Advances in secure (ASP).NET development – Break the hackers' spirit

Alexandre Herzog
IT Security Analyst – Compass Security AG
Agenda

Introduction to .NET

Configuration of (ASP).NET applications

New features of (ASP).NET 4.5

Key security points of application lifecycle
  ✦ Development
  ✦ Deployment
  ✦ Operations
  ✦ Third party component review
Aim of this talk

Discover the (ASP).NET framework and its limitations

Give you a set of points to observe for your next (ASP).NET application release

No discussion about the code

The focus is on applications
  ✦ Not on infrastructure
  ✦ nor Microsoft’s Security Development Lifecycle.

This talk won’t be too technical
  ✦ Just enough to cover these points
Bio of Alexandre Herzog

Vaudois exilé d’abord en Valais, then Wellington (NZ) und jetzt Zürich

Mainly worked for banks as sysadmin / developer

Just finished my MAS in Information Security (LU)

Author of several security advisory
  ✦ Including CVE-2013-1330 patched in MS13-067

Currently working as IT Security Analyst for Compass Security AG in Bern & Rapperswil/Jona
Introduction to .NET

Configuration of (ASP).NET applications

New features of (ASP).NET 4.5

Key security points of application lifecycle
  ✦ Development
  ✦ Deployment
  ✦ Operations
  ✦ Third party component review
Introduction to .NET

The .NET Framework is a development platform for building apps for Windows, Windows Phone, Windows Server, and Windows Azure.

It consists of the common language runtime (CLR) and the .NET Framework class library, which includes classes, interfaces, and value types that support an extensive range of technologies.

The .NET Framework provides a managed execution environment, simplified development and deployment, and integration with a variety of programming languages, including Visual Basic and Visual C#.

[MS_DotNet_Def]
Introduction to .NET

The .NET Framework is a development platform for building apps for Windows, Windows Phone, Windows Server, and Windows Azure.

It consists of the common language runtime, the Framework class library, which includes class libraries for value types that support an extensive range of data types, and a managed execution environment, simplification of development and deployment, and integration with a variety of programming language, including Visual Basic and Visual C#.

This framework is installed by default on any Windows device. It’s also used for Silverlight.

[MS_DotNet_Def]
Introduction to .NET

The .NET Framework is a development platform for building apps for Windows, Windows Phone, Windows Server, and Windows Azure.

It consists of the common language runtime (CLR) and the .NET Framework class library, which includes classes, interfaces, and value types that support an extensive range of technologies.

[MS_DotNet_Def]
Introduction to .NET

The .NET Framework is a development platform for building apps for Windows, Windows Phone, Windows Server, and Windows Azure.

It consists of the common language runtime, a Framework class library, which includes collection classes and value types that support an extensive range of technologies.

Enhances the security (e.g. no buffer overflow is possible).

The .NET Framework provides a managed execution environment, simplified development and deployment, and integration with a variety of programming languages, including Visual Basic and Visual C#.

[MS_DotNet_Def]
Introduction to .NET

The .NET Framework is a development platform for building apps for Windows, Windows Phone, Windows Server, and Windows Azure.

It consists of the common language runtime (CLR), .NET Framework class library, which includes classes, interfaces, and value types that support an extensive range of technologies.

The .NET Framework provides a managed execution environment, simplified development and deployment, and integration with a variety of programming languages, including Visual Basic and Visual C#.

You can also compile F#, IronPython, IronRuby, J# etc…

[Wiki_IL_Lang]

[MS_DotNet_Def]
Introduction to .NET

Sounds like Java!

Yes, because

- It’s byte code => the code can be reversed
- Multiplatform (can also run on Linux using Mono)

No, because

- Different versioning scheme
  - All versions of .NET but 1.0 are still supported
  - Supported versions get security patches (1.1, 2.0, 3.0, 3.5, 4.0, 4.5)
- The .NET framework is pre-installed on Windows
.NET also features runtime Trust Level

- An app running with Trust Level set to medium cannot, e.g., access the registry or files outside the app’s folder \(^{[MS\_Trust]}\)
- This is not related to the Windows Mandatory Integrity Control (MIC)

Close interaction of ASP.NET with IIS

.NET is not (yet?) as targeted / vulnerable as Java

You can compile .NET code on any Windows device
Agenda

Introduction to .NET

Configuration of (ASP).NET applications

New features of (ASP).NET 4.5

Key security points of application lifecycle
  ✦ Development
  ✦ Deployment
  ✦ Operations
  ✦ Third party component review
Config of (ASP).NET applications

