Week	Date	Teaching	Attended
8	**01/3/2010**	Lab 7: Web Authentication/ Toolkit 7 (URL Cache)	

Aim: The aim of this lab is to investigate the integration of SAML into Web Authentication.

Time to complete:
4 hours (Two supervised hours in B.56, and two additional hours, unsupervised).

Activities:

- **Complete Lab 7**: Data Web Authentication/ Toolkit Development 7
  .pdf from WebCT or http://www.dcs.napier.ac.uk/~cs342/CSN10102/Lab7.pdf
  (Use **Web Infastructures** from **Handbook** as reference while completing the lab)

- Take **End of Unit Tests** for unit5:
  http://buchananweb.co.uk/adv05.html

  Note: The module Handbook is available at:
  http://buchananweb.co.uk/adv/part2.pdf

Learning activities:
At the end of these activities, you should understand:

- Use OpenID to authenticate for a Web site.
- Configure for IIS Web Server for SAML integration.
- View URL cache within a toolkit.

Reflective statements (end-of-exercise):
What advantages do software tokens have over usernames and passwords?

What are the drawbacks of using software tokens for authentication, and how would they be overcome?

Source code used:
http://buchananweb.co.uk/ wwwroot.zip
http://buchananweb.co.uk/toolkit.zip

Lab 7: Web Authentication/ Tookit 7

Details

Aim: To investigate the usage of OpenID and SAML for Web authentication.

Activities

L7.1 Go to:

https://www.myopenid.com/

and create your new OpenID account. Next find some Web sites to login with using your new account.

☞ Write down your OpenID account, and try to find a few sites which support OpenID and log into them.

L7.2 Download, and unpack the following web site code:

http://buchananweb.co.uk/openid.rar

Run **Visual Studio** and open the web site, using **File->Open->Web Site...**

Add a new web page called **Success.aspx**, using **Website->Add new Item... (Web Form)** and add some text "Successfully Logged In". Then add the highlighted code behind the Default.aspx page:

```
protected void Page_Load(object sender, EventArgs e)
{
    OpenIdData data = OpenID.Authenticate();
    if (data.IsSuccess)
    {
        Response.Redirect("success.aspx");
    }
}
protected void Button1_Click(object sender, EventArgs e)
{
    bool success = OpenID.Login(txtOpenId.Text, "email,fullname",
 "country,language");
}
```

Prove that you can login with your **OpenID identity**. Enter your myOpenID url into the text box, and you should be prompted to sign in as shown in Figure L7.2.1

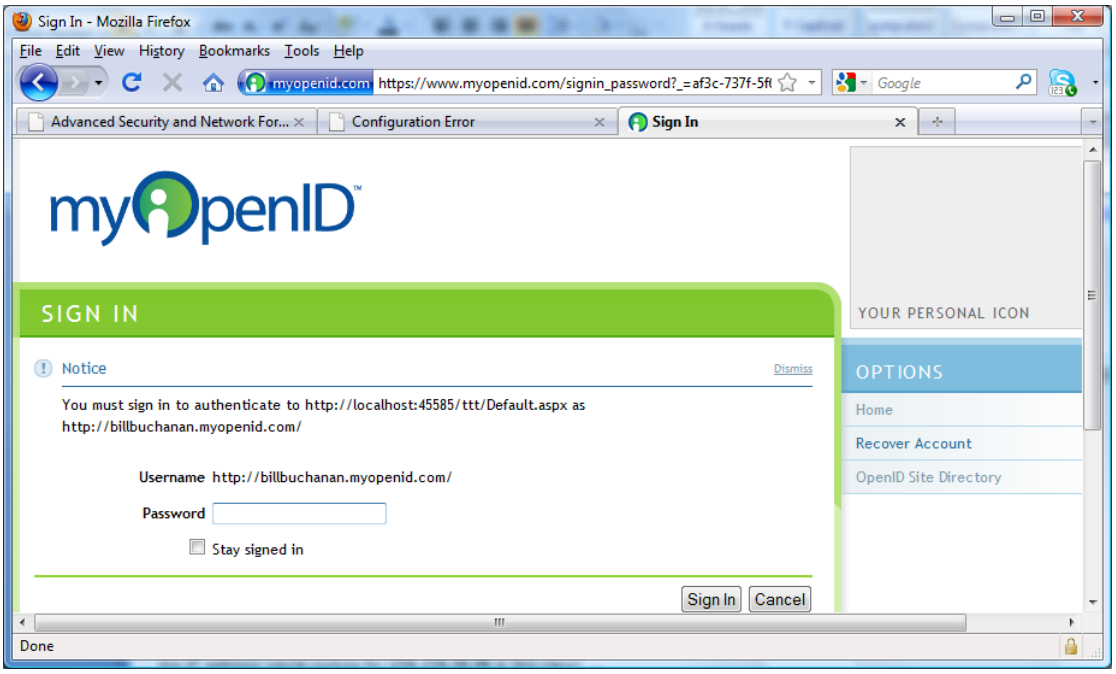


Figure L7.2.1 myOpenID Sign In Page

L7.3 Run the WINDOWS2003 VM image and Download the following to the c:\inetpub\wwwroot**test** folder:

