

Lab 12: Role-based security

The Microsoft .NET environment now offers an excellent alternative to Java in producing portable and secure code. It uses a role-based approach for user authentication, with the **WindowsIdentity** class, where the `GetCurrent()` method can be used to get the current user. The **WindowsPrincipal** class can then be used to apply the role. For example to test if the user is an administrator:

```
using System;
using System.Security;
using System.Security.Principal;

namespace ConsoleApplication3
{
    class Class1
    {
        static void Main(string[] args)
        {
            WindowsIdentity myID = WindowsIdentity.GetCurrent();

            System.Console.WriteLine("Your ID: " + myID.Name);
            System.Console.WriteLine("Authentication: " +
                                     myID.AuthenticationType);

            WindowsPrincipal myPrin = new WindowsPrincipal(myID);

            if (myPrin.IsInRole(WindowsBuiltInRole.Administrator))
                System.Console.WriteLine("You're an Administrator ");
            else
                System.Console.WriteLine("You're not an Administrator");

            Console.ReadLine();
        }
    }
}
```

A sample run gives:

```
Your ID: BILLS\William Buchanan
Authentication: NTLM
You're an Administrator
```

Other roles are also defined, such as:

```
WindowsBuiltInRole.Guest
WindowsBuiltInRole.PowerUser
WindowsBuiltInRole.User
```

Next we could apply this security to only allow an administrator to view the IP address of the computer, with:

```
using System;
using System.Security;
using System.Security.Principal;
using System.Net;
```

```

namespace ConsoleApplication3
{
    class Class1
    {
        static void Main(string[] args)
        {
            WindowsIdentity myID = WindowsIdentity.GetCurrent();

            System.Console.WriteLine("Your ID: " + myID.Name);
            System.Console.WriteLine("Authentication: " +
                myID.AuthenticationType);

            WindowsPrincipal myPrin = new WindowsPrincipal(myID);

            if (myPrin.IsInRole(WindowsBuiltInRole.Administrator))
            {
                string strHostName = Dns.GetHostName();

                IPHostEntry ipEntry = Dns.GetHostByName(strHostName);

                IPAddress [] addr = ipEntry.AddressList;
                System.Console.WriteLine("IP: " + addr[0]);
            }
            else
                System.Console.WriteLine(
                    "Sorry ... you have no permissions for this");
        }
    }
}

```

Run this program, and view the output.