# Asp.Net Security

Jim Fawcett

CSE686 – Internet Programming

Summer 2005

# Security Model

- Authentication
  - Who do you say you are?    User id
  - Do you have proof?    Password
- Authorization
  - Do you have the priviledges to do a requested action?

# Asp.Net Authentication

- Asp.Net directly supports three models:
  - Authentication mode = None
    - Application supplied security
  - Authentication mode = Windows
    - Based on Windows Accounts
    - Suitable only for local network
  - Authentication mode = Forms
    - Manged by application with support for redirection and accessing identities provided by Asp.Net
  - Authentication mode = PassPort
    - Authentication credentials stored on Microsoft server
    - Sites license the service

# No Asp Supplied Authentication

- Asp.Net allows all users access to all asp pages
- It is up to the application to provide authentication and authorization
- Authentication and Role-based access provided by user control(s).
  - Application uses session to tell if user is logged in.
  - User signs in and is assigned roles from database by user control.
  - Access to pages based on roles.
  - No help from Windows doing this.

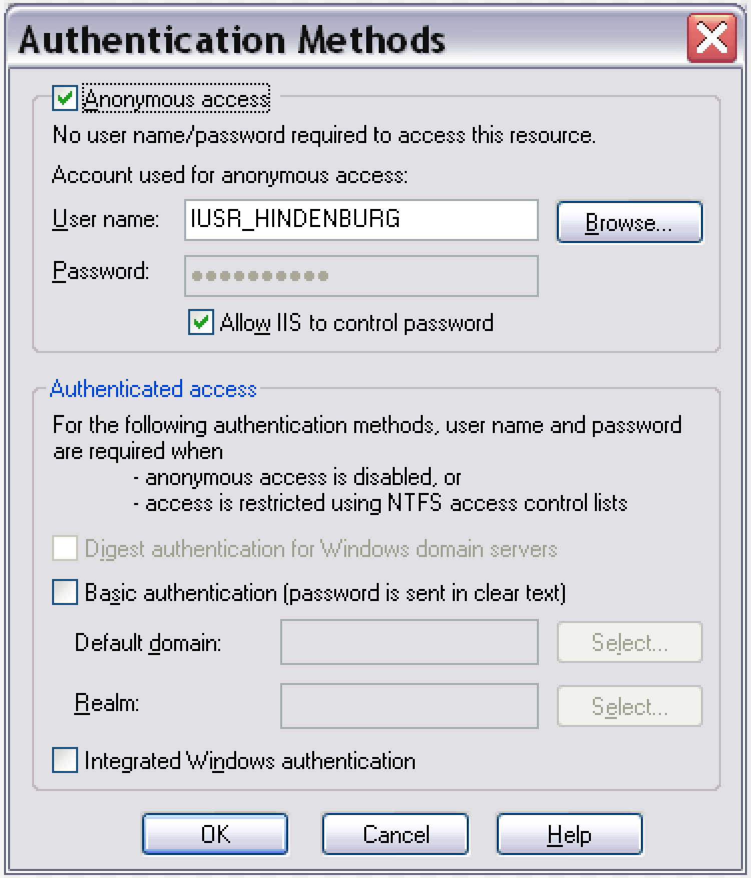
# No Authentication

- Virtual directory allows anonymous access
- Web.Config file specifies:

```
<authentication mode=“None”/>
<authorization>
  <allow users=“*”/>
 </authorization>
```

- Its up to application to provide authentication
- CSE686 Labs have encouraged you to build authenticating control and provide your own redirections.

# Security Settings for None



☑ Anonymous access

No user name/password required to access this resource.

