

# Labs



Contact: Prof Bill Buchanan  
Email: [w.buchanan@napier.ac.uk](mailto:w.buchanan@napier.ac.uk)  
Room: C.63

Week	Date	Teaching	Attended
2	18/11/10	Lab 1: Windows Services/Toolkit 1	
<p><b>Aim:</b> The aim of this lab is to investigate the discovery and configuration of services within Windows. It uses the Windows 2003 VM image (WINDOWS2003).</p> <p><b>Time to complete:</b> 4/5 hours (Two supervised hours in B.56, and two/three additional hours, unsupervised).</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Complete Lab 1: Windows Services/Toolkit.</li> </ul> <p><b>Learning activities:</b> At the end of these activities, you should understand:</p> <ul style="list-style-type: none"> <li>• How to define services in Windows.</li> <li>• How to call-up configuration commands from a toolkit.</li> </ul> <p><b>Reflective statements (end-of-exercise):</b> How does the VM image setup itself up so that it can access the Internet, and that the local host can access the services within it?</p> <p>What are the key Windows commands used to discover the services which are being run?</p> <p>What are the key folder locations for Windows services?</p> <p><b>Source code used:</b></p> <p><a href="http://buchananweb.co.uk/toolkit.zip">http://buchananweb.co.uk/toolkit.zip</a></p>			

# 1 Lab 1: Windows Services/Toolkit


## 1.1 Details

---

**Aim:** To provide a foundation in setup and consuming Windows services, and to start creating a toolkit which will be built-up over the next series of labs.


## 1.2 Windows Services

---

 On-line demo:  
[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/threat01/threat01.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/threat01/threat01.htm)

**PART A.** This part of the lab has two elements: the host machine (DESKTOP) and the Windows virtual image (WINDOWS2003).


**L1.1** Run the Windows Server 2003 virtual image (User name: Administrator, Password: napier). Within the virtual image, run the command prompt and determine its IP address using IPCONFIG.

 What are the IP addresses of the server and the network address which will be used to connect to the virtual image:


**L1.2** From DESKTOP, ping WINDOWS2003, and vice-versa.

### **SERVICE: Web**

**L1.3** In WINDOWS2003, go to the folder c:\inetpub\wwwroot.

 What are the names of the files in this folder:

**L1.4** From WINDOW2003, run netstat -a, and determine the services that are running.

 List some of the services:

**L1.5** From your host, connect to the Web Server from DESKTOP using <http://w.x.y.z>, where w.x.y.z is the IP address of WINDOWS2003 (Figure L1.1). Repeat this using:

telnet w.x.y.z 80

and then enter:

GET /iisstart.htm

☞ What is the result, and how does it relate to accessing the home page of the Web server?

**L1.6** On WINDOWS2003, using Microsoft Web Developer Express, open up the c:\inetpub\wwwroot Web folder, and Add a New Item to create your own home page. Next modify iisstart.htm so that it has a link to your home page. The home page should be:

## My Home Page

This is a sample ASP.NET page. Click [\[here\]](#) to return to the default home file.

☞ Can you access this page from the host (DESKTOP)?

☞ On WINDOWS2003, go to C:\WINDOWS\system32\LogFiles\W3SVC1. What is the contents of the folder, and what do the files contain?

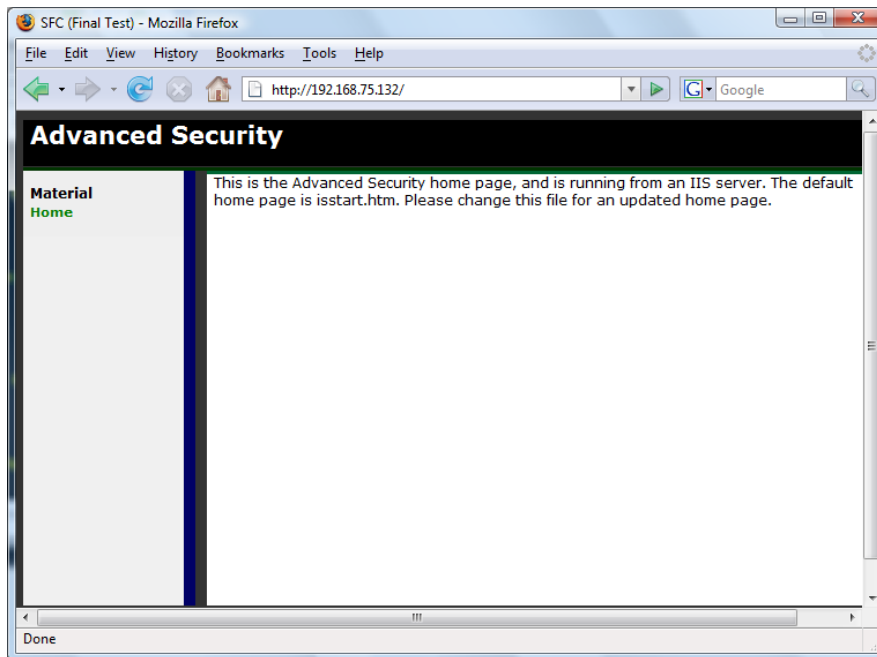
☞ How might these log files be used to trace malicious activity?

### **SERVICE: Telnet**

**L1.7** From your host, connect to the Telnet Server from DESKTOP using telnet x.y.y.z, where w.x.y.z is the IP address of WINDOWS2003. Login in with Administrator (password: Napier).

☞ What is the default home folder for Telnet on WINDOWS2003:

Quit from Telnet, using the “exit” command.



**Figure L1.1** HTTP connection

**SERVICE: FTP**

**L1.8** From your host, connect to the FTP Server from DESKTOP using ftp://w.x.y.z where w.x.y.z is the IP address of WINDOWS2003 (Figure L1.2). Repeat this using:

```
telnet w.x.y.z 21
```

and then enter the commands in bold (and note the commands that you get beside the sample return ones):

```
220 Microsoft FTP Service
USER Administrator
331 Password required for Administrator.
PASS napier
230 User Administrator logged in.
SYST
215 Windows_NT
TYPE I
200 Type set to I.
PASV
227 Entering Passive Mode (192,168,75,132,4,65).
LIST
```

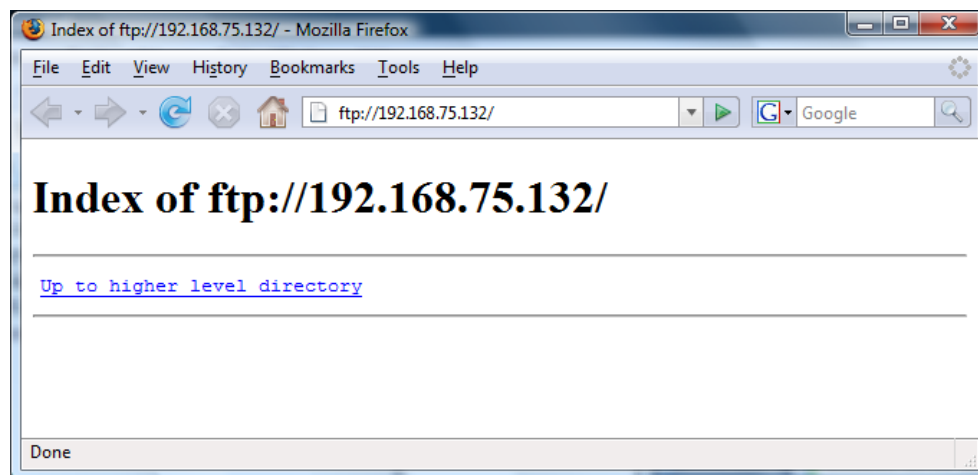
Now FTP opens up a port for the data transfer. This is calculated from the last two digits of the Passive Mode response (227 response). It is calculated as four times the second last digital plus the last digital. So, in this case, it is:

Port =  $4 \times 256 + 65 = 1089$

Next open up the data transfer by creating a new Telnet connection, such as:

```
telnet w.x.y.z 1089
```

- ☞ Can you access this page from the host?
- ☞ On WINDOWS2003, go to C:\WINDOWS\system32\LogFiles\MSFTPSVC1. What is the contents of the folder, and what do the files contain?
- ☞ How might these log files be used to trace malicious activity?



**Figure L1.2** FTP connection

### **SERVICE: SMTP**

**L1.1** From your host, use the following command:

```
telnet w.x.y.z 25
```

and connect to the SMTP server. Next enter the commands in bold:

```
220 napier Microsoft ESMTP MAIL Service, Version: 6.0.3790.3959 ready at Sun,
0 Dec 2009 21:56:01 +0000
help
214-This server supports the following commands:
214 HELO EHLO STARTTLS RCPT DATA RSET MAIL QUIT HELP AUTH TURN
    ETRN BDAT VRFY
```

```
helo me
250 napier Hello [192.168.75.1]
mail from: email@domain.com
250 2.1.0 email@domain.com....Sender OK
rcpt to: fred@mydomain.com
250 2.1.5 fred@mydomain.com
Data
354 Start mail input; end with <CRLF>.<CRLF>
From: Bob <bob@test.org>
To: Alice <alice@ test.org >
Date: Sun, 20 Dec 2009
Subject: Test message

Hello Alice.
This is an email to say hello
.
250 2.6.0 <NAPIERMp7lzvxrMVHFb00000001@napier> Queued mail for delivery
```

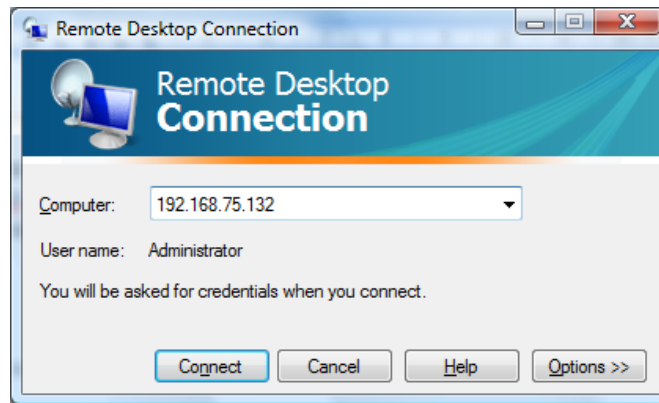
**L1.9** Go to WINDOWS2003, and go into the C:\Inetpub\mailroot\Queue folder, and view the queued email message.

