Lab 5: Invoking Snort

Details

Aim: To provide a foundation in invoking and controlling Snort

Activities

1. If Visual Studio is installed on your machine, download the following solution [1]:

http://www.dcs.napier.ac.uk/~bill/SnortCaller.zip

An outline of the code is:

```
public void runShort(string arguments)
{
  processCaller = new ProcessCaller(this);
  processCaller.FileName = @"c:\snort\bin\snort.exe";
  processCaller.Arguments = arguments;
  processCaller.StdErrReceived += new DataReceivedHandler(writeStreamInfo);
  processCaller.StdOutReceived += new DataReceivedHandler(writeStreamInfo);
  processCaller.Completed += new EventHandler(processCompletedOrCanceled);
  processCaller.Cancelled += new EventHandler(processCompletedOrCanceled);

  this.richTextBox1.Text = "Started function.  Please stand by.."
              + Environment.NewLine;

  processCaller.Start();
}
private void btnInterface_Click(object sender, System.E
{

    this.runShort("-W");
}
```

This defines the Snort arguments that are used to run the program.

2. In the Project listing, **double click** on the SnortCaller.cs file, then **double click** on the **Show interf** button, and add the following highlighted code:

```
private void btnInterface_Click(object sender, System.EventArgs e)
{
   this.runShort("-W");
}
```

3. Run the program, and show that the output is similar to the output in Figure 1:

What is/are your interface(s)?

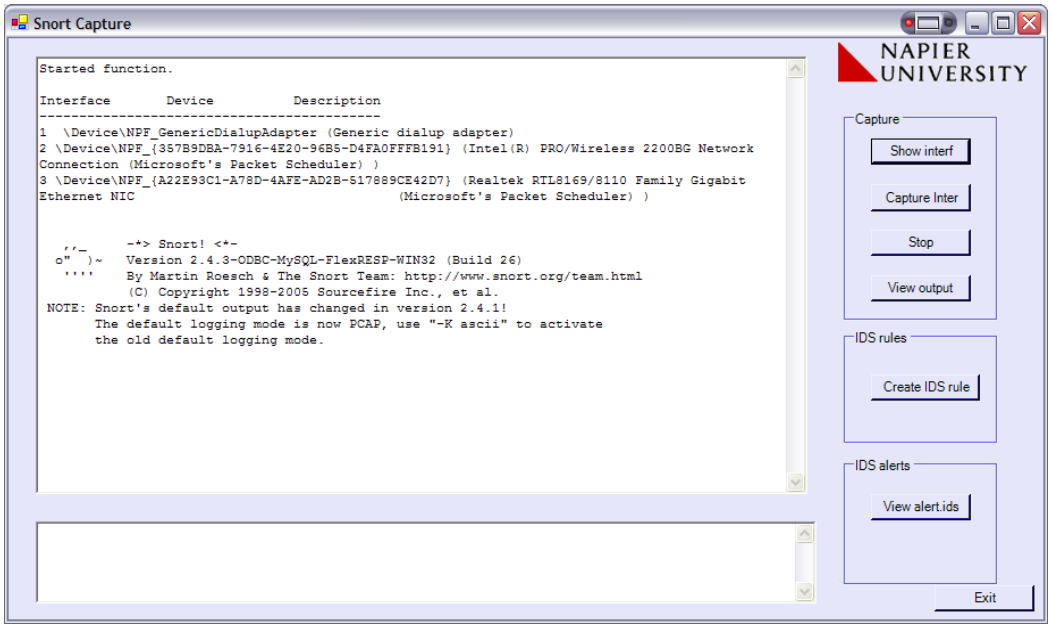
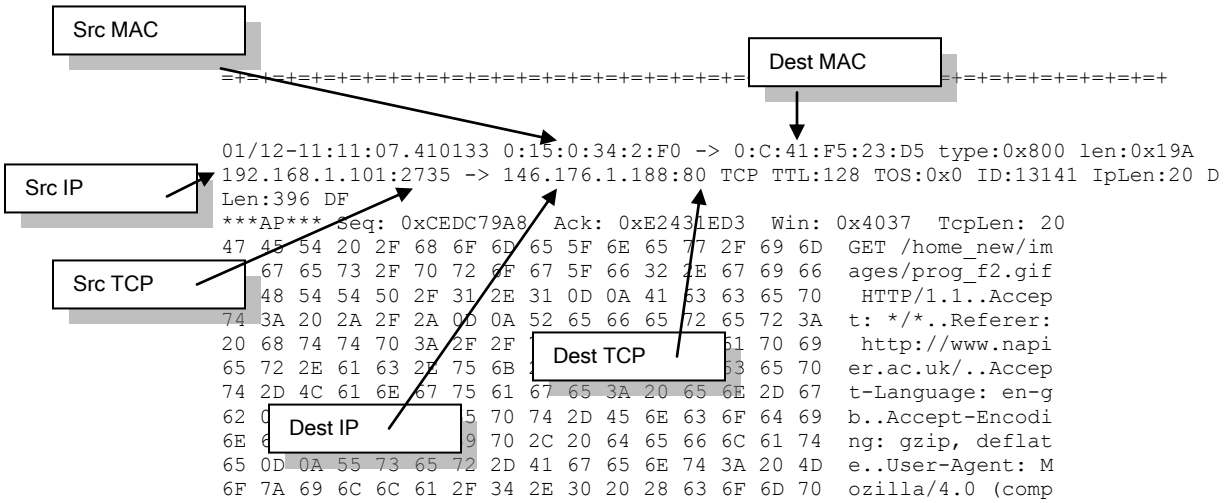


