Snort Lab CSC 433

Using two computers on the same network that also can see the internet, set up Snort on one of these computers (SnortPC) with the full set of rules from the Module. Perform the following two (2)exercises. (Note: Some individuals have had trouble doing this on wireless networks so use a wired network if possible).

1. On SnortPC run snort with the full rules and then on the second computer run a full nmap scan (-v -A) against the computer running snort. Capture all alerts generated by this nmap scan and identify the alerts that were generated by this nmap scan. In your paper show a couple of the alert texts.

2.On snortPC run snort in the sniffer mode (no rules folder designated in the command and -v switch verbose to print to screen) or as an alternative you could run Wireshark for a period sufficient to see some of the common benign packets that could be expected over a short period of time arriving at this computer (you could also simply surf the web to generate benign packets). After capturing this traffic for a short period of time, choose two unique packets you saw with the sniffer and craft two snort rules that would trigger alerts when these benign packets hit the snort computer. Be specific in your rules and avoid using generic rules such as "any" and "!" if possible. In these rules that you craft, insert your own name and a description in the title (msg option) of the alert so it will appear in the text file when alert is tripped. You can now either simply insert these rules at the top of the full rule-set or, preferably, substitute your very small rule set for the long rule set. Run this snort configuration for a time sufficient enough to catch several instances of these packets that would also be recorded in your alert.ids file.

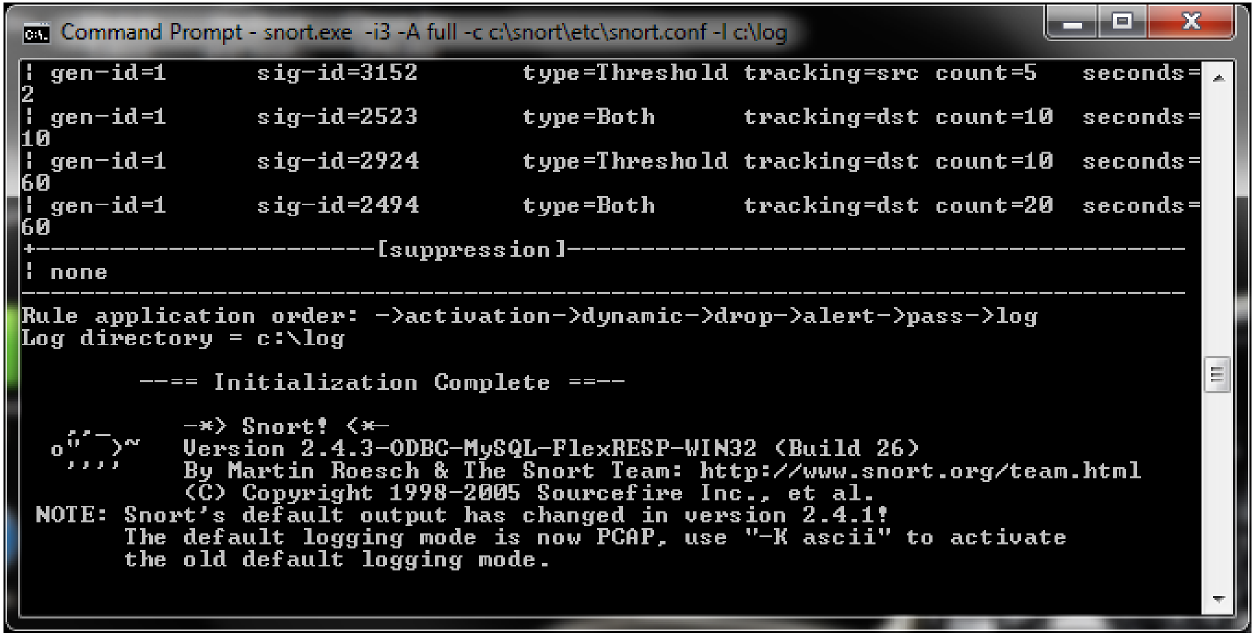
Write up these exercises explaining all your steps and presenting screen prints of your command line inputs and outputs and your alert.ids files and your logic for your rule set. This report will be a single word document and typically be approximately 7 to 8 pages including your screen prints and text files. Submit this report within this assignment (not via email).

SNORT Project Anonymized Example
CSC 433 Final Project

The first part of the final project was completed using two computers connected by their Ethernet ports on a switched environment. The first computer (PC1) was running Snort with a Windows OS, while the second computer (PC2) was running NMAP with a Mac OS.

Before scanning could occur Snort was configured using the Snort Installation Tutorial by Ryan M Rigg. Snort then had to be activated with the following commands. Notice the last command of c:\log, which tells Snort where to store the alert.ids file. This file location was created specifically for easy access of the log files.