☞ Outline the format of the EML file:

### **SERVICE: Remote Desktop**

**L1.10** From the host, connect to WINDOWS2003 using the Remote Desktop (Figure L1.3).


☞ Which is the service which is running on WINDOWS2003, that allows the remote connection to happen?



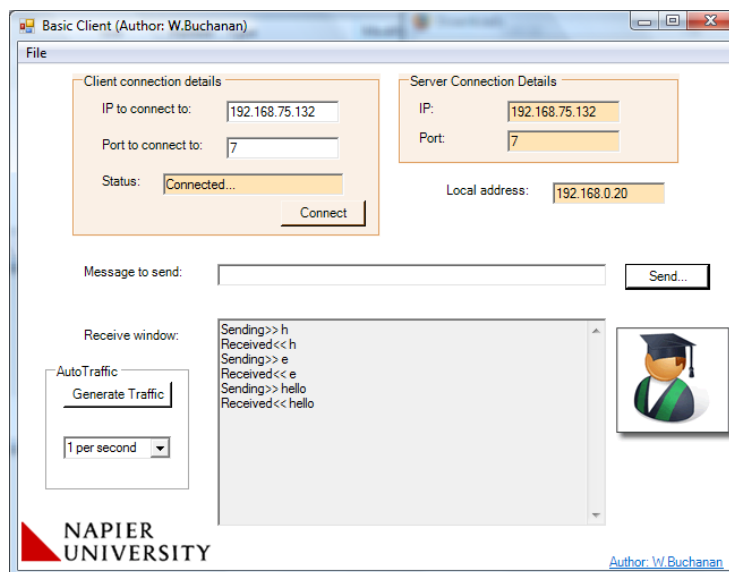
**Figure L1.3** Remote desktop connection

**SERVICE: Find the service?**

**L1.11** From your host connect to Port 7 using the client.exe program from (Figure L1.4):

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

 What is the service?



**Figure L1.4** Port 7 access

**L1.12** Run the server (on port 1024) on WINDOWS2003, and run the client from DESKTOP, and make a connection between them.

**AUDIT/LOGGING**

**L1.13** Auditing and logging are important in terms of tracing activities.

 Check in the Event Viewer in WINDOW2003 (Figure L1.5), that the logon event has been added. How might this be used to trace activity?

- ☞ From Local Security Policy, find the option to change option so that Privileged Access is audited. What is the option:
  
- ☞ From Local Security Policy, find the option to change option so that the Guest Account cannot login. What is the option:

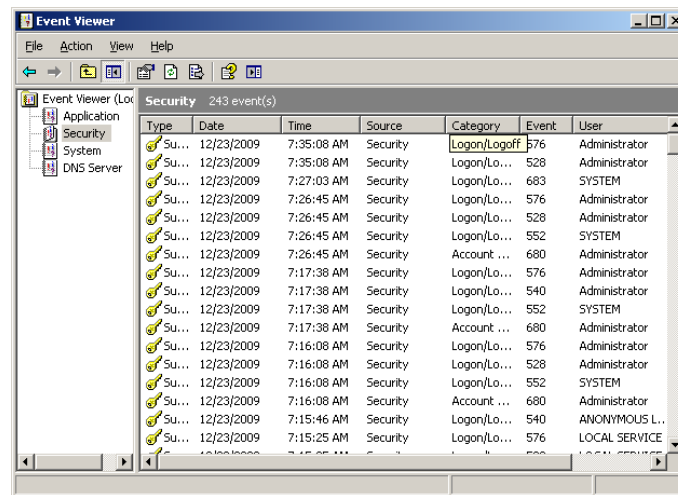


Figure L1.5 Event viewer

## 1.3 Toolkit 1

☞ On-line demo:  
[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/toolkit01/toolkit01.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/toolkit01/toolkit01.htm)

The objective of this series of labs is to build an integrated toolkit. Open up:

<http://buchananweb.co.uk/toolkit.zip>

and extract to a local folder. Next open up toolkit.sln, and double click on client.cs.

**L1.14** Select the [Network] table, and double click on the “netstat -a” button, and add the code:

```
runProgram("netstat", "-a");
```

and test the program.

**L1.15** Select the [Network] table, and complete the rest of the buttons (see <http://buchananweb.co.uk/dotnetclientserver.zip> for the functions required).

Week	Date	Teaching	Attended
3	25/1/2010	Lab 1: Linux Services/Toolkit 2	
<p><b>Aim:</b> The aim of this lab is to investigate the discovery and configuration of services within Linux. It uses the Ubuntu VM image (UBUNTU).</p> <p><b>Time to complete:</b> 4/5 hours (Two supervised hours in B.56, and two/three additional hours, unsupervised).</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Complete Lab 2.</li> </ul> <p><b>Learning activities:</b> At the end of these activities, you should understand:</p> <ul style="list-style-type: none"> <li>• How to define services in Unix.</li> <li>• How to interface to WinDump for the toolkit</li> </ul> <p><b>Reflective statements (end-of-exercise):</b> What are the key Linux commands used to discover the services which are being run?  What is the key folder location for the Web server in Linux?  Why does Linux need the VNC Client, when Windows uses the Remote Desktop Client?</p> <p><b>Source code used:</b>  <a href="http://buchananweb.co.uk/toolkit.zip">http://buchananweb.co.uk/toolkit.zip</a></p>			

## 2 Lab 2: Linux Services

### 2.1 Details

---

Aim: To provide a foundation in setup and consuming Linux services.

### 2.2 Activities

---

🔗 On-line demo:  
[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/unix/unix.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/unix/unix.htm)

PART B. This part of the lab has two elements: the host machine (DESKTOP) and the Linux virtual image (UBUNTU).

**L2.1** Run the Linux virtual image (User name: Administrator, Password: napier123). Within the virtual image, run a Terminal and determine its IP address using **ifconfig**.

👉 What are the IP addresses of the server and the network address which will be used to connect to the virtual image:

**L2.2** From DESKTOP, ping UBUNTU, and vice-versa.

#### **SERVICE: Web**

**1.2** In UBUNTU, go to the folder `/var/www`.

👉 What are the names of the files in this folder:

**L2.3** From UBUNTU, run `netstat -l`, and determine the services that are running.

👉 List some of the services:

**L2.4** From your host, connect to the Web Server from DESKTOP using `http://w.x.y.z`, where `w.x.y.z` is the IP address of UBUNTU (Figure L2.1). Repeat this using:

```
telnet w.x.y.z 80
```

and then enter:

GET /index.html

☞ What is the result, and how does it relate to accessing the home page of the Web server?

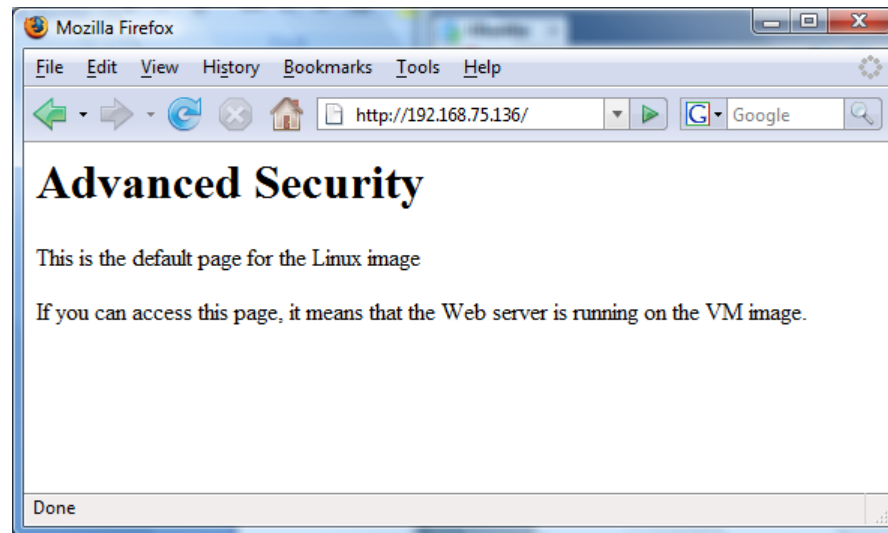


Figure L2.1 HTTP connection

L2.5 On UBUNTU, using Scream HTML/XML Editor, open up the /var/www and create a new page, such as:

## My Home Page

This is a sample ASP.NET page. Click [\[here\]](#) to return to the default home file.

☞ Can you access this page from the host (DESKTOP)?

☞ On UBUNTU, go to /var/log/apache2. What is the contents of the folder, and what do the files contain?

☞ How might these log files be used to trace malicious activity?

**SERVICE: Telnet**

**L2.6** From your host, connect to the Telnet Server from DESKTOP using telnet x.y.y.z, where w.x.y.z is the IP address of UBUNTU. Login in with napier (password: napier123).

☞ What is the default home folder for Telnet on UBUNTU (use pwd to determine the current directory):

Quit from Telnet, using the “exit” command.