Account used for anonymous access:

User name: IUSR_HINDENBURG

Password: ●●●●●●●●●●●

☑ Allow IIS to control password

Authenticated access

For the following authentication methods, user name and password are required when

- anonymous access is disabled, or
- access is restricted using NTFS access control lists

□ Digest authentication for Windows domain servers

□ Basic authentication (password is sent in clear text)

Default domain:

Realm:

□ Integrated Windows authentication

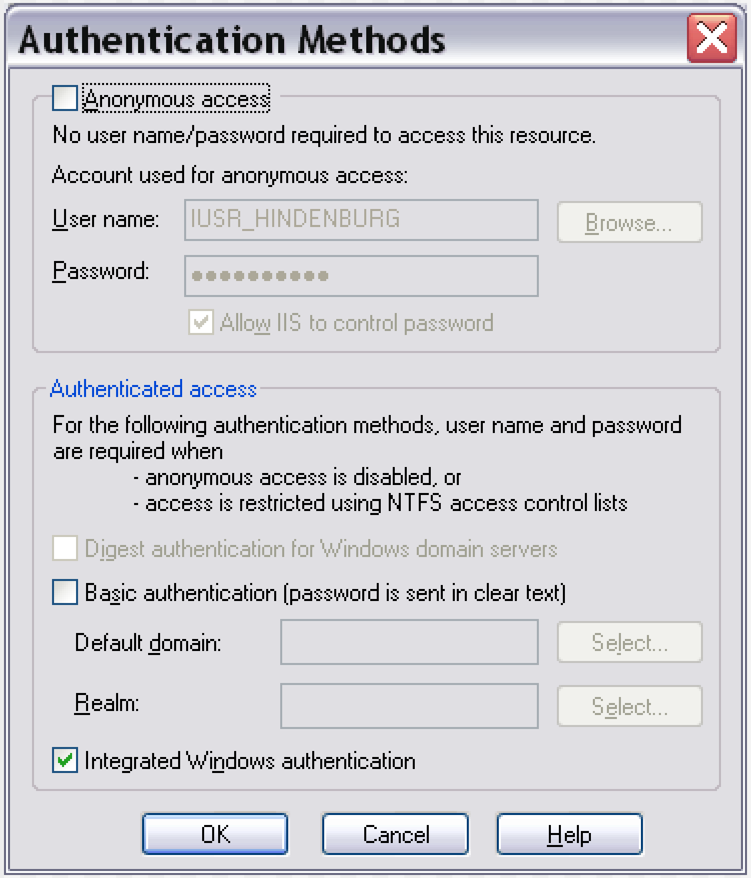
# Windows Authentication

- Uses custom socket ports, as well as port 80, so won’t go through firewalls.
- Requires all users to have Windows accounts on server.
- Suitable only for site serving a local network.
- Remote access requires operation in a domain or Active Directory with Kerberos: http://support.microsoft.com/default.aspx?scid=kb;en-us;324276 http://support.microsoft.com/default.aspx?scid=kb;en-us;810572

# Windows Authentication

- The major advantage of Windows Integrated Authentication is that you can use all of the Windows role-based security mechanisms.
- It’s easy to restrict access to a page to one or more roles and roles can be configured with specific permissions.

# Security Settings for IWA



Authentication Methods

☐ Anonymous access

No user name/password required to access this resource.

Account used for anonymous access:

User name: IUSR_HINDENBURG

Password: ●●●●●●●●●●●

☑ Allow IIS to control password

Authenticated access

For the following authentication methods, user name and password are required when
- anonymous access is disabled, or
- access is restricted using NTFS access control lists

☐ Digest authentication for Windows domain servers

☐ Basic authentication (password is sent in clear text)

Default domain:

Realm:

☑ Integrated Windows authentication

# Forms Authentication

- Application provides login page.
- Asp.Net takes care of redirections.
- Application provides id and password storage and retrieval.
- Almost no help with role-based access.
- Can configure directories, using web.config files to accept or deny non-authenticated users:
  - <deny users=‘?’/>  // anonymous users
  - <allow users=‘*’/>  // allow all others