	http://buchananweb.co.uk/wwwroot.zip

L7.4 Go into the IIS Manager and right click on the test folder (Figure L7.4.1), and set it up with an Application Name (Figure L7.4.2).

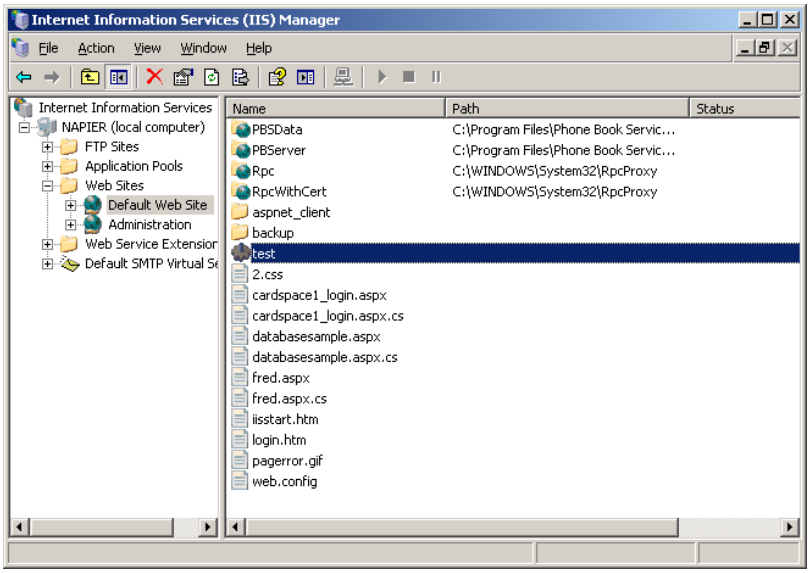


Figure L7.4.1: IIS Manager

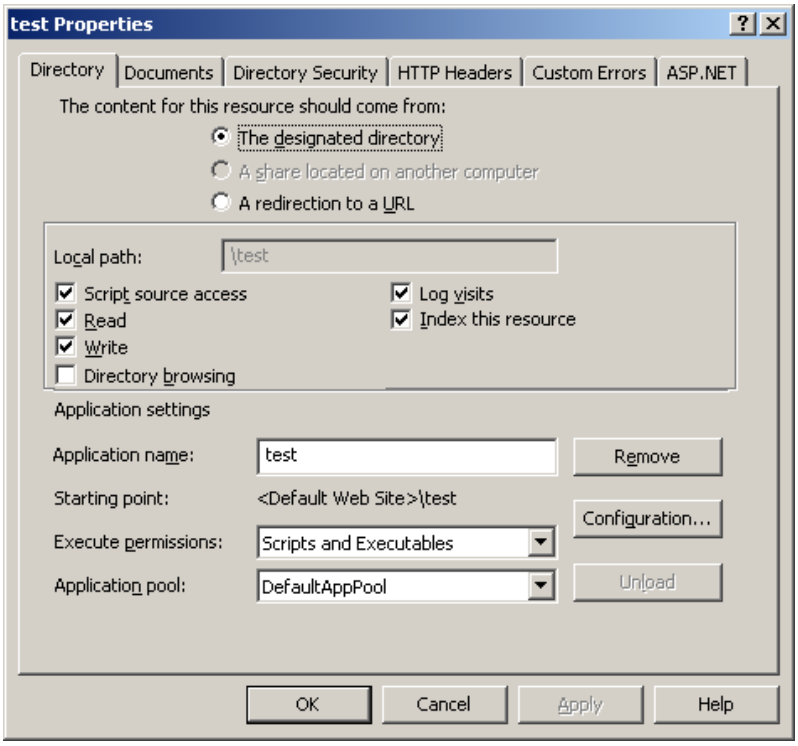


Figure L7.4.2: test Properties

L7.5 Next run Visual Studio 2008, and select Open Web site and navigate to c:\inetput\wwwroot\test.