### **SERVICE: FTP**

**L2.7** From your host, connect to the FTP Server from DESKTOP using ftp://w.x.y.z where w.x.y.z is the IP address of UBUNTU (Figure 1.24). Repeat this using:

telnet w.x.y.z 21

and then enter the commands in bold (and note the commands that you get beside the sample return ones):

```
USER napier
331 Password required for napier.
PASS napier123
230- Linux ubuntu 2.6.31-14-generic #48-Ubuntu SMP Fri Oct 16 14:04:26 UTC 2009
      i686
230-
230- To access official Ubuntu documentation, please visit:
230- http://help.ubuntu.com/
230-
230 User napier logged in.
PWD
257 "/home/napier" is current directory.
TYPE I
200 Type set to I.
PASV
227 Entering Passive Mode (192,168,75,136,146,31)
LIST
```

Now FTP opens up a port for the data transfer. This is calculated from the last two digits of the Passive Mode response (227 response). It is calculated as 146 times the second last digital plus the last digital (31). So, in this case, it is:

Port =  $146 * 256 + 31 = 37397$

Next open up the data transfer by creating a new Telnet connection, such as:

telnet w.x.y.z 37397

- ☞ Can you access this page from the host?
- ☞ On UBUNTU, go to /var/log. View the syslog file (such as with **cat syslog**). What is its contents?
- ☞ How might these log files be used to trace malicious activity?
- ☞ View the contents of /etc/inetd.conf file. How is this used to enable services?

### **SERVICE: Remote Desktop**

**L2.8** Download the VNC Client from:

<http://www.realvnc.com/cgi-bin/download.cgi>

then from the host, connect to UBUNTU using the Remote Desktop (Figure L2.2).

- ☞ Which is the service which is running on UBUNTU that allows the remote connection to happen?

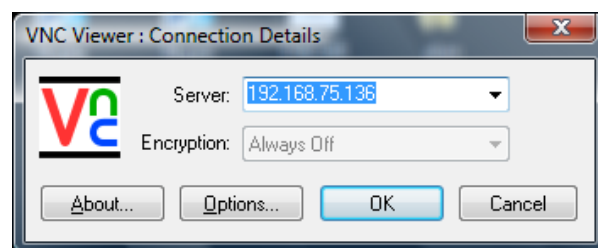


Figure L2.2 VNC Viewer

## **2.3 Toolkit 2**

- ☞ On-line demo:  
[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/toolkit02/toolkit02.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/toolkit02/toolkit02.htm)

The objective of this series of labs is to build an integrated toolkit. Open up:

<http://buchananweb.co.uk/toolkit.zip>

and extract to a local folder. Next open up toolkit.sln, and double click on client.cs.

**L2.9** Select the [WinDump] table, and double click on the Combo Box (cbInterfacesWin). Next add the following code:

```
stopProcess("windump");
if (processCaller2 != null) processCaller2.Cancel();

processCaller2 = null;
int ind = cbInterfacesWin.SelectedIndex+1;
string args="-q -i "+ind;
if (this.cbVerbose.Checked) args += " -v ";
if (tbOption.Text.Length > 0) args += " " + tbOption.Text;

runProgram2("WinDump.exe",args );
```

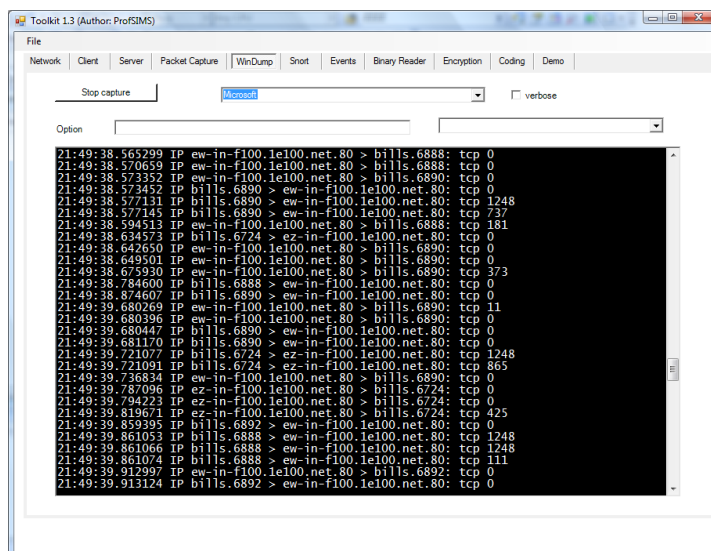
Next add the method:

```
public void stopProcess(string name)
{
    try
    {
        Process[] pArray = Process.GetProcesses();

        foreach (Process p1 in pArray)
        {
            string s = p1.ProcessName;
            s = s.ToLower();

            if (s.CompareTo(name) == 0)
            {
                p1.Kill();
            }
        }
    }
    catch (Exception ex)
    {
    }
}
```

and test the program (Figure L2.6).



**Figure L1.6** WinDump

Week	Date	Teaching	Attended
4	1/2/2010	Lab 3: Vulnerability Analysis/ Toolkit 3	
<p><b>Aim:</b> The aim of this lab is to investigate possible vulnerabilities for Windows and Ubuntu.</p> <p><b>Time to complete:</b> 4 hours (Two supervised hours in B.56, and two additional hours, unsupervised).</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Complete Lab 3.</li> </ul> <p><b>Learning activities:</b> At the end of these activities, you should understand:</p> <ul style="list-style-type: none"> <li>• How to detect the scanning for vulnerabilities.</li> <li>• How SQL can be passed through a Web page to the database.</li> <li>• How to use vulnerability tools to perform a penetration test.</li> <li>• How to integrate with Snort/Nmap within a toolkit.</li> </ul> <p><b>Reflective statements (end-of-exercise):</b> Why are tools such as Nmap, hping, hydra and so on, seen as malicious, while also being useful?</p> <p>What methods would be applied to stop SQL Injection?</p> <p><b>Source code used:</b></p> <p><a href="http://buchananweb.co.uk/toolkit.zip">http://buchananweb.co.uk/toolkit.zip</a></p>			

# 3 Lab 3: Vulnerability Analysis

## 3.1 Details

---

**Aim:** The aim of this lab is to investigate possible vulnerabilities for Windows and Ubuntu.

## 3.2 Activities

---

**L3.1** Run the Windows Server 2003 virtual image (User name: Administrator, Password: napier). Within the virtual image, run the command prompt and determine its IP address using **ipconfig**.

**L3.2** Run the Linux virtual image (UBUNTU) (User name: napier, Password: napier123). Within the virtual image, run the command prompt and determine its IP address using **ifconfig**.

**L3.3** From WINDOWS2003, run **nmap** on WINDOWS2003 and UBUNTU, and vice-versa. Note the services discovered:

**Windows Services:**

**Linux Services:**

**L3.4** From WINDOWS2003, run **windump -i 2**, and run **nmap** on UBUNTU.

**What can be observed from WINDOWS2003:**

**L3.5** From UBUNTU, run **sudo /usr/sbin/tcpdump -i eth0**, and run **nmap** on WINDOWS2003.

**What can be observed from UBUNTU:**

**L3.6** From WINDOWS2003, run Nessus, and conduct a port scan of UBUNTU to discover the services which are running:

**Ports open:**

**L3.7** From WINDOWS2003, create a folder named `zzzzzzz` (where `zzzzzzz` is your matriculation number) and create a file in this folder named `icmp.rules`, and add:

```
var EXTERNAL_NET any
var HOME_NET any
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Windows";
  itype:8; content:"abcdefghijklmnop";depth:16; sid:999)
```

and run Snort on WINDOWS2003 with:

```
snort -c test.rules -i 2 -p -l c:\\zzzzzzzz -K ascii
```

and from UBUNTU, perform a ping on WINDOWS2003.

**Did Snort detect the ping scan:**

**L3.8** From WINDOWS2003, create `portscan.rules`, and add:

```
var EXTERNAL_NET any
var HOME_NET any
preprocessor sfportscan: proto { all } scan_type { all } sense_level { high
  } logfile { portscan.log }
```

and run Snort on WINDOWS2003 with:

```
snort -c test.rules -i 2 -p -l c:\\zzzzzzzz -K ascii
```

and from UBUNTU, perform an nmap on WINDOWS2003.

**Did Snort detect the port sweep:**

**L3.9** From WINDOWS2003, create a rule which detects an incoming SYN from another host.

**L3.10** Create a new user on the FTP server in UBUNTU, using (check by viewing the `/etc/passwd` file):

```
sudo useradd fred -p fred -d /home/fred -s /bin/false
```

Next try and find the password by going to WINDOWS2003, and running hydra, such as:

```
C:\hydra-5.4-win> hydra -L login.txt -P passwd.txt 192.168.75.x ftp
```

What modifications were required to detect the user fred:

### 3.3 SQL injection tutorial

#### SQL Injection Demo:

[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/cross\\_script/cross\\_script.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/cross_script/cross_script.htm)

**L3.11** Run the Windows Server 2003 virtual image (User name: Administrator, Password: napier). Run Visual Web Developer Express 2008, and select Open Web Site, and select c:\inetput\wwwroot.