Snort was verified working correctly when this message was received:

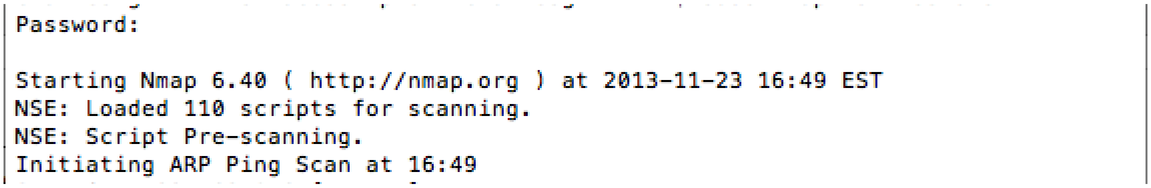

Command Prompt - snort.exe -i3 -A full -c c:\snort\etc\snort.conf -l c:\log

```
| gen-id=1      sig-id=3152       type=Threshold tracking=src count=5    seconds=
2
| gen-id=1      sig-id=2523       type=Both      tracking=dst count=10   seconds=
10
| gen-id=1      sig-id=2924       type=Threshold tracking=dst count=10   seconds=
60
| gen-id=1      sig-id=2494       type=Both      tracking=dst count=20   seconds=
60
+-----------------------[suppression]------------------------------------------
| none
-------------------------------------------------------------------------------
Rule application order: ->activation->dynamic->drop->alert->pass->log
Log directory = c:\log

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.4.3-ODBC-MySQL-FlexRESP-WIN32 (Build 26)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
           (C) Copyright 1998-2005 Sourcefire Inc., et al.
 NOTE: Snort's default output has changed in version 2.4.1!
       The default logging mode is now PCAP, use "-K ascii" to activate
       the old default logging mode.
```

It was now time to perform a full scan of services. From the Terminal application on PC2, NMAP was used to scan PC1. The command prompt input plus the first few lines of output from NMAP are shown below:

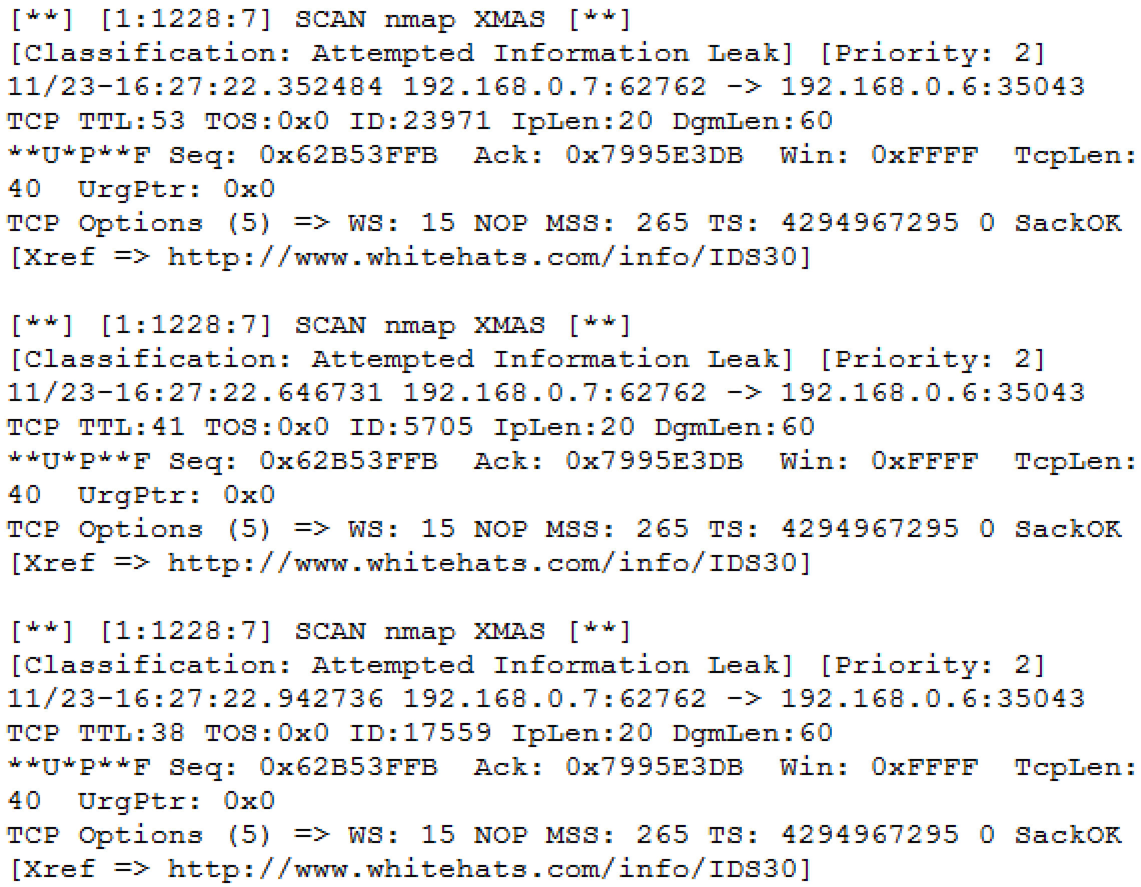


```
Password:

Starting Nmap 6.40 ( http://nmap.org ) at 2013-11-23 16:49 EST
NSE: Loaded 110 scripts for scanning.
NSE: Script Pre-scanning.
Initiating ARP Ping Scan at 16:49
```

