
erver,established; uricontent:"/active.log"; nocase; reference:bugtraq,1497; reference:cve,2000-0642;
 reference:nessus,10470; classtype:web-application-activity; sid:1851; rev:6;)

                                                                          291,2          67%
```

Writing and Adding a Snort Rule

Next, we are going to add a simple snort rule. You should add your own rules at /etc/snort/rules/local.rules.

Add the following line into the local.rules file alert icmp any any -> any any (msg:"ICMP Packet found"; sid:1000001; rev:1;), this rule defines that an alert will be logged if an ICMP packet is found. The ICMP packet could be from any IP address and the rule ID is 1000001. Make sure to pick a SID greater 1000000 for your own rules. The screenshot below shows the contents of the local.rules file after adding the rule.



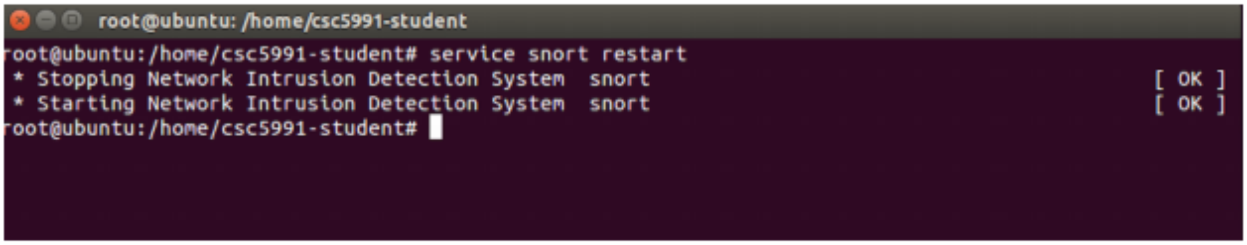
```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
# Added by Fengwei Zhang, you need to add more rules here.............
alert icmp any any -> any any (msg:"ICMP Packet found - Fengwei"; sid:1000001; rev:1;)

~
~
~
~
"/etc/snort/rules/local.rules" 9L, 358C                               1,1           All
```

To make the rule become effective, you need to restart the snort service by typing the following command.

$ service snort restart

or

$ /etc/init.d/snort restart



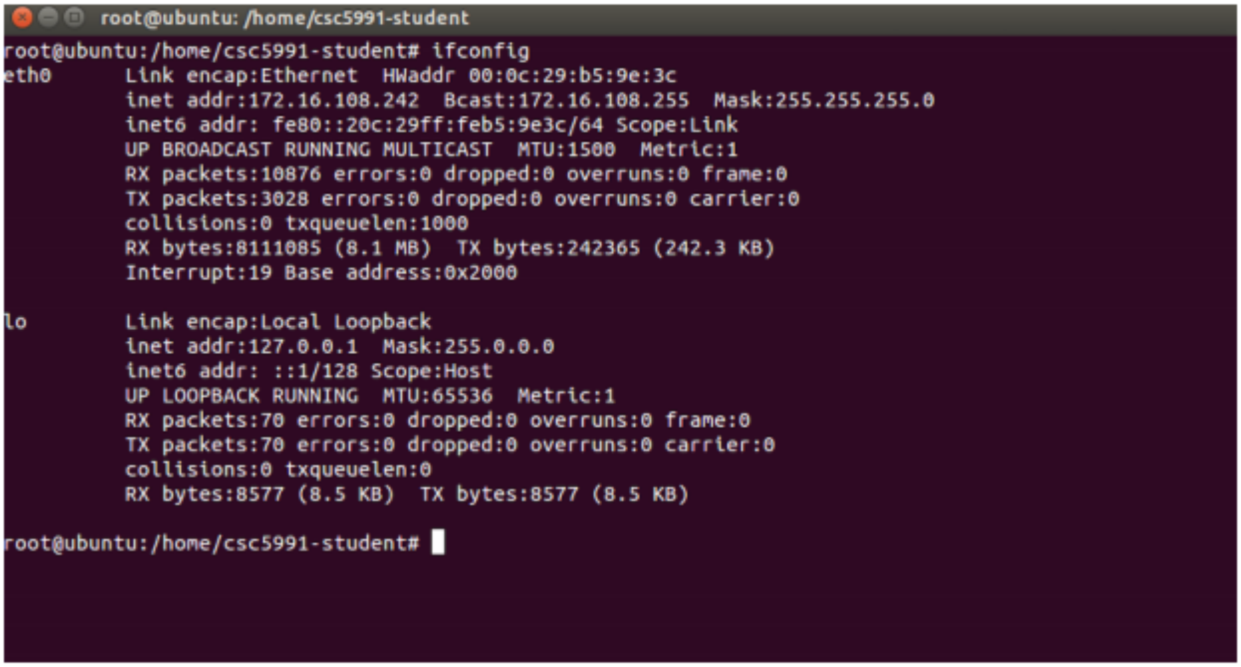
```
root@ubuntu:/home/csc5991-student# service snort restart
 * Stopping Network Intrusion Detection System  snort                   [ OK ]
 * Starting Network Intrusion Detection System  snort                   [ OK ]
root@ubuntu:/home/csc5991-student# 
```

Triggering an Alert for the New Rule

To trigger an alert for the new rule, you only need to send an ICMP message to the VM image where snort runs. First, you need to find the IP address of the VM by typing the following command.

$ ifconfig

For instance, the screenshot shows the execution result on my VM image, and the IP address is 172.16.108.242.