**L3.12** On the Database Explorer, select Connect to Database, and setup as in Figure L3.1.

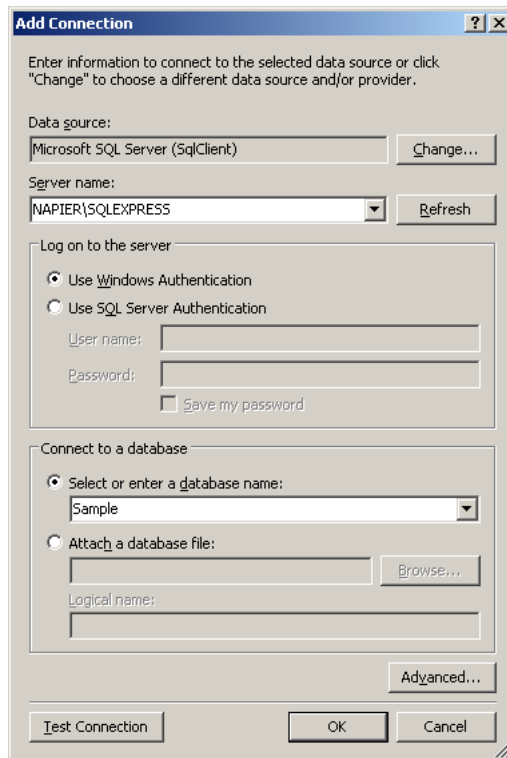


Figure L3.1 Database connection

**L3.13** Create a new **databasesample.aspx** Web page, and add a GridView component. Double click on the form, and then add the following code:

```
protected void Page_Load(object sender, EventArgs e)
{
    SqlCommand s = null;
    string param = Request.QueryString["test"];
    MySqlConnection=createConn("Sample");
    MySqlConnection.Open();
    s = new SqlCommand("SELECT * FROM db1", MySqlConnection);

    if (param != null) s = new SqlCommand(param, MySqlConnection);
    SqlDataReader myDataReader = s.ExecuteReader();

    GridView1.DataSource = myDataReader;
    GridView1.DataBind();

    closeConn();
}
```

Next add the following code:

```
public SqlConnection mySqlConnection;
public SqlCommand mySqlCommand;
public SqlDataReader mySqlDataReader;

private void closeConn()
{
    if (mySqlConnection != null)
    {
        if (mySqlConnection.State == ConnectionState.Open)
        {
            mySqlConnection.Close();
        }
        mySqlConnection.Dispose();
    }
}

private SqlConnection createConn(string database)
{
    string mySqlConnectionString =
@"Data Source=NAPIER\SQLEXPRESS;Initial Catalog=Sample;
Integrated Security=True";

    if (mySqlConnection == null) {
        mySqlConnection = new SqlConnection(mySqlConnectionString); };

    return mySqlConnection;
}
```

```
}
```

**L3.14** Set `databasesample.aspx` as the default startup, and press Start Debugging (F5).

**What are the contents of the table:**

**L3.15** Next replace the `s = new SqlCommand("SELECT * FROM db1", mySqlConnection);` line with:

```
s = new SqlCommand("INSERT INTO db1 VALUES ('Bert',  
'Smith4','25','25','35')", mySqlConnection);
```

and execute. After this replace the original line, and rerun the code.

**What are the contents of the table:**

**Has a new line been added:**

**L3.16** Next from the Host computer (HOST), access the Web server with:

```
http://192.168.75.132/databasesample.aspx?test=SELECT%20*%20FROM%20db1
```

**L3.17** Next from the Host computer (HOST), access the Web server with:

```
http://192.168.75.132/databasesample.aspx?test=INSERT%20INTO%20db1%20VALUES%  
20('Bert','Smith6','35','55','95')
```

followed by:

```
http://192.168.75.132/databasesample.aspx?test=SELECT%20*%20FROM%20db1
```

**What are the contents of the table:**

**Has a new line been added:**

**L3.18** Create an SQL injection in calculate the average mark for Test 1, such as for:

```
s = new SqlCommand("SELECT avg([Test 1]) FROM db1",  
mySqlConnection);
```

Test on the local Web server, and then use an SQL injection from a URL. Repeat for the minimum and maximum mark for Test 1.

**L3.19** With an SQL injection, change Ian Archibalds mark to 100%.

**L3.20** Modify the code so that it detects an SQL inject, and identifies the SQL command used.

## 3.4 Toolkit 3 (Snort/NMAP)

🔗 On-line demo:  
[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/toolkit02/toolkit02.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/toolkit02/toolkit02.htm)

The objective of this series of labs is to build an integrated toolkit. Open up:

<http://buchananweb.co.uk/toolkit.zip>

and extract to a local folder. Next open up toolkit.sln, and double click on client.cs.

**L3.21** Select the [Snort] table, and double click on the "Show Interf" button and add the following code:

```
runProgram3("snort", "-W");
```

on View IDS add:

```
Process.Start("notepad.exe", "log\\alert.ids");
```

on the View button:

```
Process.Start("notepad.exe", tbSnortFile.Text);
```

on View ARP:

```
Process.Start("wordpad.exe", "log\\ARP.");
```

on Delete alert.ids:

```
try {  
    File.Delete("log\\alert.ids");
```

```

        File.Delete("log\\ARP");
    }
    catch (Exception ex)
    { }

```

on the ComboBox (cbInterfacesSnort) and add the code:

```

stopProcess("snort");
if (processCaller3!=null) processCaller3.Cancel();

processCaller3 = null;

int ind = cbInterfacesSnort.SelectedIndex + 1;
string args = " -p -K ascii -N ";
args += " -i " + ind;
if (this.cbVerboseSnort.Checked) args += " -v ";
if (tbSnortFile.Text.Length > 0) args += " -c " +
tbSnortFile.Text;

runProgram3("snort.exe", args);
timer2.Enabled = true;

```

**L3.22** Download nmap.exe, and integrate it in with the toolkit (add a box for an IP address to scan), and an options box.

Week	Date	Teaching	Attended
5	15/2/2010	Lab 4: Network Forensics/Toolkit 4	
<p><b>Aim:</b> The aim of this lab is to investigate network traffic flows, and make sense of them.</p> <p><b>Time to complete:</b> 4 hours (Two supervised hours in B.56, and two additional hours, unsupervised).</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Complete Lab 4.</li> </ul> <p><b>Learning activities:</b> At the end of these activities, you should understand:</p> <ul style="list-style-type: none"> <li>• How to analyse network packet dumps.</li> <li>• How to use Trip wire.</li> <li>• How to integrate WinPCap into the Toolkit.</li> </ul> <p><b>Reflective statements (end-of-exercise):</b> What might be the problems (such as for IP address spoofing) that could cause a network capture in question?</p> <p><b>Source code used:</b></p> <p><a href="http://buchananweb.co.uk/dotnetclientserver.zip">http://buchananweb.co.uk/dotnetclientserver.zip</a>  <a href="http://buchananweb.co.uk/toolkit.zip">http://buchananweb.co.uk/toolkit.zip</a></p>			

# 4 Lab 4: Network Forensics

## 4.1 Details

Aim: The aim of this lab is to investigate network traffic flows, and make sense of them.

## 4.2 Activities

A. Download, install and run client.exe from:

<http://buchananweb.co.uk/dotnetclientserver.zip>

B. Within Toolkit, select the Packet Capture tab and then the Open TCPDump tab.

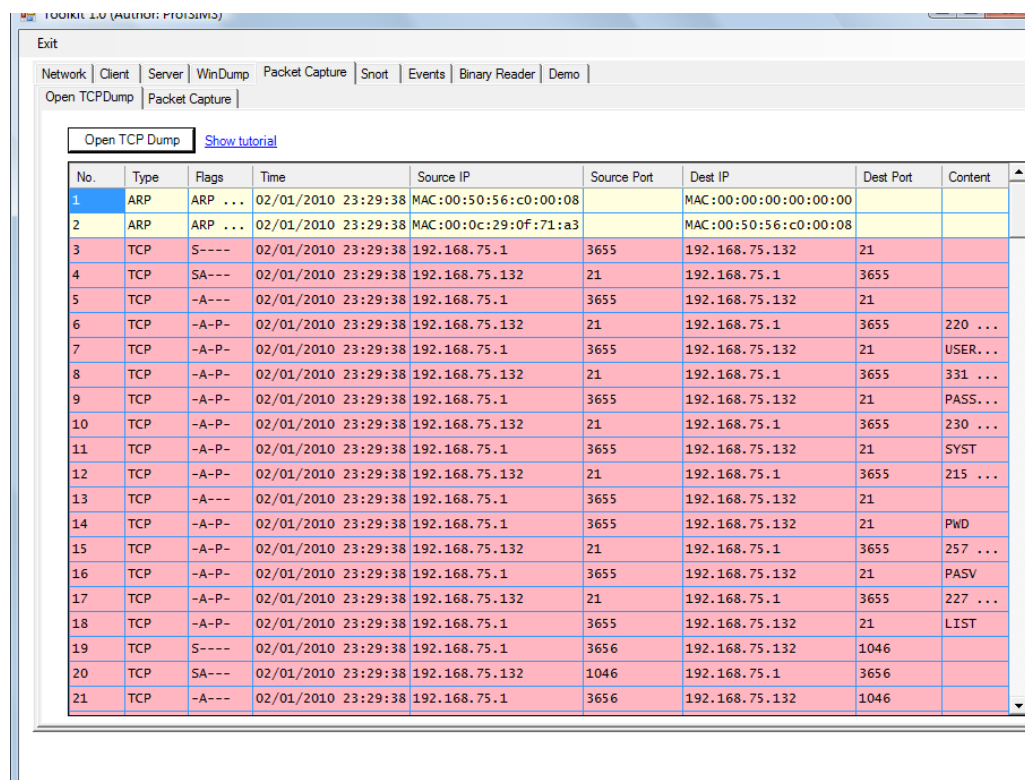
### FTP Analysis Demo:

[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/tcpdump01/tcpdump01.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/tcpdump01/tcpdump01.htm)

Note: If you prefer to use Wireshark, the Pcap dump files are at:

<http://buchananweb.co.uk/log/>

### L4.1 Open ftp dump (see Figure L1.1).



No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	ARP	ARP ...	02/01/2010 23:29:38	MAC:00:50:56:c0:00:08		MAC:00:00:00:00:00:00		
2	ARP	ARP ...	02/01/2010 23:29:38	MAC:00:0c:29:0f:71:a3		MAC:00:50:56:c0:00:08		
3	TCP	S----	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	
4	TCP	SA---	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	
5	TCP	-A---	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	
6	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	220 ...
7	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	USER...
8	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	331 ...
9	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	PASS...
10	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	230 ...
11	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	SYST
12	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	215 ...
13	TCP	-A---	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	
14	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	PwD
15	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	257 ...
16	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	PASV
17	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	227 ...
18	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	LIST
19	TCP	S----	02/01/2010 23:29:38	192.168.75.1	3656	192.168.75.132	1046	
20	TCP	SA---	02/01/2010 23:29:38	192.168.75.132	1046	192.168.75.1	3656	
21	TCP	-A---	02/01/2010 23:29:38	192.168.75.1	3656	192.168.75.132	1046	

### Figure 3.1 FTP Dump

Determine the following:

Host src TCP port (Hint: Examine the Source Port on Packet 3):

Server src TCP port (Hint: Examine the Destination Port on Packet 3):

Host src IP address (Hint: Examine the Source IP on Packet 3):

Server src IP address (Hint: Examine the Dest IP on Packet 3):

What is the MAC address of the server (Hint: Examine the reply for Packet 2):

Identify the packets used for the SYN, SYN/ACK and ACK sequence (Hint: packets 3 to 5 look interesting):

Which is the return code used by the FTP server to identify:

Password Required (Hint: Examine the content on Packet 9):

Server type (Hint: Examine the content on Packet 12):

Which FTP command is used to determine the current working folder (Hint: Examine the content on Packet 15):

Which FTP command is used to determine the files in a folder (Hint: Examine the content on Packet 18):

Which FTP port has been used for the FTP directory list (hint: Examine the contents of Packet 17, and the last two digits of the 227 response (first multiplied by 256 added to the second):

Identify the data packets used to list the contents (Hint port 1046 looks interesting):

Which FTP port has been used for the FTP file transfer (hint: it is the last two digits of the 227 response (first multiplied by 256 added to the second):

Identify the data packets used to transfer the file:

What is the name of the file transferred:

L4.2 Open Telnet dump.