L7.6 Next select sample1.htm, and add the following code:

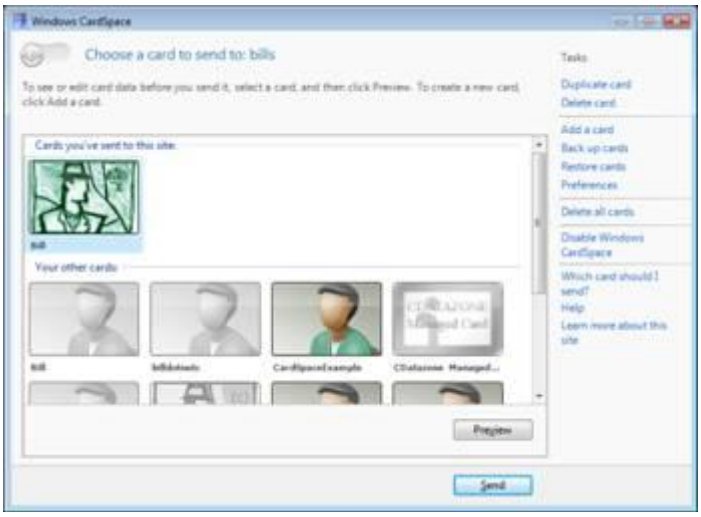
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>Sample 1</title>
</head>
<body>
<form id="form1" method="post" action="cardspace1_login.aspx">
<div>
<button type="submit">Click here to sign in with your Information Card</button>
<object type="application/x-informationcard" name="xmlToken">
<param name="tokenType" value="urn:oasis:names:tc:SAML:1.0:assertion" />
<param name="requiredClaims"
value="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier" />
</object>
</div>
</form>

</body>
</html>
```

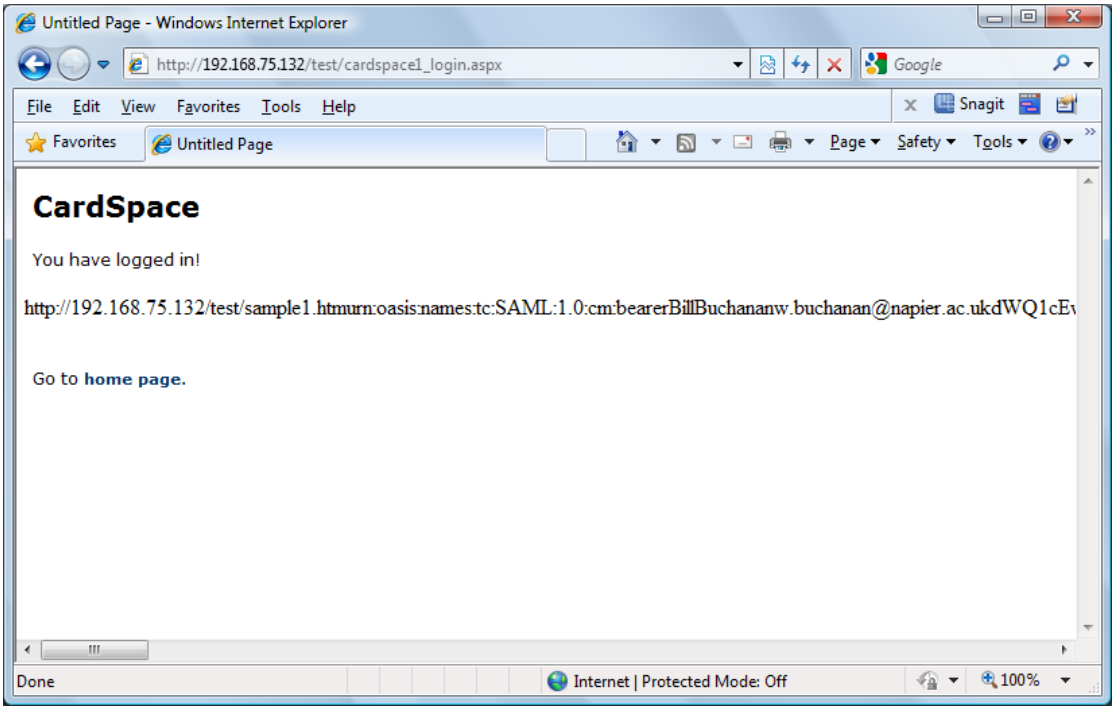
L7.7 Next select cardspace1_login.aspx.cs, and add the highlighted code:

```
protected void Page_Load(object sender, EventArgs e)
{
Label1.Text = Request.Params["xmlToken"];
}
```

L7.8 Next load https://localhost, and select the first example (sample1.htm). Select your card (or create one), and login, such as:



L7.9 Next login remotely from your desktop into the virtual image, such as with:



L7.10 Next select sample2.htm, and add the following code:.

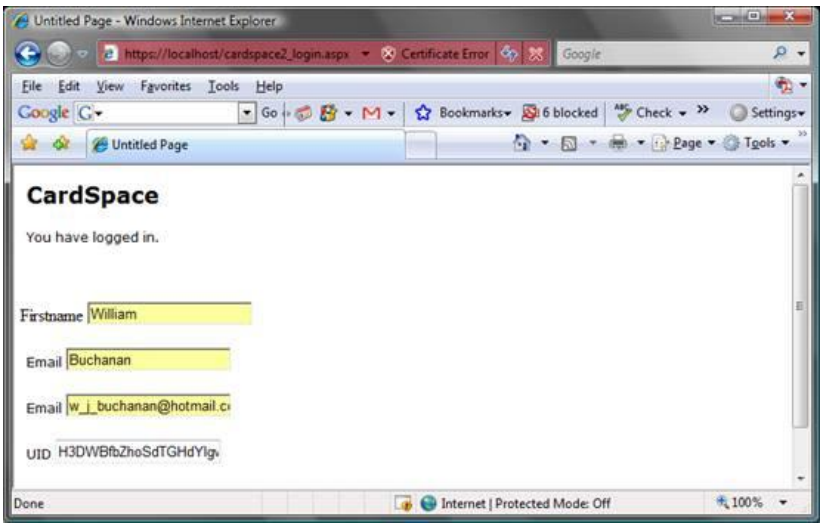
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>Authenticate</title>
<object type="application/x-informationcard" name="_xmlToken">
<param name="tokenType" value="urn:oasis:names:tc:SAML:1.0:assertion" />
<param name="requiredClaims"
value="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier" />
```

```
</object>
<script language="javascript">
function GoGetIt()
{
var xmltkn=document.getElementById("_xmltoken");
var thetextarea = document.getElementById("xmltoken");
thetextarea.value = xmltkn.value ;
}
</script>
</head>
<body>
<form id="form1" method="post" action="cardspace2_login.aspx">
<div>
<button name="go" id="go" onclick="javascript:GoGetIt();">Click here to get the
token.</button>
<button type="submit">Click here to send the card to the server</button>
<textarea cols=100 rows=20 id="xmltoken" name="xmlToken" ></textarea>
</div>
</form>
</body>
</html>
```

L7.11 Next select cardspace2_login.aspx.cs, and add the highlighted code:

```
protected void Page_Load(object sender, EventArgs e)
{
string xmlToken;
xmlToken = Request.Params["xmlToken"];
if (xmlToken == null || xmlToken.Equals(""))
{
// ShowError("Token presented was null");
}
else
{
Token token = new Token(xmlToken);
firstname.Text = token.Claims[ClaimTypes.GivenName];
surname.Text = token.Claims[ClaimTypes.Surname];
email.Text = token.Claims[ClaimTypes.Email];
uid.Text = token.UniqueID;
}
}
```

Next show that the Web site now displays the details from the card, such as:



L7.12 Export the card you have created, and view its contents. Now import it into WINDOWS2003.

Toolkit 7 (URL cache)

On-line demo:
http://buchananweb.co.uk/adv_security_and_network_forensics/toolkit07/toolkit07.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 C# solution file **toolkit.sln**, and double click on **client.cs**.

L7.1 Add a new tab named [**OS**], and add another tab into this tab (see Figure L7.3). Next add two DateTimePickers (dtStart and dtEnd), two buttons, and two datagridviews (dgURLCache and dgFileCache). Add the following code on the Show History button:

```
Showhistory();
```

And the method:

```
using UrlHistoryLibrary;

        public void Showhistory()
        {
            this.dgURLCache.Rows.Clear();
            this.dgFileCache.Rows.Clear();
            urlHistory = new UrlHistoryWrapperClass();
            enumerator = urlHistory.GetEnumerator();
            list = new ArrayList();


            GetHistoryItems();

            list.Reverse();


            if (textBoxFilter.Text != "")
            {
                enumerator.SetFilter(textBoxFilter.Text, STATURLFLAGS.STATURLFLAG_ISTOPLEVEL);
            }
            foreach (STATURL u in list)
            {
                string[] url = new string[2];

                url[0] = Convert.ToString(u.LastVisited);
                url[1] = u.URL;
                STATURL u1 = (STATURL)list[0];

                if (u.LastVisited >= dtStart.Value && u.LastVisited <= dtEnd.Value)
                {
                    u1 = (STATURL)list[list.Count - 1];

                    if (url[1].StartsWith("http")) this.dgURLCache.Rows.Add(url);
                    else if (url[1].StartsWith("file")) this.dgFileCache.Rows.Add(url);
                }
            }

            GC.Collect();

        }
```

