

## Lab 6: IDS 2 (Snort)

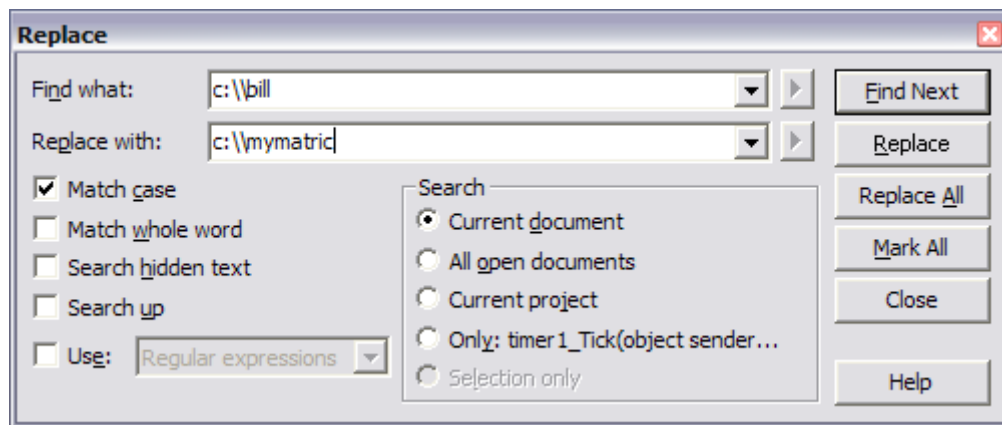
### Details

**Aim:** To use Snort to detect attacks

**Note:** To enhance the development, you can use the following program:

<http://www.dcs.napier.ac.uk/~bill/SnortAnalyser.zip>

Before you start... double click on the form, and reveal the code. Now select **Edit**, then **Find and Replace**, and then **Replace**. After this, change all the occurrences of c:\bill to c:\mymatric (where *mymatric* is your matriculation number), such as:



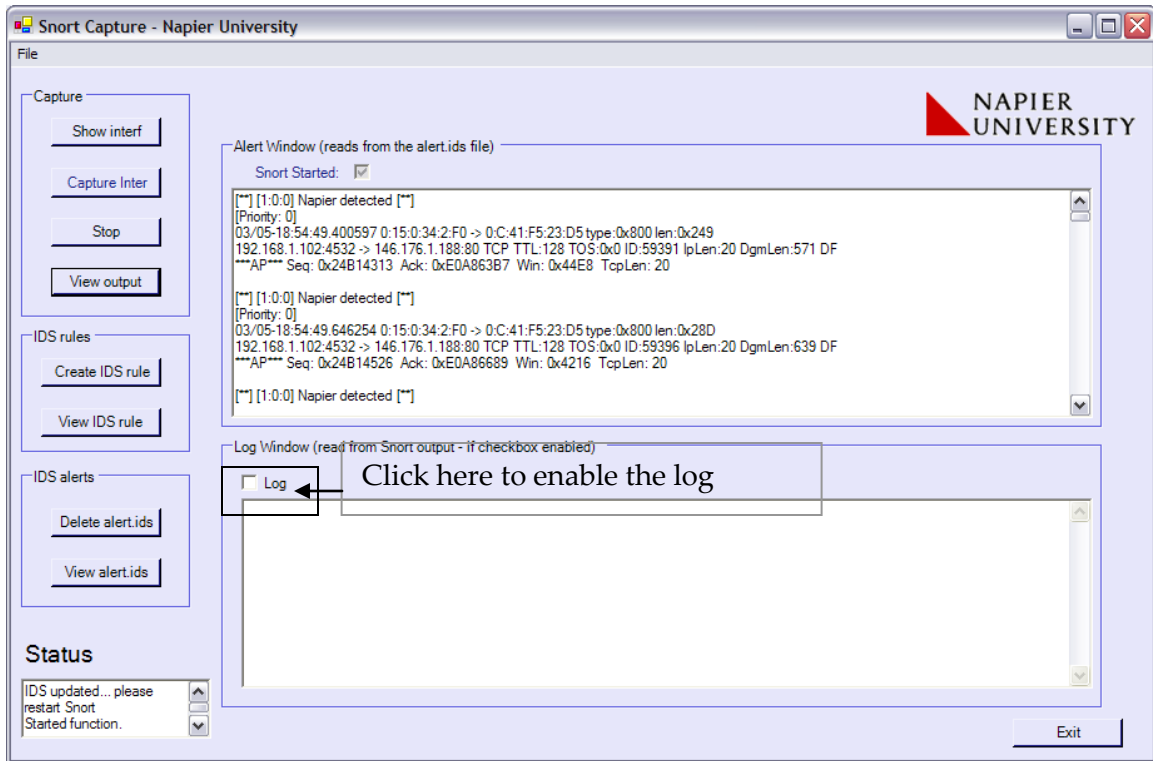
To update the rules, **double click** on the **Create IDS rule** button, and add the necessary rules. For example to add two rules:

```
string rule1,rule2;

rule1 = "alert tcp any any -> any 80 (content:\"napier\"; msg:\"Napier detected\");";
rule2 = "alert tcp any any -> any 80 (content:\"fred\"; msg:\"Napier detected\");";

StreamWriter SW;
SW=File.CreateText("c:\\snort\\bin\\napier.txt");
SW.WriteLine(rule1);
SW.WriteLine(rule2);
SW.Close();
```