Several types of traffic caused Snort to send output to the alert file. This traffic included XMAS, SNMP AgentX/TCP request, and SNMP TCP request packets. Several screen shots from the alert.ids file are

shown below.



```
[**] [1:1228:7] SCAN nmap XMAS [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/23-16:27:22.352484 192.168.0.7:62762 -> 192.168.0.6:35043
TCP TTL:53 TOS:0x0 ID:23971 IpLen:20 DgmLen:60
**U*P**F Seq: 0x62B53FFB  Ack: 0x7995E3DB  Win: 0xFFFF  TcpLen:
40  UrgPtr: 0x0
TCP Options (5) => WS: 15 NOP MSS: 265 TS: 4294967295 0 SackOK
[Xref => http://www.whitehats.com/info/IDS30]

[**] [1:1228:7] SCAN nmap XMAS [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/23-16:27:22.646731 192.168.0.7:62762 -> 192.168.0.6:35043
TCP TTL:41 TOS:0x0 ID:5705 IpLen:20 DgmLen:60
**U*P**F Seq: 0x62B53FFB  Ack: 0x7995E3DB  Win: 0xFFFF  TcpLen:
40  UrgPtr: 0x0
TCP Options (5) => WS: 15 NOP MSS: 265 TS: 4294967295 0 SackOK
[Xref => http://www.whitehats.com/info/IDS30]

[**] [1:1228:7] SCAN nmap XMAS [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/23-16:27:22.942736 192.168.0.7:62762 -> 192.168.0.6:35043
TCP TTL:38 TOS:0x0 ID:17559 IpLen:20 DgmLen:60
**U*P**F Seq: 0x62B53FFB  Ack: 0x7995E3DB  Win: 0xFFFF  TcpLen:
40  UrgPtr: 0x0
TCP Options (5) => WS: 15 NOP MSS: 265 TS: 4294967295 0 SackOK
[Xref => http://www.whitehats.com/info/IDS30]
```