```
root@ubuntu:/home/csc5991-student# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b5:9e:3c
          inet addr:172.16.108.242  Bcast:172.16.108.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb5:9e3c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10876 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3028 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8111085 (8.1 MB)  TX bytes:242365 (242.3 KB)
          Interrupt:19 Base address:0x2000

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:65536  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8577 (8.5 KB)  TX bytes:8577 (8.5 KB)

root@ubuntu:/home/csc5991-student# █
```

Next, you can open a terminal in your host. If you host is a Windows OS, you can use one of the following two ways to open a terminal

1. Press "Win-R," type "cmd" and press "Enter" to open a Command Prompt session using just your keyboard.

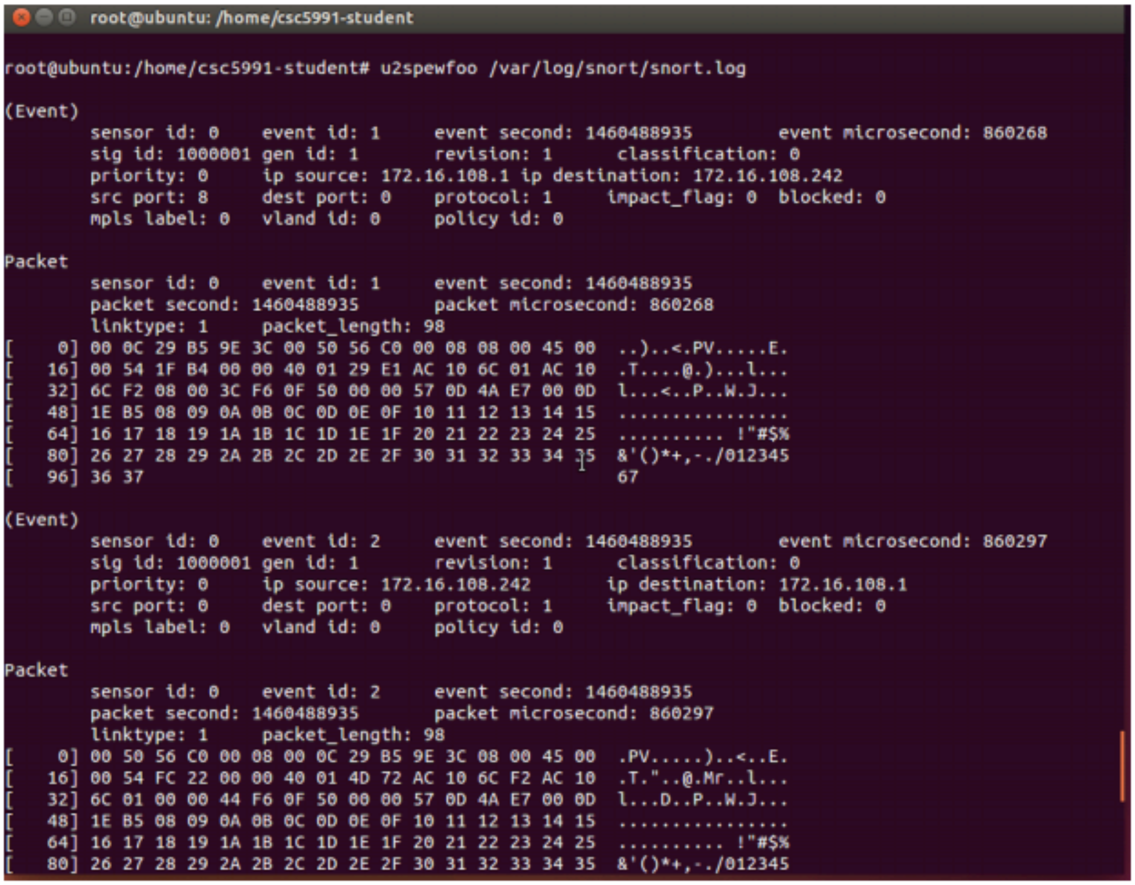
2. Click the "Start | Program Files | Accessories | Command Prompt" to open a Command Prompt session using just your mouse.

After you have a terminal, you can just type the following command to send ping messages to the VM.

$ ping 172.16.108.242

After you send the ping messages, the alerts should be trigged and you can find the log messages in /var/log/snort/snort.log. However, the snort.log file will be binary format.

You need to use a tool, called u2spewfoo, to read it. The screenshot below shows the result of reading the snort alerts.