Figure 1:

4. **Double click** on the **Capture Inter** button, and add the following highlighted code. Replace the c:\\bill with c:*yourMatricNo*, and replace the value after the –i option with the interface number. This should log to the folder defined.

```
private void btnStart_Click(object sender, System.EventArgs e)
{
  if (!Directory.Exists("c:\\bill")) Directory.CreateDirectory("c:\\bill");
  this.runShort("-dev -i 1 -p -l c:\\bill -K ascii");
}
```

5. Run the program and get Snort to capture the packets, and then stop it with the **Stop** button (Figure 2). Generate some Web traffic, and view the output, and verify that it is capturing data packets, such as:



6. Select one of the TCP data packets, and determine the following:

The source IP address:

The source TCP port:

The destination IP address:

The destination TCP port:

The source MAC address:

The destination MAC address:

The TCP flags:

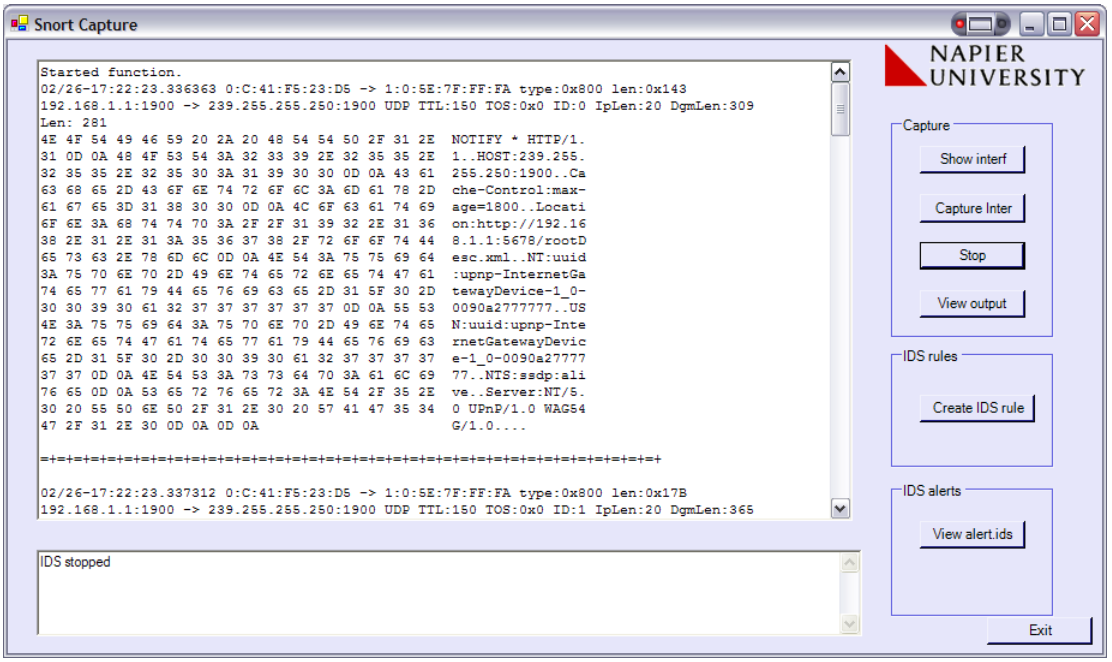


Figure 2:

7. **Double click** on the **View Output** button, and add the following highlighted code. Replace the c:\\bill with c:*yourMatricNo*.

```
private void btnView_Click(object sender, System.EventArgs e)
```

```
{
    openFileDialog1.InitialDirectory="c:\\bill";
    openFileDialog1.ShowDialog();
    Process.Start("wordpad.exe", openFileDialog1.FileName);
}
```

8. Run the program, and select the **View Output** button, and verify that you get the output seen in Figure 3, and open one of the IDS files in the subfolders, and verify the output, as shown in Figure 4.

What are the contents of the folder:

Go into one of the folders and view the contents of the IDS file. What does it contain:

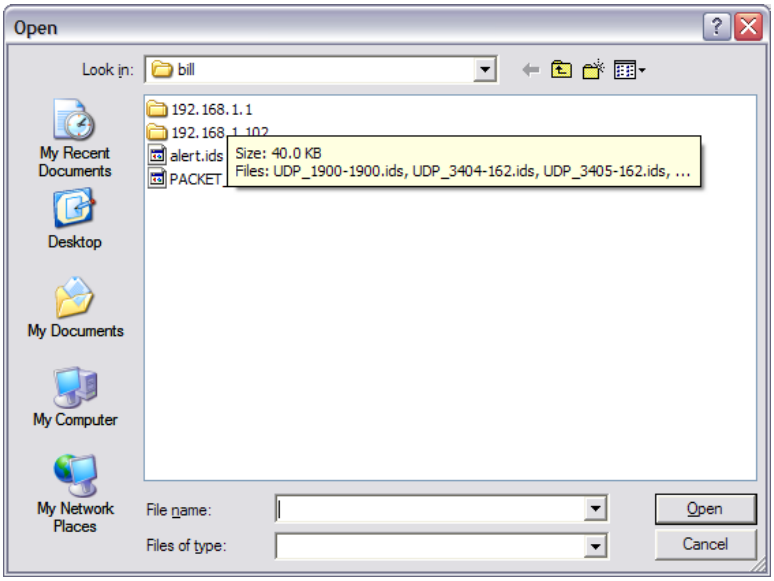


Figure 3:

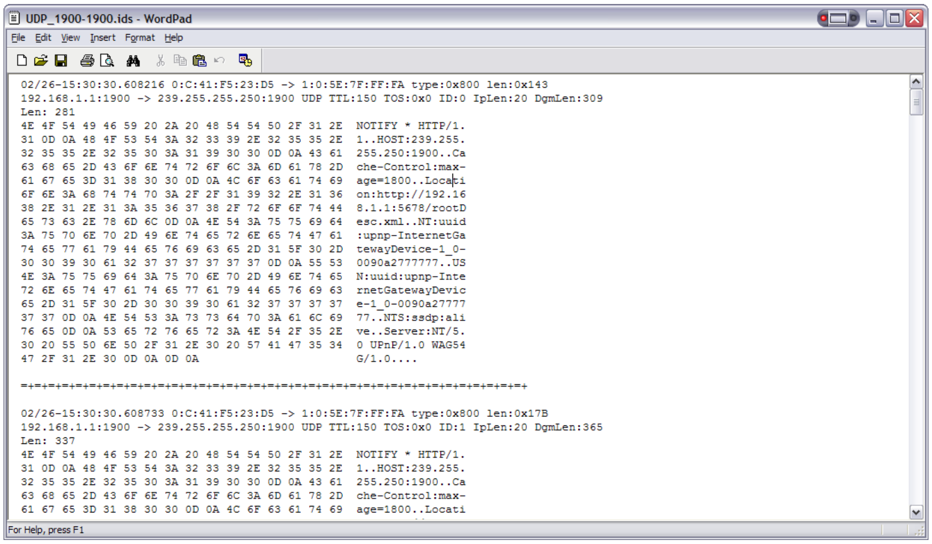


Figure 4:

9. **Double click** on the **Create IDS rule** button, and add the following code:

```
private void btnIDSRule_Click(object sender, System.EventArgs e)
{
   string rule;

rule = "alert tcp any any -> any 80 (content:\"napier\"; msg:\"Napier detected\";)";
   StreamWriter SW;
   SW=File.CreateText("c:\\snort\\bin\\napier.txt");
   SW.WriteLine(rule);
   SW.Close();
   statusIDS.Text+="IDS updated... please restart Snort";
}
```

which writes a Snort rule to the napier.txt file.

10. **Double click** on the **View alert.ids** button, and add the following code (remember to replace the c:\\bill with c:*yourMatricNo*):

```
private void btnViewAlert_Click(object sender, System.EventArgs e)
{
  if (File.Exists("c:\\bill\\alert.ids"))
  {
        Process.Start("wordpad.exe", "c:\\bill\\alert.ids");
  }
  else statusIDS.Text+="File does not exist...";
}
```

also update the line:

```
this.runShort("-dev -i 1 -p -l c:\\bill -K ascii");
```

with (to allow Snort to read-in the newly created rules file):

```
this.runShort("-dev -i 1 -p -l c:\\bill -K ascii –c c:\\snort\\bin\\napier.txt");
```

11. Run the program, and capture some Web traffic with the name **napier** in it. Then **Stop** the capture, and select the **View alert.ids** button (Figure 5).

What are the contents of the alert.ids file: **Did it detect "napier":**

12. Next download the client and server programs from:

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

13. In groups of two, one person should run the server on their computer, and the other person runs the client, and connects to the server on port **1001**. Make sure that you can chat, before going onto the next part of the tutorial (Figure 6).

14. Write a Snort rule which detects the word "napier" in the communications between the client and server.

What is the Snort rule for this:

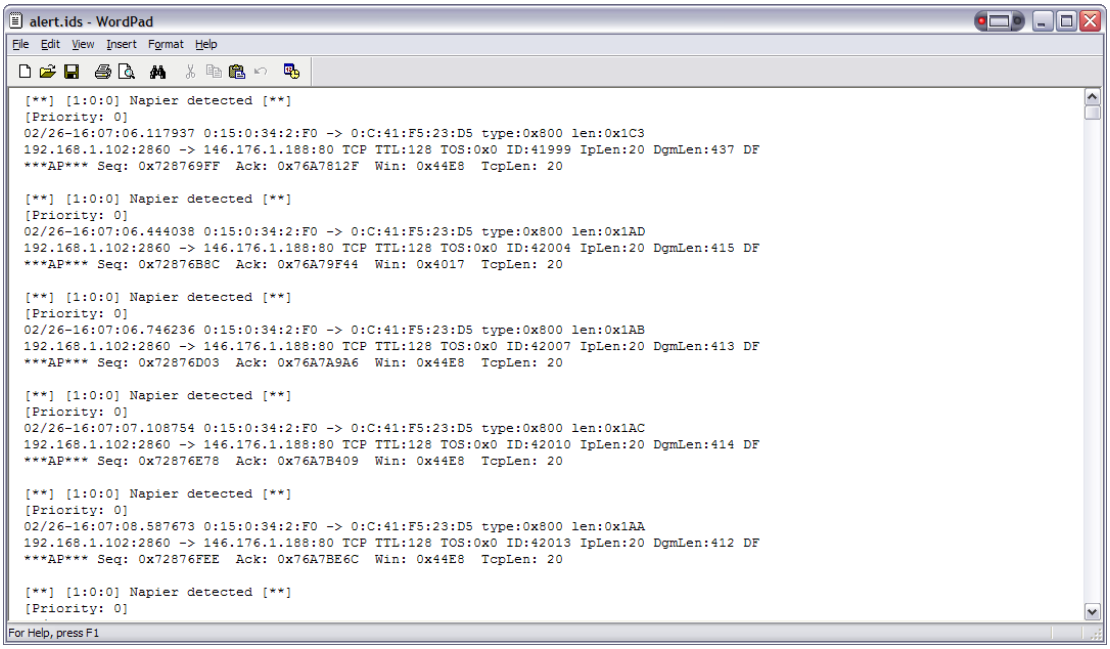


Figure 5:

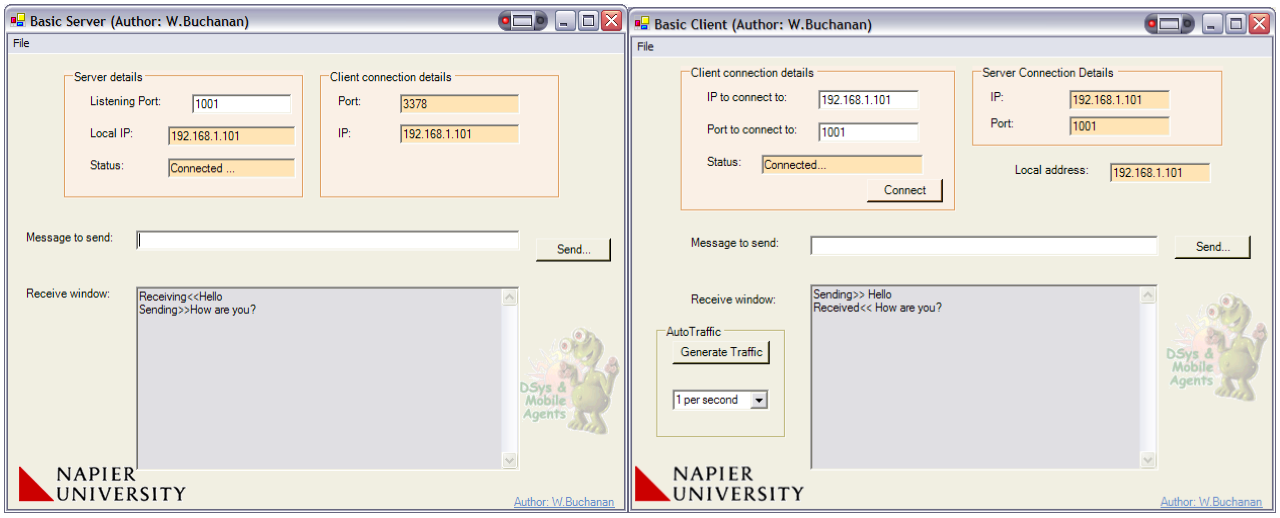


Figure 6:

Note: If you want the complete solution at any time, use:

↳http://www.dcs.napier.ac.uk/~bill/SnortCallerComplete.zip

[1] Code is based on http://www.codeproject.com/csharp/LaunchProcess.asp.