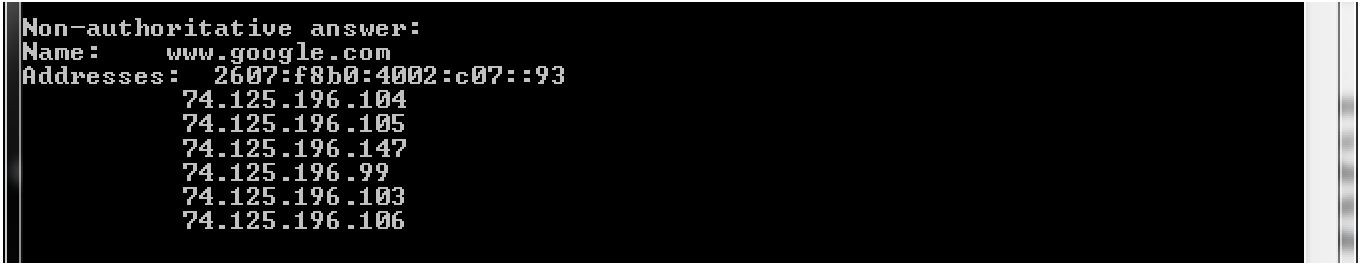
```
[**] [1:1421:11] SNMP AgentX/tcp request [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/23-16:25:49.771164 192.168.0.7:55868 -> 192.168.0.6:705
TCP TTL:57 TOS:0x0 ID:39103 IpLen:20 DgmLen:44
******S* Seq: 0x1045F155  Ack: 0x0  Win: 0x400  TcpLen: 24
TCP Options (1) => MSS: 1460
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-
0013][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=
2002-0012][Xref => http://www.securityfocus.com/bid/4132][Xref
=> http://www.securityfocus.com/bid/4089][Xref =>
http://www.securityfocus.com/bid/4088]

[**] [1:1421:11] SNMP AgentX/tcp request [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/23-16:25:49.882893 192.168.0.7:55869 -> 192.168.0.6:705
TCP TTL:53 TOS:0x0 ID:10889 IpLen:20 DgmLen:44
******S* Seq: 0x1044F154  Ack: 0x0  Win: 0x400  TcpLen: 24
TCP Options (1) => MSS: 1460
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-
0013][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=
2002-0012][Xref => http://www.securityfocus.com/bid/4132][Xref
=> http://www.securityfocus.com/bid/4089][Xref =>
http://www.securityfocus.com/bid/4088]


[**] [1:1418:11] SNMP request tcp [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/23-16:25:48.229007 192.168.0.7:55868 -> 192.168.0.6:161
TCP TTL:59 TOS:0x0 ID:32285 IpLen:20 DgmLen:44
******S* Seq: 0x1045F155  Ack: 0x0  Win: 0x400  TcpLen: 24
TCP Options (1) => MSS: 1460
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-
0013][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=
2002-0012][Xref => http://www.securityfocus.com/bid/4132][Xref
=> http://www.securityfocus.com/bid/4089][Xref =>
http://www.securityfocus.com/bid/4088]

[**] [1:1418:11] SNMP request tcp [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/23-16:25:48.338358 192.168.0.7:55869 -> 192.168.0.6:161
TCP TTL:59 TOS:0x0 ID:2427 IpLen:20 DgmLen:44
******S* Seq: 0x1044F154  Ack: 0x0  Win: 0x400  TcpLen: 24
TCP Options (1) => MSS: 1460
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-
0013][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=
2002-0012][Xref => http://www.securityfocus.com/bid/4132][Xref
=> http://www.securityfocus.com/bid/4089][Xref =>
http://www.securityfocus.com/bid/4088]
```

The second part of the final project consisted of creating custom rules stored on the local.rules file for Snort to use. The first custom rule was for a UDP DNS response from the default DNS server 8.8.8.8 on port 53. The second custom rule was for a request from the user for one of the Google servers using TCP on port 443. Google has several different IP addresses listed that do not fall sequentially, as shown from an nslookup output below. So in order to prevent false-positives that would occur when using a network mask, several separate rules were created to account for different addresses.



```
Non-authoritative answer:
Name:    www.google.com
Addresses:  2607:f8b0:4002:c07::93
          74.125.196.104
          74.125.196.105
          74.125.196.147
          74.125.196.99
          74.125.196.103
          74.125.196.106
```

The custom rules:

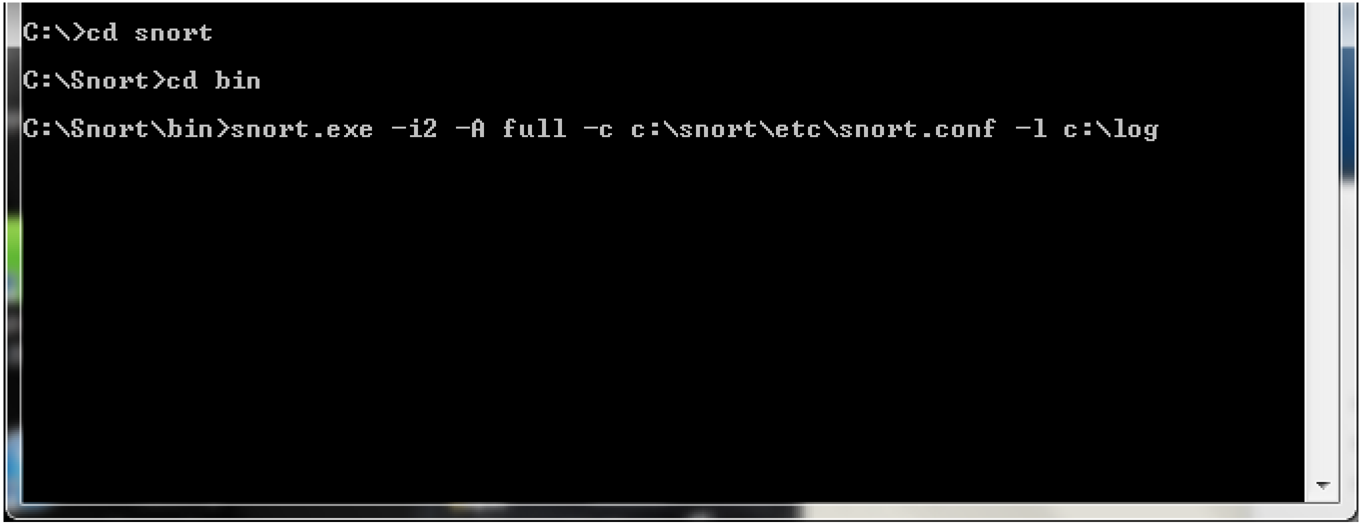
Before running Snort, the configuration file had to be edited so Snort only used the local.rules file when scanning packets. All other rule files where commented out using the # character. Below you can see that only the local.rules file does not have a # character before it.

```
include $RULE_PATH/local.rules
#include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/exploit.rules
#include $RULE_PATH/scan.rules
#include $RULE_PATH/finger.rules
#include $RULE_PATH/ftp.rules
#include $RULE_PATH/telnet.rules
#include $RULE_PATH/rpc.rules
#include $RULE_PATH/rservices.rules
#include $RULE_PATH/dos.rules
#include $RULE_PATH/ddos.rules
#include $RULE_PATH/dns.rules
#include $RULE_PATH/tftp.rules

#include $RULE_PATH/web-cgi.rules
#include $RULE_PATH/web-coldfusion.rules
#include $RULE_PATH/web-iis.rules
#include $RULE_PATH/web-frontpage.rules
#include $RULE_PATH/web-misc.rules
#include $RULE_PATH/web-client.rules
#include $RULE_PATH/web-php.rules

#include $RULE_PATH/sql.rules
#include $RULE_PATH/x11.rules
#include $RULE_PATH/icmp.rules
#include $RULE_PATH/netbios.rules
#include $RULE_PATH/misc.rules
#include $RULE_PATH/attack-responses.rules
#include $RULE_PATH/oracle.rules
```