L7.2 Test that the program can view the URL history. Next add the following code to the Clear URL History button:

```
            DialogResult rtn=MessageBox.Show("Are you sure you want to delete all your URL
history?","URL History",MessageBoxButtons.YesNo);
            if (rtn == DialogResult.Yes)
                urlHistory.ClearHistory();
```

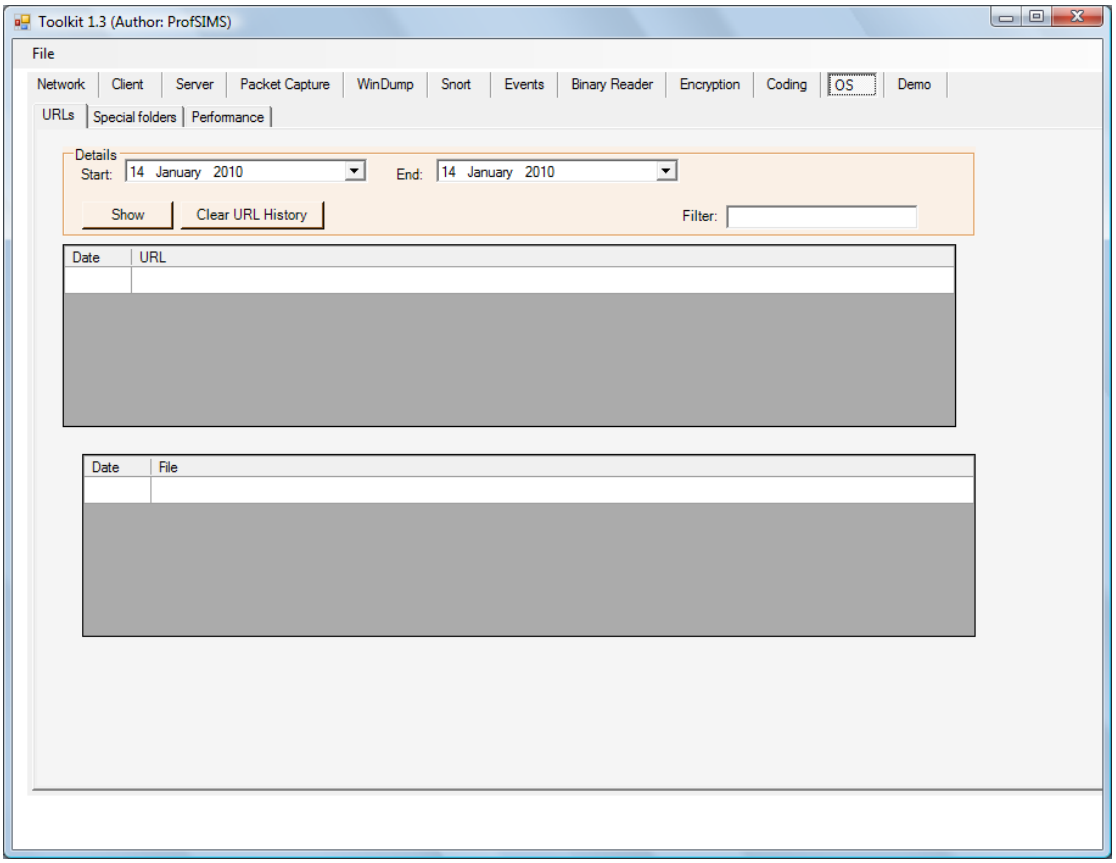


Figure L7.3

L7.3 Test the program for its operation.

L7.4 If you have time, investigate the "Special Folders" tab (see Figure L7.3), such as with the following code:

```
            DirectoryInfo d = new
DirectoryInfo(System.Environment.GetFolderPath(Environment.SpecialFolder.Recent));
            ShowFiles(dgFilesRecent, d.FullName);


    public void ShowFiles(DataGridView dg,  string folder)
        {
            try
            {
                dg.Rows.Clear();
                string[] files = Directory.GetFiles(folder);
                CreateMessageForStatus(tbFiles, folder);
                foreach (string s in files)
                {

                    string filename = s;
                    FileInfo f = new FileInfo(filename);
                    string[] s1 = new string[2];
                    s1[0] = Convert.ToString(f.LastAccessTime);
                    s1[1] = s;
                    CreateMessageForStatusAppend(dg, s1);
                }
```

```
            }
            catch (Exception ex)
            {
            }
        }
```

This code allows the user to view the "Recent" special folder. If you get this code to work, try and view the other "Special folders".