Determine the following:

**Host src TCP port:**

**Server src TCP port:**

**Host src IP address:**

**Server src IP address:**

**Identify the packets used for the SYN, SYN/ACK and ACK sequence:**

**What is the login name:**

**What is the password:**

**What commands were entered, once the Telnet connection was made:**

L4.3 Open dns dump.

Determine the following:

**What is the transport layer protocol used for DNS:**

**Host src UDP port:**

**Server (DNS) src UDP port:**

**Host src IP address:**

**Server (DNS) src IP address:**

**Identify the data packets used to for the DNS lookup:**

L4.4 Open ping dump.

Determine the following:

**Host src IP address:**

**Server (DNS) src IP address:**

**Identify the data packets used to for the ping:**

**How many ECHO's where send from the host, and how many replies where there:**

**L4.5 Open webpage dump.**

Determine the following:

**Host src TCP port:**

**Server src TCP port:**

**Host src IP address:**

**Server src IP address:**

**Identify the packets used for the SYN, SYN/ACK and ACK sequence:**

**What is the HTTP command used to get the default page (Hint: put your cursor over the content of the 4<sup>th</sup> data packet):**

**What is the HTTP response to a successful request (Hint: put your cursor over the content of the 5<sup>th</sup> data packet):**

**L4.6 Open hping\_fin dump.** We can see that a remote host is sending TCP segments with the FIN flag sent.

Determine the following:

**Sending src TCP port range:**

**Receiver src TCP port:**

**Sending src IP address:**

**Receiver src IP address:**

**L4.7 Open hping\_port80 dump.** We can see that a remote host is sending TCP segments with the SYN flag sent.

Determine the following:

**Sending src TCP port range:**

**Receiver src TCP port:**

**Sending src IP address:**

**Receiver src IP address:**

**L4.8** Open **hydra\_ftp** dump. We can see that a Hydra attack has been conducted on our server.

Determine the following:

**Sending src TCP port range:**

**Receiver src TCP port:**

**Sending src IP address:**

**Receiver src IP address:**

**What are the logins used:**

**What are the passwords used:**

**What is the successful login/password:**

**L4.9** Open **hydra\_telnet** dump. We can see that a Hydra attack has been conducted on our server.

Determine the following:

**Sending src TCP port range:**

**Receiver src TCP port:**

**Sending src IP address:**

**Receiver src IP address:**

**What are the logins used:**

**What are the passwords used:**

**What is the successful login/password:**

**L4.10** Open `hping_udp_scan` dump.

Determine the following:


**Sending src UDP port range:**

**Receiver src UDP port:**


**Sending src IP address:**

**Receiver src IP address:**

## 4.3 Tripwire tutorial

 On-line demo:  
[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/tripwire/tripwire.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/tripwire/tripwire.htm)

**L4.11** Run the Linux virtual image (User name: Administrator, Password: napier123). Within the virtual image, run a Terminal and determine its IP address using `ifconfig`.


 What are the IP addresses of the server and the network address which will be used to connect to the virtual image:

**L4.12** Go to the `/etc/tripwire` folder, and view the `twpol.txt` file. Next run the following commands:

```
twadmin --create-polfile --cfgfile ./tw.cfg --site-keyfile ./site.key ./twpol.txt
tripwire --init --cfgfile /etc/tripwire/tw.cfg --polfile /etc/tripwire/tw.pol --site-keyfile /etc/tripwire/site.key --local-keyfile /etc/tripwire/ubuntu-local.key
```

**L4.13** Go to the `/etc/passwd` file and change the owner to “napier”. Next go to the `/tmp` folder and change the ownership of this file too. Next run a check with tripwire:

```
tripwire --check
```

 What do you observe from the results:

**L4.14** Create a new folder in your home directory, and add a rule to the policy file for Tripwire, and see if you can detect any changes on this folder.

☞ Rule used:

## 4.4 Toolkit 4 (Packet Capture/Analysis)

🔗 On-line demo:

[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/toolkit04/toolkit04.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/toolkit04/toolkit04.htm)

The objective of this series of labs is to build an integrated toolkit. Open up:

<http://buchananweb.co.uk/toolkit.zip>

and extract to a local folder. Next open up toolkit.sln, and double click on client.cs.

**L4.15** Select the [Packet Capture] and then [Open TCPDump] tab, and double click on the "Open TCP Dump" button and add the following code:

```
this.dgPackets.Rows.Clear();
        PcapDevice device = null;
        Packet packet = null;
        openFileDialog1.InitialDirectory = homeFolder+"\\log";
        openFileDialog1.Filter = "pcap files (*.pcap)|*.pcap|All files (*.*)|*.*";
        openFileDialog1.FilterIndex = 1;
        openFileDialog1.FileName = "*.pcap";
        openFileDialog1.ShowDialog();

        try
        {
            fileName = openFileDialog1.FileName;
            linkLabel7.Visible = true;
            tbFile.Text = fileName;
            device = SharpPcap.PcapOfflineDevice(fileName);
            device.PcapOpen();

        }
        catch (Exception e1)
        {
            MessageBox.Show("Error: " + e1.Message);
            return;
        }
        int count = 0;
        while ((packet = device.PcapGetNextPacket()) != null)
        {
            if (packet is TCPpacket)
            {
                count++;
                saveTCPpackets(false, count, dgPackets, packet);
            }

            else if (packet is ICMPpacket)
            {
            }
        }
    }
```

```

        count++;
        saveICMPPackets(false, count, dgPackets, packet);
    }
    else if (packet is UDPPacket)
    {
        count++;
        saveUDPPackets(false, count, dgPackets, packet);
    }
    else if (packet is ARPPacket)
    {
        count++;
        saveARPPackets(false, count, dgPackets, packet);
    }
    else
    {
        this.dgPackets.Rows.Add(count.ToString(), "N/K", "", "", "", "",
"", "", "");
    }
}
try
{
    dgPackets.CurrentCell = this.dgPackets[0, 0];
}
catch { }

```

Then add the following code to saveTCPPackets():

```

try
{
    string time = packet.PcapHeader.Date.ToShortTimeString();
    TCPpacket tcp = (TCPpacket)packet;
    string srcIp = tcp.SourceAddress;
    string dstIp = tcp.DestinationAddress;
    int srcPort = tcp.SourcePort;
    int dstPort = tcp.DestinationPort;

    ASCIIEncoding utf = new System.Text.ASCIIEncoding();
    string s = utf.GetString(getridofnonprint(tcp.Data));

    CreateMessageForStatusAppend(dg, count.ToString(), "TCP", showflag(tcp),
time.ToString(), srcIp, srcPort.ToString(), dstIp, dstPort.ToString(), s);

    if (realtime == false)
    {
        dg.RowsDefaultCellStyle.BackColor = Color.LightPink;
        for (int i = 0; i < 9; i++)
        {
            dg.CurrentCell = dg[i, count - 1];

            dg.CurrentCell.ToolTipText = string.Format("{0}:\r\n{1}",
dg.Columns[i].HeaderText, dg[i, count - 1].Value);
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show("Exception (TCP) Save:" + ex.Message);
}

```

And for saveUDPPackets ():

```

try
{
    string time = packet.PcapHeader.Date.ToShortTimeString();
    int len = packet.PcapHeader.PacketLength;

    UDPPacket udp = (UDPPacket)packet;
    string srcIp = udp.SourceAddress;
    string dstIp = udp.DestinationAddress;
    int srcPort = udp.SourcePort;
    int dstPort = udp.DestinationPort;
    int packetLength = udp.Length;
    ASCIIEncoding utf = new System.Text.ASCIIEncoding();
    string s = utf.GetString(getridofnonprint(packet.Data));
}

```

```

        //count++;
        CreateMessageForStatusAppend(dg, count.ToString(), "UDP", "",
time.ToString(), srcIp, srcPort.ToString(), dstIp, dstPort.ToString(), s);

        if (realtime == false)
        {
            dgPackets.RowsDefaultCellStyle.BackColor = Color.LightYellow;

            for (int i = 0; i < 9; i++)
            {
                dg.CurrentCell = dg[i, count - 1];
                dg.CurrentCell.Style.BackColor = Color.PowderBlue;
                dg.CurrentCell.ToolTipText = string.Format("{0}:\r\n{1}",
dg.Columns[i].HeaderText, dg[i, count - 1].Value);
            }
        }
        catch (Exception ex)
        {
            // MessageBox.Show("Exception (UDP): " + ex.Message);
        }
    }
}