Typical configuration

- Proxy settings
- Cryptographic keys
- Cookie settings
- Compilation details
- Error handling
- Retail mode
- Trust level
- Database connections
- ViewState parameters
- Trace configuration
- Request validation
- Application settings
- ...
- See [MS_AspNet_config], [MS_Net_config], [HL_Decompile]
Config of (ASP).NET applications

Configuration is based on .config files

For .NET executables, the config hierarchy is

- Server level config (machine.config)
- Application specific config ([AppName].exe.config)
- Optional user settings (roaming.config & user.config)
Config of (ASP).NET applications

Do not use Group Policy Preferences to distribute configurations in your Windows domain!

- For more details see [CSNC_GPP]
Config of (ASP).NET applications

Config hierarchy for ASP.NET code

- Server level config (machine.config & web.config)
- Web site (web.config)
- ASP.NET application and subfolders (web.config)
- Further details available in [ASPNET_config]

IIS configuration is also involved for ASP.NET code

- %windir%\system32\inetsrv\config\ApplicationHost.config
Config of (ASP).NET applications

Configuration Hierarchy

Machine.config

[AppName].exe.config

<custom merge>

Roaming.config

User.config

AppSettings.config (External)

Root\SiteA\Web.config

Root\SiteB\Web.config

Root\SiteA\Site1\Web.config

Source:
[CP_net_config]
Config of (ASP).NET applications

The configuration can be locked at any level [MS_lock_config]

The configuration on the server level is dependent of the .NET version and CPU architecture

- %systemroot%\Microsoft.NET\Framework[64]\[version]\CONFIG

It is possible to encrypt sections of the configuration file (only useful for web.config files) [MS_enc_config]
Why encrypt the configuration file?

- Limits the impact of file inclusion issues or leaking code / configuration files
- An attacker first needs to execute a command on the web server before being the config is in clear text

Recommended sections to encrypt

- `<MachineKey />`
- `<ConnectionString />`
- Any other settings where keys, passwords or endpoint information is stored
Agenda

Introduction to .NET

Configuration of (ASP).NET applications

New features of (ASP).NET 4.5

Key security points of application lifecycle

- Development
- Deployment
- Operations
- Third party component review
New features of (ASP).NET 4.5

Microsoft is continuously improving .NET

✦ Task based async model
✦ Enhanced Strong Naming for Windows Store apps
✦ WebSockets, etc

Especially security relevant is that

✦ The standalone Anti-XSS library is now integrated
✦ Several changes occurred in the handling of cryptography
New features of (ASP).NET 4.5

Why such crypto improvements in version 4.5?

- “Cryptography in the Web: The Case of Cryptographic Design Flaws in ASP.NET” research of [Duong_Rizzo]
- Padding oracle attack patched by Microsoft for all versions in MS10-070
- All details of these changes in (ASP).NET 4.5 are described in [MS_Improv_1] to [MS_Improv_3]
New features of (ASP).NET 4.5

Visible impact of (ASP).NET 4.5

- Several changes are opt-in
  - Action is required!
- ViewState fields will be encrypted
- Some compatibility with ASP.NET 2.0 may/will be lost
New features of (ASP).NET 4.5

Extract of the appendix of this talk:

Configuration checklist

If you run ASP.NET 4.5

- Ensure section `<httpRuntime>` enables all new features with attribute `targetFramework="4.5"`.
- Once done, ensure the following config sections are either absent or set to the following values:
  - `<machineKey compatibilityMode="Framework45" />`
  - `<compilation targetFramework="4.5" />`
  - `<pages controlRenderingCompatibilityVersion="4.5" />`
- Configure AntiXSS to be the default encoding routine
  `<httpRuntime [...] encoderType="System.Web.Security.AntiXss.AntiXssEncoder, System.Web, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />`
New features of (ASP).NET 4.5

Do these changes justify migration to .NET 4.5?

- YES absolutely
- Several defence in-depth mechanisms were added
Agenda

Introduction to .NET

Configuration of (ASP).NET applications

New features of (ASP).NET 4.5

Key security points of application lifecycle

- Development
- Deployment
- Operations
- Third party component review
Key security points of app lifecycle

> Development

Why not use this opportunity to start setting up a Security Development Lifecycle for your apps? [MS(SDL)]
Key security points of app lifecycle > Development

What is the Security Development Lifecycle?