Run the program, and verify that it detects the presence of the word “Napier” in the outgoing network traffic, such as:



## Activities

1. Write rules which will detect the word **Intel** in the payload, so that the alerts are:
  - A. Intel found on outgoing WWW traffic (port 80). Change it so that it detects Intel either in upper or lower case.
  - B. Intel found on incoming WWW traffic (port 80).

Verify your rules by running tests.

**What are the rules:**

2. Write a rule which detects the following:

- A. An incoming Web page with the words "John" and "Napier".

**What is the rule:**

**Show that it works with the site:** <http://www.johnnapier.com/>  
**and not with:** <http://www.napier.ac.uk>

3. Run the program, and click on the **Log** checkbox, and start Snort (with **Capture Inter**). Run Snort, and ping one or more hosts. From the Log window, scroll until you find your ping activity. From this locate the ARP and ping activity (see Appendix A for an example of the packets):

**What information does the sending ARP and also the receiving ARP packet have:**

**What are the contents of the ping packet:**

4. Run the program, and click on the **Log** checkbox, and start Snort (with **Capture Inter**). Run Snort, and access the main Web site of the University of Edinburgh ([www.ed.ac.uk](http://www.ed.ac.uk)). From the Log window, scroll until you find your DNS activity (see Appendix A for an example of the packets):

**What information does the sending DNS and also the receiving DNS packet have:**

**Which TCP port does the DNS server use:**

**From the contents of the DNS return, and using nslookup on [www.ed.ac.uk](http://www.ed.ac.uk), is it possible to determine the IP address that is returned from the DNS server (see Appendix A)? Yes/No**

5. A typical signature of a network attack is a port scan, where an intruder scans the open ports on a host. Using Netstat, determine your connected ports, and using netstat -a, determine the all your listening port.

**Some of the connected ports:**

**Some of the listening ports:**

6. A factor in security is to determine the TCP ports which are listening on hosts, as these can be one way that an intruder can gain access to a host. Also it is possible to detect an intruder if they are scanning a network. Thus, download the NMAP portscanner. Note: **DO NOT PORT SCAN ANY OTHER MACHINE THAN YOUR NEIGHBOURS COMPUTER.** An example is at:

<http://download.insecure.org/nmap/dist/nmap-3.95-win32.zip>

A sample run is:

```
> nmap 192.168.1.1
Starting Nmap 3.95 ( http://www.insecure.org/nmap ) at 2006-01-12 13:26 GMT Standard Time
Interesting ports on 192.168.1.1:
(The 1668 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
80/tcp    open  http
8080/tcp  open  http-proxy
MAC Address: 00:0C:41:F5:23:D5 (The Linksys Group)
Nmap finished: 1 IP address (1 host up) scanned in 2.969 seconds
```

For your host, and using NMAP, complete the following:

**Which ports are open:**

**Using the command netstat -a verify that these ports are open:**

7. Download the client and server program, and run the server on one machine and set its listening port to 1001. Rerun the port scanner from your neighbour's machine.

<http://www.dcs.napier.ac.uk/~bill/dotNetClientServer.zip>

**Does the port scanner detect the new server port: Yes/No**

8. Next with the server listing on port 1001. Now write a Snort rule which detects the incoming SYN flag for a connection from a client to the server.