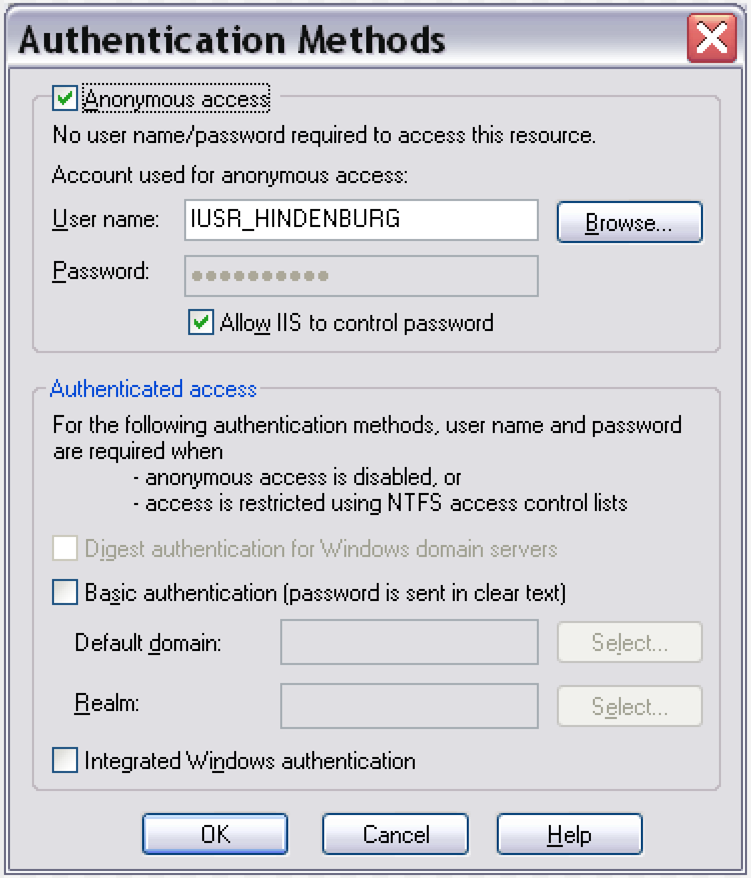
# Forms Authentication

- Virtual directory allows anonymous access
- Web.Config file specifies:

```
<authentication mode="Forms"/>
   <forms loginUrl="login.aspx">
      <credentials … />
   </forms>
</authentication>
<authorization>
   <deny users="?"/>
</authorization>
```

- Application provides login.aspx which uses System.Web.Security.FormsAuthentication to redirect after authentication.
- Application uses database to store and retreive user ids and passwords.
- Can logout using FormsAuthentication.SignOut();

# Security Settings for Forms



Authentication Methods

☑ Anonymous access

No user name/password required to access this resource.

Account used for anonymous access:

User name: IUSR_HINDENBURG

Password: ●●●●●●●●●●●

☑ Allow IIS to control password

Authenticated access

For the following authentication methods, user name and password are required when

- anonymous access is disabled, or
- access is restricted using NTFS access control lists

□ Digest authentication for Windows domain servers

□ Basic authentication (password is sent in clear text)

Default domain:

Realm:

□ Integrated Windows authentication

# Cardspace (Passport) Authentication

- Fee-based service provided by Microsoft
- Won’t be discussed further

# Role-Based Security without Windows

- Public web sites will almost certainly use Application supplied or Forms based authentication.
- Clients will not have a user account on the server, so Windows role-based security is no help.
- The site may need to define at least simple roles:
  - New user
  - Registered user
  - Premium member

# Role-Based Authorization

- So how do ***you*** provide role-base access?
  - At login, retrieve user’s roles from db and store in session.
  - Provide control on each page that specifies allowed roles.
  - OnPageLoad, check user roles from session against allowed roles from control.
  - Probably easiest to do this with custom authentication but workable with Forms Auth.
  - Would help to have an administrator’s page to add users and define roles and role membership.

# Security Issues

- Authentication √
  - Who are you?
- Authorization √
  - What are you allowed to access?
- Confidentiality
  - Hiding content in volatile environment
- Integrity
  - Detecting modification