```

**L4.16** Now do the same for `saveICMPPackets()` and `saveARPPackets()`. Using a similar code.

**L4.17** Select the [Packet Capture] and then [Packet Capture] tab, and double click on the "Start Capture" button and add the following code:

```

packetcount = 0;
dgPackets1.Rows.Clear();
enableNewInterface();

```

and add the following code to `enableNewInterface()`:

```

packetcount = 0;
try
{
    if (comboBox2.Text == "" || comboBox2.Text.StartsWith("-") )
    {
        CreateMessageForStatus(tbPacket, "Interface not set yet!");
        return;
    }
    if (device != null)
    {
        device.PcapStopCapture();
        device.PcapClose();
        CreateMessageForStatus(tbPacket, "Interface disconnected");
        device = null;
    }

    PcapDeviceList getNetConnections = SharpPcap.GetAllDevices();

    NetworkDevice netConn =
(NetworkDevice)getNetConnections[comboBox2.SelectedIndex];

    device = netConn;
    CreateMessageForStatus(tbPacket, "Network connection: " +
device.PcapDescription+"\r\n");
    device.PcapOpen(true, 500);

    device.PcapOnPacketArrival +=
new
SharpPcap.PacketArrivalEvent(device_PcapOnPacketArrival);

    device.PcapSetFilter(tbFilter.Text);
    CreateMessageForStatusAppend(tbPacket, "Filter: " + tbFilter.Text
+ "\r\n");
    device.PcapStartCapture();
}

```

```
    }  
    catch (Exception ex)  
    {  
        CreateMessageForStatusAppend(tbPacket, "Problem opening connection.  
Error: " + ex.Message);  
    }
```

**L4.18** Test the program for opening TCP dumps, and for packet capture.

Week	Date	Teaching	Attended
6	15/2/2010	Lab 5: Data Hiding/ Toolkit 5 (File Analysis)	
<p><b>Aim:</b> The aim of this lab is to investigate data hiding and file type analysis.</p> <p><b>Time to complete:</b> 4 hours (Two supervised hours in B.56, and two additional hours, unsupervised).</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Complete Lab 5.</li> </ul> <p><b>Learning activities:</b> At the end of these activities, you should understand:</p> <ul style="list-style-type: none"> <li>• How to analyse file for their file type.</li> <li>• How to understand encoding formats.</li> <li>• How to investigate differing coding/encryption methods.</li> </ul> <p><b>Reflective statements (end-of-exercise):</b> Why is Base-64 uses for sending email attachment?  How would you identify a possible Base-64 encoding format?  How would you identify a possible Hex encoding format?</p> <p><b>Source code used:</b>  <a href="http://buchananweb.co.uk/dotnetclientserver.zip">http://buchananweb.co.uk/dotnetclientserver.zip</a> <a href="http://buchananweb.co.uk/toolkit.zip">http://buchananweb.co.uk/toolkit.zip</a></p>			

# 5 Lab 5: Data Hiding

## 5.1 Details

---

Aim: The aim of this lab is to investigate data hiding and file type analysis.

## 5.2 Activities

---

Download and install: <http://buchananweb.co.uk/dotnetclientserver.zip>

**L5.1 Select [Encryption->Hashing tab]** Determine the Base-64 hash signature for "test" for the following:

MD5:

SHA-1:

SHA-256

How many bits does each of these signatures have:

**L5.2 Select [Encryption->Hash (Collision) tab]** Determine the ASCII message for the following hash signatures:

AD5F82E879A9C5D6B5B442EB37E50551

15B6AF8D85CBE1229C7150E10D5A55BD3417B40C

EEBC8CF2B3B360C51A34E0E8EBD98B8F37F348B7

1F7BA58706F9D405023DA32864D059C8

**L5.3 Select [Encryption->Base-64 tab]** Determine the ASCII message from the following Base-64 messages:

SGVsbG8gaG93IGFyZSB5b3U/

Q2FuIH1vdSBYXZlcnN1IG10Pw==

VGhpcyBpcyBhIHh0b3R0ZSBwaWVjZSBvZiB0ZXh0Li4u

**L5.4 Select [Encryption->Base-64 tab]** Determine the Base-64 string for the following encrypted strings in 3DES and AES, which have been encrypted with the key word of "sample1234":

napier  
fullstop  
apple.tree

How many bits does the result have, and how does it vary for the following words, and explain the reason for the changes in the output size:

aaaaa  
aaaaaa  
aaaaaaa  
aaaaaaaa  
aaaaaaaaa  
aaaaaaaaaa  
aaaaaaaaaaa

What does the "=" represent at the end of the encrypted string?

**L5.5 Select [Encryption->Private-key encryption tab]** The result of an encryption process is "7xCJIB1RVG5/2HQFrDH9Kw==", which was encrypted from either "foxtrot", "orangepeel", or "interrupt".

**Which password was used, what encryption type was used, and what was the original message:**

**L5.6 Select [Encryption->Brute force tab]** Using a brute-force dictionary search, determine the AES encryption key for the following:

**Determine the encryption key, and the original message:**

2AC3B3211DEADC97C824307090BD33EA  
194E22BF7A463D8A048140400497DCA7  
F2BE257B9B13B72634013D9E528B6A9F  
60FA30C4E4EAFF88EB741BCEEE976CD7D66DC12EBE2C9425C331F4B01FC65A  
2A

**L5.7 Select [Encryption->Public-key encryption/decryption tabs]** Download the following public-key:

<http://buchananweb.co.uk/publickey01.txt>

and use it to encrypt the word "test", and prove the result is:

"17500DDDBD378..."

**L5.8 Select [Encryption->Public-key encryption/decryption tabs]** Download the following private-key:

<http://buchananweb.co.uk/privatekey01.txt>

and provide that it can decrypt the ciphertext.

**L5.9 Select [Encryption->Public-key encryption/decryption tabs]** Using the private-key (<http://buchananweb.co.uk/privatekey02.txt>), and the following cipher stream (copy it from the PDF document), determine the message:

```
2FB7C6F9719A05E79FA0591E92CE1884DB9CDB015F4F29D405B7ED5216
03AFEB404E9884BE0F83597C3054BC721CD0F15E39091B7894B11929CA
CFE7B77F7A29DD41ED3AC27D4C825157B61A1775B104045731A1B3CDD8
BDDCB091544D2FAC7D50DEBC8AD79D1BE1F73999D7FE6B8E8AB61142B7
1A0F274E0053D9C1FE3B80F3
```

**What is the message:**

**L5.10 Select [Encryption->Digital certificate tab]** Open up fred.cer, and determine its main parameters:

**Certificate details:**

**L5.11 Select [Encryption->Digital certificate tab]** Open up sample01, sample02, sample03, sample04 and sample05, and determine their passwords:

**Passwords on certificates:**

**L5.12 Select [Coding->Ex-OR tab]** If the message is "Testing", what is the single digital Ex-OR key for the following Base-64 strings:

NwYQFwONBA==

EiM1Mi8oIQ==

Lh8JDhMUHQ==

**L5.13 Select [Coding->Encoding tab]** Determine the message for the following encoding formats:

48656C6C6F20686F772061726520796F753F

2431323334353637383924

VGvzdGluZyAiMTizIiAuLi4=

**L5.14 Select [Coding->Caesar code tab]** Determine the message for the following Caesar codes:

OLSSV OVD HYL FVB

MABL BL HGER T FXLLTZX

PEEAT RDGT

**L5.15 Select [Binary Reader],** for open the first file (file1). The output should be something like in Figure 1.

**Refer to the Appendix given, and determine the format of the file.**

**What is the format of the file (such as GIF, JPEG, ZIP, etc):**

Now repeat for files 2 to 10, and complete the following table:

<i>Name</i>	<i>File format (circle correct one)</i>	<i>Is there any copyright information in the file (or associated information that is readable)?</i>
<i>File2</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	
<i>File3</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	
<i>File4</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	
<i>File5</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	
<i>File6</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	
<i>File7</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	
<i>File8</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	
<i>File9</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	
<i>File10</i>	<i>DOC/PPT/XLS/JPEG/GIF/WMF/ZIP</i>	

**L5.16 Select [Binary Reader], for the ZIP file:**

**Identify the file name contained within the ZIP file:**

**What is the termination character used to terminate the file name:**

**Can you tell the date and time that it was last modified?**

**L5.17 For other binary file formats, determine their signature (if possible).**

**PDF file signature:**

**SWF (Flash) file signature:**

DLL file signature:

RTF file signature (open up a Word document, and save it in an RTF file format):

**L5.18 Select [Coding]**, performance a frequency analysis on the following, and determine the original text:

XQG XP MJG PAEDM XBBKEEQBGD XP BXC-LKMGE MGBJQHXFT XBBKEEGO AQ MJG KDY AQ MJG 1880D. AM VYD OKG MX MJG YCGEABYQ BXQDMAMKMAXQ OGCYQOAF MJYM Y DKERGT UG KQOGEMYIGQ GRGET 10 TGYED. YD MJG LXLKHYMAXQ AQ MJG KDY AQBEGYDGO, AM MXXI YQ AQBEGYDAQF YCXKQM XP MACG MX LEXOKBG MJG DMYMAD-MABD. UT MJG 1880D, AM HXXIGO HAIGHT MJYM MJG 1880 DKERGT VXKHO QXM UG BXCLHGMG KQMAH 1890. MX XRGEBCXG MJAD, JGECYQ JXHHGEAMJ (VJX VXEIGO PXE MJG FXRGEQCGQM) OGRADGO Y CY-BJAQG MJYM YBBGLMGO LKQBJ BYEOD VAMJ AQPXECYMAXQ XQ MJGC. MJGDG BYEOD YHHXVGO Y BKEEQM MX LYDD MJEXKFJ XQHT VJGQ MJGEG VYD Y JXHG LEGDQGM.