```
root@ubuntu:/home/csc5991-student# u2spewfoo /var/log/snort/snort.log

(Event)
        sensor id: 0    event id: 1     event second: 1460488935        event microsecond: 860268
        sig id: 1000001 gen id: 1       revision: 1      classification: 0
        priority: 0     ip source: 172.16.108.1 ip destination: 172.16.108.242
        src port: 8     dest port: 0    protocol: 1     impact_flag: 0  blocked: 0
        mpls label: 0   vland id: 0     policy id: 0

Packet
        sensor id: 0    event id: 1     event second: 1460488935
        packet second: 1460488935       packet microsecond: 860268
        linktype: 1     packet_length: 98
[    0] 00 0C 29 B5 9E 3C 00 50 56 C0 00 08 08 00 45 00  ..)..<.PV.....E.
[   16] 00 54 1F B4 00 00 40 01 29 E1 AC 10 6C 01 AC 10  .T....@.)...l...
[   32] 6C F2 08 00 3C F6 0F 50 00 00 57 0D 4A E7 00 0D  l...<..P..W.J...
[   48] 1E B5 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15  ................
[   64] 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25  .......... !"#$%
[   80] 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35  &'()*+,-./012345
[   96] 36 37                                            67

(Event)
        sensor id: 0    event id: 2     event second: 1460488935        event microsecond: 860297
        sig id: 1000001 gen id: 1       revision: 1      classification: 0
        priority: 0     ip source: 172.16.108.242       ip destination: 172.16.108.1
        src port: 0     dest port: 0    protocol: 1     impact_flag: 0  blocked: 0
        mpls label: 0   vland id: 0     policy id: 0

Packet
        sensor id: 0    event id: 2     event second: 1460488935
        packet second: 1460488935       packet microsecond: 860297
        linktype: 1     packet_length: 98
[    0] 00 50 56 C0 00 08 00 0C 29 B5 9E 3C 08 00 45 00  .PV.....)..<..E.
[   16] 00 54 FC 22 00 00 40 01 4D 72 AC 10 6C F2 AC 10  .T."..@.Mr..l...
[   32] 6C 01 00 00 44 F6 0F 50 00 00 57 0D 4A E7 00 0D  l...D..P..W.J...
[   48] 1E B5 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15  ................
[   64] 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25  .......... !"#$%
[   80] 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35  &'()*+,-./012345
```

You can see that the SID is 1000001, and the alerts are generated by the ICMP messages.

HOMEWORK

1. Read the lab instructions above and finish all the tasks.

2. Answer the questions in the Introduction section, and justify your answers.

    a. What is a zero-day attack?

    b. Can Snort catch zero-day network attacks? If not, why not? If yes, how?

3. What are a network intrusion detection system (NIDS) and host intrusion detection system (HIDS)?

4. How are intrusions detected?

5. What is an advantage of anomaly detection?

6. How does a NIDS match signatures with incoming traffic?