The Security Development Lifecycle (SDL) is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost.

Assess your security

Discover ways to improve your security practices.

Get Started

Tools

- **Attack Surface Analyzer 1.0**
  Understand your attack surface before & after new apps are deployed.

- **SDL Threat Modeling Tool v3.1.8**
  A tool to help engineers find and address system security issues.

- **MiniFuzz basic file fuzzing tool**
  A simple fuzzer designed to ease adoption of fuzz testing.

- **Regular expression file fuzzing tool**
  A tool to test for potential denial of service vulnerabilities.

Design Phase

SDL Practice #5: Establish Design Requirements

Considering security and privacy concerns early helps minimize the risk of schedule disruptions and reduce a project's expense.

Key security points of app lifecycle
> Development

SDL Practice #10: Perform Static Analysis

Analyzing the source code prior to compilation provides a scalable method of security code review and helps ensure that secure coding policies are being followed.

When should this practice be implemented?
Traditional Software development: Implementation Phase
Agile development: Every Sprint

Resources specific to this practice

<table>
<thead>
<tr>
<th>DOWNLOADS</th>
<th>VIDEOS</th>
<th>WEBCASTS</th>
<th>TRAINING</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT.NET 64-bit</td>
<td>CAT.NET 64-bit</td>
<td>Detecting and Mitigating Security Issues Using the Code Analysis Tool .NET (Level 200)</td>
<td>SDL Quick Security References</td>
</tr>
<tr>
<td>Anti-XSS</td>
<td>Anti-XSS</td>
<td></td>
<td>SDL Developer Starter Kit</td>
</tr>
<tr>
<td>FxCop</td>
<td>FxCop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code Analysis for C/C++</td>
<td>Code Analysis for C/C++</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDL Developer Starter Kit – Code Analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Key security points of app lifecycle

Development

Develop on .NET 4.5 (especially for web apps) and for a medium trust level whenever possible

Use the free Microsoft SDL tools while developing
- FxCop [MS_FxCop] & CATNET [MS_CATNET]

Do not turn off security features
- Request Validation, ViewState MAC, ...

Do not rely on client side only validation or include/hide secrets in client side applications

Teach best practices to your developers...
Key security points of app lifecycle

> Deployment

Lock down the server and app configuration

Consider an obfuscator for your client side apps
  ✦ Executable or Silverlight only

Do not use GPPs to distribute configurations!

Consider reducing the trust level of your app whenever possible

Perform a general server hardening (OS & IIS)
  ✦ Again, this “infrastructure” part is not covered here
Key security points of app lifecycle

> Operations

Run ASP.NET 4.5 with medium trust apps

Encrypt sensitive sections of the web.config file

Manage the cryptographic keys you use!
  ✤ Web.config encryption & ASP.NET features (Machine Key)

Patch the server & configure IIS adequately

Communicate
  ✤ Be ready in case of a (security) incident
  ✤ All technical stakeholders should come together…
Key security points of app lifecycle
> Third party component review

The same recipes apply:
- As it's just byte code, let's decompile the application!

Loi sur le droit d'auteur
Art. 21 Décryptage de logiciels
1 La personne autorisée à utiliser un logiciel peut se procurer, par le décryptage du code du programme, des informations sur des interfaces avec des programmes développés de manière indépendante. Elle peut opérer elle-même ou mandater un tiers.
2 Les informations sur des interfaces obtenues par le décryptage du code du programme ne peuvent être utilisées que pour développer, entretenir et utiliser des logiciels interopérables, pourvu qu'une telle utilisation ne porte pas atteinte à l'exploitation normale du programme ni ne cause un préjudice injustifié aux intérêts légitimes de l'ayant droit.
Key security points of app lifecycle
> Third party component review

The same recipes apply:
- Audit source code & configuration
- Audit assemblies with static analysis tools
- Run the component with the lowest possible trust level
- Regenerate all keys / secrets shipped by the vendor

Manage the component by
- Monitoring for security patches
- Update it periodically
Agenda

Introduction to .NET

Configuration of (ASP).NET applications

New features of (ASP).NET 4.5

Key security points of application lifecycle
  ✦ Development
  ✦ Deployment
  ✦ Operations
  ✦ Third party component review
Conclusion