JXHHGEAMJ'D GHGBMEXCGBJYQABYH CYBJAQG VYD GZMEGCGHT DKBBGDDPKH YQO VYD KDGO AQ MJG 1890 YQO 1900 BGQDKDGD. JG GRGQ PXXQOGO MJG BXCLYQT MJYM VXKHO HYMG UGBXCG AQMGQY-MAXQYH UKDAQGDD CYBJAQGD (AUC).

**L5.19 Select [Coding]**, performance a frequency analysis on the following, and determine the original text:

FN 1985, GLLBK TGH IGOFNE AFXXFUMBJ JFSKH. JIK HGBKH CX JIK SGUFNJCHI TKWK NCJ GH EWKGJ GH KRLKUJKA, GNA JIK GLLBK FF TGH XGUFNE G EWKGJ AKGB CX UCCLKJFJFCN XWCS CJKW SGNMXGUJMWKWH. SGNV LKCLBK GJ JIK JFSK, FNUBMAFNE QFBB EGJKH, TKWK GAOFHNE GLLBK JC CLKN-ML JIK SGWDKJ XCW SGUFNJCHI UCCLMJKWH QY GBBCTFNE CJKW SGNMXGUJMW-KWH QMFBA JIKFW CTN HYHJKSH, MNAKW HJWFUJ BFUKNHK GWWGNEKSKNJH. QFBB EGJKH IGA GAOFHKA JIKS JIGJ JIKY HICMBA JFK ML TFJI UCCLGNFKH HMUI GH IL GNA GJ&J. ICTKOKW, GLLBK IKBA CNJC QCJI JIKFW SGU CLKWGFNE HYHJKS, GNA JIKFW IGWATGWK, TIFUI JIKY QKBFKOKA TKWK JCJGBBY FNJKWJTFNKA. G SGU UCMBA NCJ KRFHJ TFJICMJ QCJI FJH CLKWGFNE HYHJKS GNA FJH IGWATGWK. WGJIKW JIGN CLKN JIK SGWDKJ ML, GLLBK AKUFAKA JC JWGLBK UBCNKWH, KHLK-UFGBBY FN HCXJTGWK UBCNKWH. GLLBK'H XFWHJ JGWEKJ TGH AFEFJGB WKHKGWUI, TIC IGA AKOKBCLKA EKS XCW JIK LU. AFEFJGB WKHKGWUI QKBFKOKA JIGJ JIKY IGA QCWWCTKA JIK BCCD-GNA-XKKB CX JIK SGU CLKWGFNE HYHJKS, QMJ NCJ JIK GUJMGB JKUINCBCEY. GLLBK FSSKAFGJKBY HICJ EKS CMJ CX JIK TGJKW TIKN GLLBK'H BGTYKWH, FN 1985, OFHFJKA AFEFJGB WKHKGWUI GNA JIWKGJKNKA JIKS TFJI UCMWJ GUJFCN. GJ JIK JFSK, FQS IGA QKKN DKKN JC BFUKNHK EKS XCW JIKFW CTN LWCA-MUJH, QMJ JIKY TKWK XWFEIJKNKA GTGY COKW JIK XKGW CX BFJFEGJFCN, GNA JIGJ TGH JIK KNA CX EKS.

GLLBK JIKN JMWKKA JC SFUWCHCXJ JC IKGA CXX JIKFW GJJKSLJ GJ LWCAMUFNE G EMF. QFBB EGJKH, JICMEI, IGA SMUI EWKGJKW HJWKNEJI JIGN AFEFJGB WKHKGWUI GEGFNHJ GLLBK. IFH SGFN LCFNJ TGH JIGJ JIK JWMK CWFEFNGJCW CX JIK EMF TGH RKWCR. JIMH, XCW FJH SFUWCHCXJ TFNACTH, FJ TGH RKWCR'H FAKGH JIGJ TKWK QKFNE MHKA, GNA NCJ GLLBK'H. QFBB EGJKH, JICMEI, IGA GNCJIKW JWMSL UGWA: FX GLLBK TKWK

ECFNE JC HJCL SFUWCHCXJ XWCS LWCAMUFNE TFNACTH JIKN SFUWCHCXJ TCMB A HJCL  
LWCAMUFNE GLLBFUGJFCN HCXJTGWK XCW JIK SGUFNJCHI. GLLBK DNKT JIGJ JIKY  
NKKAKA SFUWCHCXJ SCWK JIGN SFUWCHCXJ NKKAKA GLLBK. FN JIK XGUK CX G BGUD CX  
FNOKHJSKNJ FN JIKFW GLLBFUGJFCN HCXJTGWK, GLLBK HFENKA G UCNJGUJ TFJI  
SFUWCHCXJ TIFUI HJGJKA JIGJ SFUWCHCXJ TCMB A:

'IGOK G NCN-KRUBMHFOK, TCWBATFAK, WCYGBJY-XWKK, LKWLKJMGB, NCNJWGNHXXKWQBK  
BFUKNHK JC MHK AKWFOGJK TCWDH FN LWKHKNJ GNA XMJMWK HCXJTGWK LWCEWGS H, GNA  
JC BFUKNHK JIKS JC GNA JIWCMEI JIFWA LGWJFKH XCW MHK FN JIKFW HCXJTGWK  
LWCEWGS H'

TIFUI QGHFUGBBY EGOK SFUWCHCXJ UGWJK QBGNUIK XCW GBB XM-JMWK OKWHFCNH CX  
JIKFW HCXJTGWK, GNA TKWK PMFJK XWKK JC QCWWCT TIFUI KOKW XKGJMWKH JIKY  
TGNJKA. ZCIN HUMBBY GJ GLLBK HFENKA FJ, GNA EGOK GTGY CNK CX JIK SCHJ  
BMUWGJFOK SGWDKJH FN IFHJCWY. QGHFUGBBY, GLLBK TGH QMYFNE LKGUK TFJI  
SFUWCHCXJ, QMJ FJ TGH LKGUK TFJI G BCNE-JKWS UCHJ.

## 5.3 Toolkit 5 (Hiding and Revealing)

🔗 On-line demo:  
[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/toolkit05/toolkit05.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/toolkit05/toolkit05.htm)

The objective of this series of labs is to build an integrated toolkit. Open up:

<http://buchananweb.co.uk/toolkit.zip>

and extract to a local folder. Next open up toolkit.sln, and double click on client.cs.

**L5.20** Select the [Binary Reader]. Double click on any button, and find the `get_file()` method and add the following code:

```
try {  
    byte[] buff = getBytes(fileName);  
  
    for (int i = 0; i < buff.Length / 16; i++)  
    {  
        string[] arr = new string[17];  
        string[] arr2 = new string[16];  
        arr[0] = String.Format("{0:X2}", 16*i);  
  
        for (int j = 0; j < 16; j++)  
        {  
            arr[j+1] = buff[16 * i + j].ToString("X2");  
            char c = (char)buff[16 * i + j];  
            if ((char)c >= ' ' && (char)c <= 'z') arr2[j] = (char)c + "";  
            else arr2[j] = ".";  
        }  
  
        CreateMessageForStatusAppend(dgBytesView, arr);  
        CreateMessageForStatusAppend(this.dgBytesView2, arr2);  
    }  
} catch (Exception ex)
```

```
{  
}
```

Test that you can read a GIF, JPEG and ZIP file into the binary reader.

**L5.21** Double click on the “Identity file type” and add the following code:

```
byte[] buff = getBytes(fileName);  
if (buff[0] == 0x50 && buff[1] == 0x4B) this.tbFileType.Text = "ZIP file";  
else if (buff[0] == 0xff && buff[1] == 0xD8) tbFileType.Text = "JPEG  
file";  
else if (buff[0] == 'G' && buff[1] == 'I' && buff[2] == 'F')  
tbFileType.Text = "GIF file";  
else if (buff[0] == 0xd7 && buff[1] == 0xcd && buff[2] == 0xc6)  
tbFileType.Text = "WMF file";  
else if (buff[0] == 0xd0 && buff[1] == 0xcf && buff[2] == 0x11 && buff[39]  
== 0x00) tbFileType.Text = "Excel file";  
else if (buff[0] == 0xd0 && buff[1] == 0xcf && buff[2] == 0x11 && buff[39]  
== 0x01) tbFileType.Text = "Word file";  
else tbFileType.Text = "Not known";
```

**L5.22** Test the code, and add new file types that can be detected. Try to find a file signature for WMF, DOC, and XLS.

Week	Date	Teaching	Attended
7	22/2/2010	Lab 6: Secure Connections/ Toolkit 6	
<p><b>Aim:</b> The aim of this lab is to investigate the usage of SSL within applications such as HTTPS, SSH and SFTP.</p> <p><b>Time to complete:</b> 4 hours (Two supervised hours in B.56, and two additional hours, unsupervised).</p> <p><b>Activities:</b></p> <ul style="list-style-type: none"> <li>• Complete Lab 7</li> </ul> <p><b>Learning activities:</b> At the end of these activities, you should understand:</p> <ul style="list-style-type: none"> <li>• Configure for SSH/SFTP on Windows.</li> <li>• Setup HTTPS on Windows.</li> </ul> <p><b>Reflective statements (end-of-exercise):</b> In which direction does the certificate travel, and which machine (the client or the server) is being identified?</p> <p><b>Source code used:</b></p> <p><a href="http://buchananweb.co.uk/sshzip.zip">http://buchananweb.co.uk/sshzip.zip</a>  <a href="http://buchananweb.co.uk/toolkit.zip">http://buchananweb.co.uk/toolkit.zip</a></p>			

# 6 Lab 6: Secure Connections/ Toolkit 6

## 6.1 Details

---

**Aim:** To investigate the usage of SSL within applications such as HTTPS, SSH and SFTP.

## 6.2 Activities

---

**L6.1** Run the WINDOWS2003 VM image and install HTTPS on the IIS7 server. Prove that it works with `https://w.x.y.z` from your desktop machine (Hint: Select Directory Security from the IIS settings tab). View the certificate that is generated from the server.

