ASP.NET Web Application Security

Hannes Preishuber
ppedv AG
HannesP@ppedv.de

Classic rules

- Passwords
  - encrypted
  - Min. length & case sensitive & unusual
    - Dictionary attack
- Sniffers
  - Men
  - Trojan
  - Network
- Not limited to Microsoft!

ASP .NET Features

- Authentication
  - IIS, ASP.NET
    - ASP.NET: Forms, Windows, Passport, Default, and Custom
- Authorization
  - Access to Directories, Files
- Role-Based Security
  - if User.IsInRole("Admin")
- Impersonation
  - Code and User

Authentication

- ASP.NET is an ISAPI extension
  - Only receives requests for mapped content
- Windows Authentication (via IIS)
  - Basic, Digest, NTLM, Kerberos, Certificate Support
  - Leverages platform authentication
- Forms-based (Cookie) Authentication
  - Application credential verification
- Supports Microsoft® Passport Authentication
- Custom Authentication

Forms-Based Authentication

- Easy to implement
  - ASP.NET provides redirection
- Steps
  - Configure IIS to allow anonymous users (typically)
  - Configure ASP.NET cookie authentication
  - Write your login page
- Secures not all
  - Only Files with named extensions

Forms Auth Configuration

```
<authentication mode= "Forms">
  <forms
        name=".ASPXAUTH"
         loginUrl="login.aspx"
         protection="all"
         timeout="30"
         path="/"
     />
</authentication>
```

Risk

- Authentication Data
  - Username
    - Shown in web pages
  - Password
- Authentication Flow
  - HTTP is clear text
    - use SSL
  - ASP.NET to Database is clear text
    - Store hashed passwords

Show

Risk Cookieless

- Sends Session ID in Query String
- Web.Config
  - <sessionState cookieless="true"
- Session lives 20 minutes
  - From last activity
- Attach on Session
  - public terminal
  - Sniffer
- Also for HTTP Headers and Cookies

Show

Config Topics

- Machine.config
  - System.Web.HttpForbiddenHandler
  - <processModel
    - userName=“machine"
- Web.Config
  - <customErrors mode="On" />
  - Encrypt Connection Strings
- HttpOnly
  - Client side script

Show

SQL Injection

- How Web pages works?
- INPUT rendered from Textbox Web Control
- Query String
- Use values concat a SQL command
  - Search knowledge base
  - Paged results
  - Look for specific record
    - User credentials

What really exists!

- DON'T *LIKE*
  - More comfort for the user

```
string sql = "select * from KB where
        content like '" + search.Text + "'
```

- Hacker types: %

```
string sql = "select * from KB where
        content like '%'
```

- User authentication!

SQL Injection Attack

- Developer concate SQL statements

```
string sql = "select * from Users where
        user ='" + User.Text + "'
        and pwd='" + Password.Text + "'"
```

- Hacker types: ‘ or 1=1 --‘

```
string sql = "select * from Users where
        user =''or 1=1 --' and pwd=''"
```

- Result is the first database entry
  - Maybe the Admin

Show

SQL Injection Attack

- Take over control
- User types: ; xp_cmdshell 'format c: /q /yes '; drop database myDB; --

```
select * from tabelle where id=1;
xp_cmdshell 'format  c: /q /yes ';
 drop database myDB; --
```

- Result: Hacker can do everything
  - SQL process runs with system privileges

SQL Injection Attack

- Never use “sa”
  - Default blank password
  - Hacker knows a lot about sa
  - Trusted Security
  - Application user
    - Only with needed access rights
- Storing Connection Strings
  - Web.Config
    - Hashed not clear text
  - error case source code is often visible

Best Tip

- Use parameterized Select

```
sql = "select * from Users where
      user = @user and pwd = @pwd";
SqlCommand cmd = new SqlCommand(sql,con);
cmd.Parameters.Add("@user",User.Text);
cmd.Parameters.Add("@pwd",Password.Text);
```

- Use Stored Procedures
- Cookie & URL Injection

Show

Cross site-scripting

- User Input is stored in Database
- Database content is presented
- Injection of
  - HTML code
  - JScript code
- A different denial of service

```
<script>
```

- Redirect the user to dialer page

```
<script language=Jscript>
window.navigate('net.htm');</script>
```

Cross site-scripting

- Don’t trust the user
  - Use validators controls
  - Use regexp
    - Remove: < > " ' % ;) (& + -
  - Check for the length
  - Use Server.HtmlEncode
- .NET 1.1
  - Default no HTML code in Textboxes
  - Page Attribut ValidateRequest =false

HTTP Harvesting

- Database driven websites
- Display result based on
  - Text Input, Querystring, Cookie
- Special type of SQL query language
- Datagrid list with detail link
  - Detail.aspx?id=1
- Session attaching+ pagelink
- Email address for spammer

Prevent HTTP harvesting

- Encrypt querystrings
- Combine user input with textboxes
- Use Jscript to write the data
- Draw the data
  - System.drawing
- Monitor the web usage
- Third party review

Canonicalization

- Character Sets URL, Querystring, Filename
  - %20=“ “
- IP Address as decimal
- Compare values
  - HTMLDecode

Much more…

Architecture

- Operation System
  - Reduce the rights of accounts
    - Never use Admin Rights
  - Switch of unused services and ports
- Web Farm
  - Use ipsec to encrypt traffic
    - Between SQL Server and Web Application
    - Session Management
  - IP restrictions
- Change common used things
  - Directories, users, path

Tools

- Microsoft Baseline Security Analyzer 1.2
  - Scan network or local
  - Scan installed updates
  - Scan well-known issues

How to be secure

- Don't believe in 100 %
- Evaluate the risk
  - Risk of attack
  - Damage result
- Train everybody
  - Architects, Developer, User, Administrator
- Review
  - Code and user interface

Security begins in mind

Hinweise

- Abo Angebot ASP.NET professional
  - 24 Euro statt 36 (hier und jetzt)
- ASP-Konferenz
  - 14.-15 Juni Burghausen
- DevTrain Camp
  - 5 Tage – 50 h - .NET 29.03-02.04 Burghausen
- DevTrain.de
  - Kostenfreies Community Portal