Top security issues in .NET include

- Application information leak
  - Verbose error messages
  - Secrets stored within the code (executable or Silverlight)

- Injections
  - SQL injections due to unsafe database requests

- Unsafe application settings
  - Unencrypted communication
  - Unsafe distribution of credentials

Solved by configuration (server or app)
Static code analysis
No secrets in the code! Consider an obfuscator
Education of devs, code review & static analysis
Education of devs, code and config review
No secrets in the code! Rely on Windows auth. when possible
Conclusion

Top security issues in ASP.NET include

- Application information leak
  - Secrets stored in the ViewState
  - Verbose error messages

- Unsafe application settings
  - Session cookie parameters
  - Request validation disabled
  - Unencrypted configuration file

- Injections
  - XSS due to user inputs in JS or HTML attributes
  - SQL injections due to unsafe database requests

Configure this field to be encrypted or migrate to ASP.NET 4.5

Hardening / lockdown of the configuration (server or app)
Static code analysis

Encryption of configuration file

Education of devs, code review & static analysis
Conclusion

.NET is a secure framework following the SD³+C principle
[MS_SD³C]:

- Secure by Design, Secure by Default, Secure in Deployment, and Communications

Your applications can also benefit from this security during their lifecycle

This talk focused on application security

- You still have to harden your infrastructure (OS & IIS)!
Danke/Merci/Thank you!

Contact:

alexandre.herzog@csnc.ch

Company blog

- http://blog.csnc.ch/

LinkedIn

- http://ch.linkedin.com/in/alexandreherzog/

Google+

- https://plus.google.com/u/1/109572456864701444940/
References (1/3)


[Wiki_Components]

[Wiki_IL_Lang]
http://en.wikipedia.org/wiki/Category:.NET_programming_languages


References (2/3)


[MS(SDL)] http://www.microsoft.com/security/sdl/default.aspx

References (3/3)


Configuration checklist

This checklist is by no means complete. It’s just the starting point of your configuration journey…

Depending on your situation, you may want to configure these settings on a server (e.g. machine.config) and lock them or on an application level (web.config)
Configuration checklist

List of configuration which should be forced on an integration / production server

In the `machine.config` for all .NET versions

```xml
[...]
<system.web>
  <deployment retail="true" />
  <pages viewStateEncryptionMode="Always" />
  <httpCookies httpOnlyCookies="true"
    requireSSL="true" />
</system.web>

<authentication>
  <forms requireSSL="true" />
</authentication>
[...]
```
List of settings which are secure by default. They should not be disabled in configuration or code:

- For the `<pages>` configuration section
  - Property `enableEventValidation` should stay `true`
  - Property `enableViewStateMac` should stay `true`
  - Property `validateRequestMethod` should stay `true`

- For the `<forms>` configuration section
  - Property `enableCrossAppRedirects` should stay `disabled`
  - Property `protection` should stay `all`
List of settings which are secure by default. They should not be disabled in configuration or code:

- For the `<trace>` configuration section
  - Property enabled should stay `false`
- For the `<customErrors>` configuration section
  - Property mode should stay `RemoteOnly` or `On`
- For the `<compilation>` configuration section
  - Property debug should stay `false`

All these properties are set to a safe value if `<system.web><deployment retail="true" />`
If you run ASP.NET 4.5

- Ensure section `<httpRuntime>` enables all new feature with attribute `targetFramework="4.5"`[MS_Run_45]
- Once done, ensure the following config sections are either absent or set to the following values:
  - `<machineKey compatibilityMode="Framework45" />`
  - `<compilation targetFramework="4.5" />`
  - `<pages controlRenderingCompatibilityVersion="4.5" />`
- Configure AntiXSS to be the default encoding routine

```xml
<httpRuntime [...]
encoderType="System.Web.Security.AntiXss.AntiXssEncoder,System.Web, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
```
Configuration checklist

Trust level and their impact [MS_Trust_HowTo]

- Example of a web app running with medium trust

```xml
<system.web>
  <trust level="Medium" originUrl="" />
</system.web>
```

This web application would not be able to

- Call unmanaged code.
- Call serviced components.
- Write to the event log.
- Access Microsoft Message Queuing queues.
- Access ODBC, OleDb, or Oracle data sources.
- Access files outside the application directory.
- Access the registry.
- Make network or Web service calls (allowed URLs can be defined)