After proper configuration, the commands below were executed to start Snort.



```
C:\>cd snort

C:\Snort>cd bin

C:\Snort\bin>snort.exe -i2 -A full -c c:\snort\etc\snort.conf -l c:\log
```

After starting Snort, a web browser was opened and several sites were visited including google.com. Enough sites were visited to ensure the browser had to complete several DNS lookups. Snort was then

stopped and the alert.ids file from the log folder was opened to check the results. As you can see from the output below, packets from the google.com request and the DNS responses were recorded.

Snort Advice for Lab

- Make sure WinPcap is installed on the machine
- Install snort to the C: root directory
- When you uncompress the snort rules file you will need to copy the Rules folder to the C:/snort directory where it will replace an empty rules folder. Also copy the "signature" folder (in the Doc folder)from uncompressed rules in to the Doc folder on the snort directory.
- Go to C:\snort\etc folder and find snort.conf file
- open in Wordpad (not notepad)
- All the # signs indicate comments and are not executed.

- You can leave the default home network and external network settings as "any" (or you can change them to your own home network (for example, 10.123.10.0/24))
- Change the line which reads var RULE_PATH .../rules to read var RULE_PATH c:\snort\rules
- Now you can tell the installation to put alerts in a file called alert.ids (can open with notepad) by setting the following in the output section of snort.conf. Find the line that begins #output log tcpdump:tcpdump.log and un-comment and change to: output alert_fast: alert.ids
- When you run snort you may need to include a switch to tell it which adapter to use. On my machine (#1 is modem) so the NIC would be "-i2"

- Run from command line changing to the snort\bin folder.
- cd \
- cd snort\bin
- To run with a sample command for sniffer set as: snort.exe -i2 -v
- To log the same traffic: snort.exe -i2 -l C:\snort\log (where -l (lower case L) indicates logging)
- Note: don't expect to read this file as it is binary not ascii
- To set as IDS with full rule set and full logging: snort.exe -i2 -A full -c C:\snort\etc\snort.conf -l C:\snort\log
- This will put your alerts in the C:\snort\log\alert.ids file which can be opened with Wordpad
- Note that all rules can be opened and edited with Wordpad.

When you write your special rule. Put it in the local.rules rule set and comment out (#) (or erase) all "Include rules" sets on snort.conf except "Include local.rules"

Another way to use your own rules is put them in a new rules file (for example local.rules) and then use the following command.

snort.exe -c C:\snort\rules\local.rules -l C:\snort\log

Just remember that this way you CANNOT use any of the variables defined by the snort.conf file such as the external_net and home_net variables.

Please make your snort rule non-trivial. That is, I want to see rule that has at least two conditions (e.g. specific port, specific text string, specific source IP, etc) that will select a unique packet that you have identified and are sure that you will see over a fixed period of time (perhaps by eliciting this packet?).