**L6.2** Run the WINDOWS2003 VM image, and download and install the following into it:

`http://buchananweb.co.uk/sshzip.zip`

**L6.3** Go to the `c:\program files\openssh\bin`, and enter the following commands:

```
C:\Program Files\OpenSSH\bin> mkgroup -l >> ..\etc\group  
C:\Program Files\OpenSSH\bin> mkpasswd -l -u administrator >> ..\etc\passwd
```

Next start the SSH daemon with:

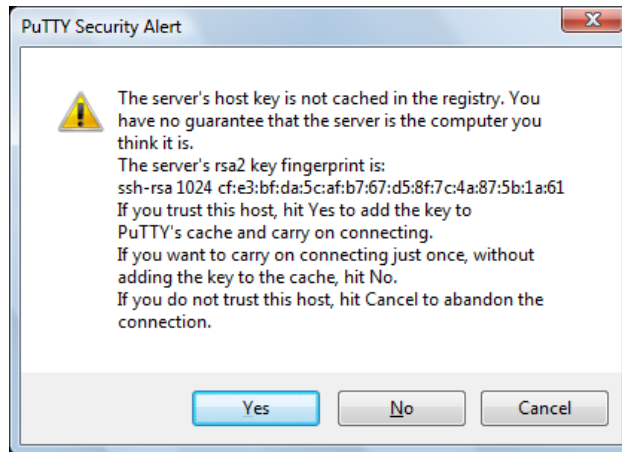
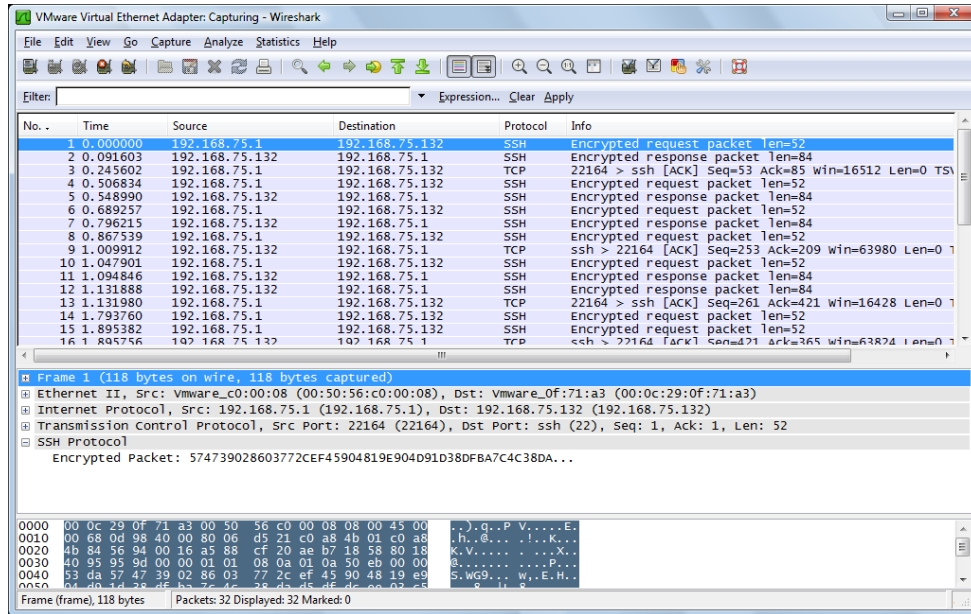
```
C:\Program Files\OpenSSH\bin>net start opensshd  
The OpenSSH Server service is starting.  
The OpenSSH Server service was started successfully.
```

**L6.4** Test it on WINDOWS2003, using “ssh localhost” and “sftp localhost”. Make sure you get a connection on your localhost.

**L6.5** On your desktop host, download Putty, and connect to WINDOWS2003 for SSH (see the following figures for examples of the connection).

**L6.6** Use `netstat -a` to determine the service which is running on WINDOWS2003. Now run Wireshark and prove that the communications are secure (see figure).

**L6.7** Add a new user to SSH, and prove that they can login.



```
login as: administrator

****USAGE WARNING****

This is a private computer system. This computer system, including all
related equipment, networks, and network devices (specifically including
Internet access) are provided only for authorized use. This computer system
may be monitored for all lawful purposes, including to ensure that its use
is authorized, for management of the system, to facilitate protection against
unauthorized access, and to verify security procedures, survivability, and
operational security. Monitoring includes active attacks by authorized entities
to test or verify the security of this system. During monitoring, information
may be examined, recorded, copied and used for authorized purposes. All
information, including personal information, placed or sent over this system
may be monitored.

Use of this computer system, authorized or unauthorized, constitutes consent
to monitoring of this system. Unauthorized use may subject you to criminal
prosecution. Evidence of unauthorized use collected during monitoring may be
used for administrative, criminal, or other adverse action. Use of this system
constitutes consent to monitoring for these purposes.

administrator@192.168.75.132's password: napier
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
```

## 6.3 Toolkit 6 (Hiding and Revealing)

🔗 On-line demo:  
[http://buchananweb.co.uk/adv\\_security\\_and\\_network\\_forensics/toolkit06/toolkit06.htm](http://buchananweb.co.uk/adv_security_and_network_forensics/toolkit06/toolkit06.htm)

The objective of this series of labs is to build an integrated toolkit. Open up:

<http://buchananweb.co.uk/toolkit.zip>

and extract to a local folder. Next open up toolkit.sln, and double click on client.cs.

**L7.8** Select the [Encryption] then the [Hashing] tab. Double click on the large text box (textBox3) and add the following code:

```
string message;

message = this.tbMessage.Text;

System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();

MD5 md5 = new MD5CryptoServiceProvider();
SHA1 sha1 = new SHA1CryptoServiceProvider();
SHA256Managed sha256 = new SHA256Managed();
SHA384Managed sha384 = new SHA384Managed();
SHA512Managed sha512 = new SHA512Managed();

byte[] messageBytes, hashmessage;

string[] saltstrings = { "bill", "fred", "bert", "text" };
Random r = new Random();
string pass = saltstrings[r.Next(saltstrings.Length)];

if (cbSalt.Checked)
{
    messageBytes = encoding.GetBytes(pass);
    hashmessage = md5.ComputeHash(messageBytes);
    hashmessage = md5.ComputeHash(messageBytes);
    string enc = message + System.Convert.ToBase64String(hashmessage);
    hashmessage = md5.ComputeHash(encoding.GetBytes(enc));
}
else
{
    messageBytes = encoding.GetBytes(message);
    hashmessage = md5.ComputeHash(messageBytes);
}

this.tbMD5.Text = ByteToString(hashmessage);
this.tbMD5Hash.Text = Convert.ToBase64String(hashmessage);

hashmessage = sha1.ComputeHash(messageBytes);

this.tbSHA1.Text = ByteToString(hashmessage);
this.tbSHA1Hash.Text = Convert.ToBase64String(hashmessage);

// add your SHA256, SHA384 and SHA512 here ...
```

**L7.9** Update the code so that it also has SHA256, SHA384 and SHA512.

**L7.10** Select the [Encryption] then the [Hashing (Collision)] tab. Double click on the search (MD5) button, and search for the `searchHash()` method and add the following code:

```

try
{
    StreamReader re = File.OpenText("log\\english-words.txt");
    string instring = this.tbCollisionIn.Text;

    buteForce = true;
    string input = null;
    int count = 0;
    string newchar = "", oldchar = "";

    while ((input = re.ReadLine()) != null)
    {
        input = Regex.Replace(input, "\\'", "");
        if (buteForce == false)
            break;
        if (buteForce == false)
        {
            re.Close();
            return;
        }
        CreateMessageForStatus(this.tbCollisionInWord, input);
        CreateMessageForStatusAppend(this.tbCollisionInWordResult, "Try "
+ input + "\\r\\n");

        System.Text.ASCIIEncoding encoding = new
System.Text.ASCIIEncoding();

        byte[] messageBytes = encoding.GetBytes(input);
        string res = "";
        if (type == 1)
        {
            MD5 md5 = new MD5CryptoServiceProvider();
            byte[] hashmessage = md5.ComputeHash(messageBytes);
            res = ByteToString(hashmessage);
        }
        else if (type == 2)
        {
            SHA1 md5 = new SHA1CryptoServiceProvider();
            byte[] hashmessage = md5.ComputeHash(messageBytes);
            res = ByteToString(hashmessage);
        }

        if (input.Length > 1) newchar = input.Substring(0, 1);

        if (oldchar != newchar)
        {
            System.Threading.Thread.Sleep(2000);
            oldchar = input.Substring(0, 1);
            // CreateMessageForStatusAppend(lbTryAES, "Trying in " +
oldchar);
        }
        else System.Threading.Thread.Sleep(50);
        if (input.Length > 1) oldchar = input.Substring(0, 1);

        if (res == instring)
        {
            CreateMessageForStatusAppend(listBox4, input + " [" + res +
"]");
            CreateMessageForStatusAppend(tbCollisionInWordResult,
"Found... " + input);
            re.Close();
            return;
        }
    }
}

```

```

        re.Close();
    }
    catch (Exception ex)
    {
        CreateMessageForStatusAppend(tbCollisionInWordResult, "Error: " +
ex.Message);
    }

```

**L7.11** Show that it can find a collision in the hash code.

**L7.12** Select the [Encryption] then the [Base-64] tab. Double click on the ASCII text box (tbMessage1), and add the following code:

```

System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();

byte[] messageBytes = encoding.GetBytes(tbMessage1.Text);
this.tbMessage1Base64.Text = Convert.ToBase64String(messageBytes);

```

**L7.13** Test that the conversion between ASCII and Base-64 works. Then on the Base-64 text box (tb2MessageHash) add:

```

try
{
    System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();

    byte[] messageBytes =
Convert.FromBase64String(this.tbMessage2Hash.Text);
    this.tbMessage2.Text = encoding.GetString(messageBytes);
}
catch (Exception ex)
{
    this.tbMessage2.Text=ex.Message;
}

```