**What is the Snort rule:**

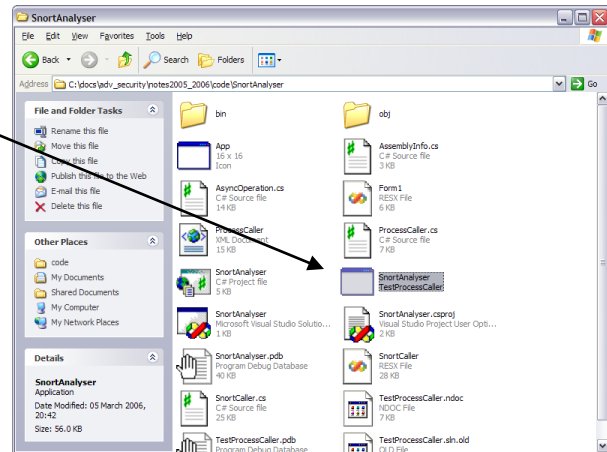
9. Write a rule for Snort which allows a port scan to be detected, and verify that it works:

**Snort rule:**

**Did it detect the port scan: Yes/no**

## Note

If you ever want to run the program as a stand-alone file, you will find the EXE in the solution folder, such as:



# Appendix A

**ARP.** An ARP packet has the format:

```
03/05-19:59:56.376568 ARP who-has 192.168.1.101 tell 192.168.1.102
03/05-19:59:56.378315 ARP reply 192.168.1.101 (0:C:41:38:9B:A4) is-at
0:60:B3:9F:CA:E1
```

**Ping (echo).** A ping packet has the following format:

```
03/05-19:59:56.378331 0:15:0:34:2:F0 -> 0:60:B3:9F:CA:E1 type:0x800 len:0x4A
192.168.1.102 -> 192.168.1.101 ICMP TTL:128 TOS:0x0 ID:2861 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:4096 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi
```

**Ping (echo-reply).** A ping packet has the following format:

```
03/05-19:59:56.379672 0:C:41:38:9B:A4 -> 0:15:0:34:2:F0 type:0x800 len:0x4A
192.168.1.101 -> 192.168.1.102 ICMP TTL:128 TOS:0x0 ID:21803 IpLen:20 DgmLen:60
Type:0 Code:0 ID:512 Seq:4096 ECHO REPLY
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi
```

**DNS (request).** A DNS request packet has the following format:

```
03/05-20:21:33.008948 0:15:0:34:2:F0 -> 0:C:41:F5:23:D5 type:0x800 len:0x48
192.168.1.102:1082 -> 195.92.195.94:53 UDP TTL:128 TOS:0x0 ID:3318 IpLen:20
DgmLen:58
Len: 30
80 07 01 00 00 01 00 00 00 00 00 00 03 77 77 77 .....www
02 68 77 02 61 63 02 75 6B 00 00 01 00 01 .....hw.ac.uk.....
```

ping payload

**DNS (reply).** A DNS reply packet has the following format:

```
03/05-20:21:33.034234 0:C:41:F5:23:D5 -> 0:15:0:34:2:F0 type:0x800 len:0xF6
195.92.195.94:53 -> 192.168.1.102:1082 UDP TTL:62 TOS:0x0 ID:0 IpLen:20
DgmLen:232 DF
Len: 204
80 07 81 80 00 01 00 01 00 04 00 04 03 77 77 77 .....www
02 68 77 02 61 63 02 75 6B 00 00 01 00 01 C0 0C .....hw.ac.uk.....
00 01 00 01 00 00 B4 36 00 04 89 C3 96 32 C0 10 .....6.....2..
00 02 00 01 00 00 B4 36 00 0C 03 6E 73 32 02 6A .....6...ns2.j
61 03 6E 65 74 00 C0 10 00 02 00 01 00 00 B4 36 a.net.....6
00 0A 07 6E 65 6D 65 73 69 73 C0 10 C0 10 00 02 ...nemesis.....
00 01 00 00 B4 36 00 0C 09 6E 61 6D 65 73 65 72 .....6...nameser
76 65 C0 10 C0 10 00 02 00 01 00 00 B4 36 00 0C ve.....6..
09 6E 65 74 73 65 72 76 65 31 C0 10 C0 3A 00 01 .netserver1.....
00 01 00 00 D3 24 00 04 C1 3F 69 11 C0 52 00 01 .....$...?i..R..
00 01 00 00 B4 36 00 04 89 C3 97 6E C0 68 00 01 .....6.....n.h..
00 01 00 01 16 D9 00 04 89 C3 97 69 C0 80 00 01 .....i....
00 01 00 00 B4 36 00 04 89 C3 96 3D .....6.....=
```

DNS Server port