# Encrypted Channel with SSL

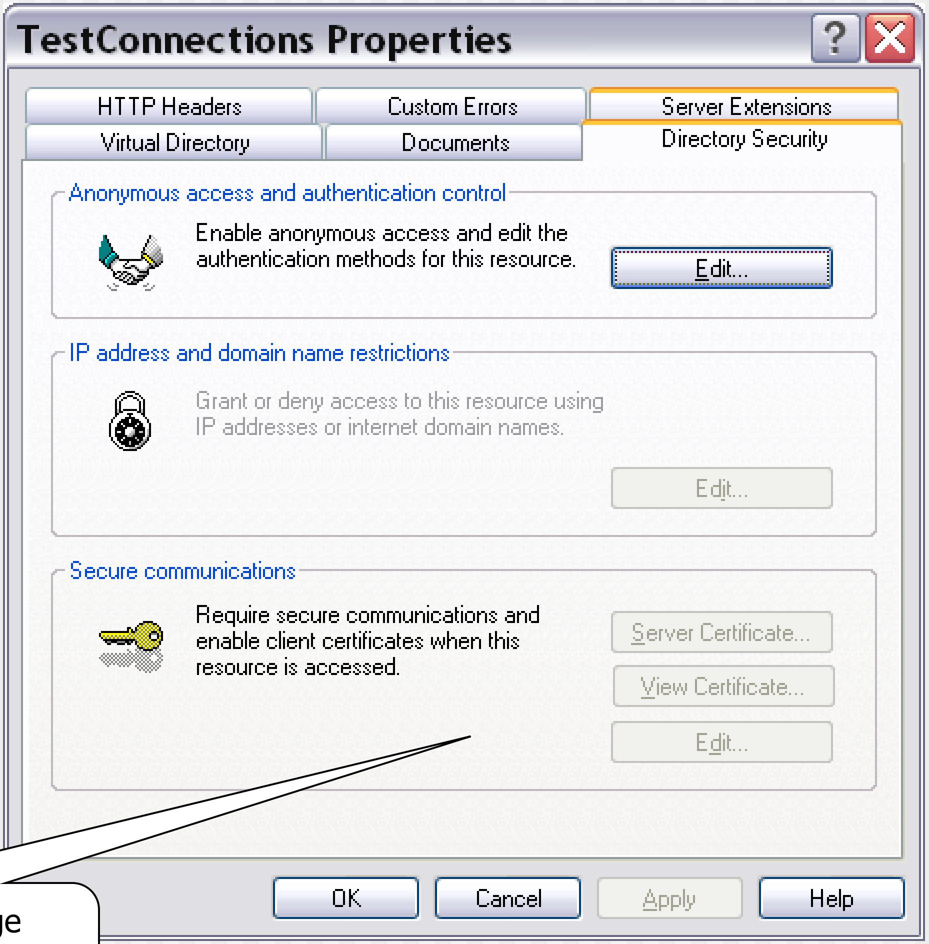
- Secure Sockets Layer provides an encrypted channel for transmitting sensitive data.
  - Recognized by most browsers.
  - Used by all the major sites: Amazon, …
  - Uses 128 bit encryption.

# Secure Sockets Layer (SSL)

- Requires third party certificate
  - You generate a certificate request file using web server certificate wizard.
  - Send to certificate authority, Verisign, … along with a check for $349 (renewed each year for $249).
  - Wait for about three weeks.
  - Install the certificate using the web server certificate wizard.
  - You can generate certificates used only for development.

# Requiring SSL

- SSL is invoked whenever the url prefix is https.
- You can force users to use SSL by setting directory properties.



TestConnections Properties

Anonymous access and authentication control

Enable anonymous access and edit the authentication methods for this resource.

IP address and domain name restrictions

Grant or deny access to this resource using IP addresses or internet domain names.

Secure communications

Require secure communications and enable client certificates when this resource is accessed.

Virtual directory properties page allows you to require SSL if you have installed a certificate.

# Using .Net Encryption

- You may need to encrypt password files or other sensitive information stored on your site.
- System.Security.Cryptography
  - Public Key (asymmetric) algorithms
    - DSA – DSACryptoServiceProvider
    - RSA – RSACryptoServiceProvider
  - Private Key (symmetric) algorithms
    - DES – DESCryptoServideProvider
    - Triple DES, RC2, Rijndael

# Using .Net Hashing

- You may need to ensure that messages or files have not been tampered with.
- System.Security.Cryptography
  - 128 Bit Hash
    - MD5 – MD5CryptoServiceProvider class.
  - 160 Bit Hash
    - SHA1 – SHA1CryptoServiceProvider

# References

- Asp Applications & Authentication
  - Programming .Net, Jeff Prosise, Microsoft Press, 2002
- Applications, Authentication, SSL
  - ASP.NET Unleased, Second Edition, Stephen Walther, SAMS, 2004