Advances in secure (ASP).NET development - break the hackers' spirit

Alexandre Herzog

IT Security Analyst – Compass Security AG

Application Security Forum - 2013
Western Switzerland

15-16 octobre 2013 - Y-Parc / Yverdon-les-Bains
http://www.appsec-forum.ch

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 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

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

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 (CL .NET Framework class library, which includes cl interfaces, and value types that support an exte 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#.

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

[MS_DotNet_Def]

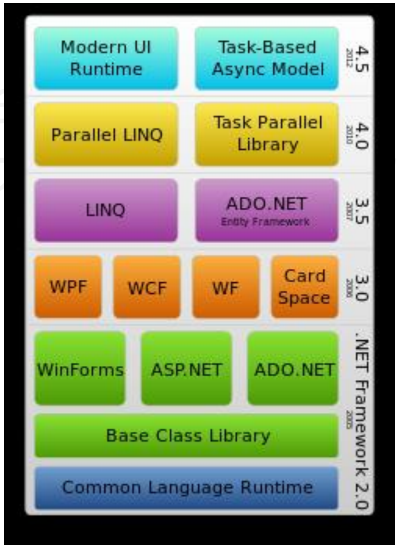
Introduction to .NET

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.

The .NET Framework provides a managed e environment, simplified development and d nd integration with a variety of programming including Visual Basic and Visual C#.

[MS_DotNet_Def]



[Wiki_Components]

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 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) and the .NET Framework class library, which inc interfaces, and value types that support technologies.

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

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

- 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

Introduction to .NET

- .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

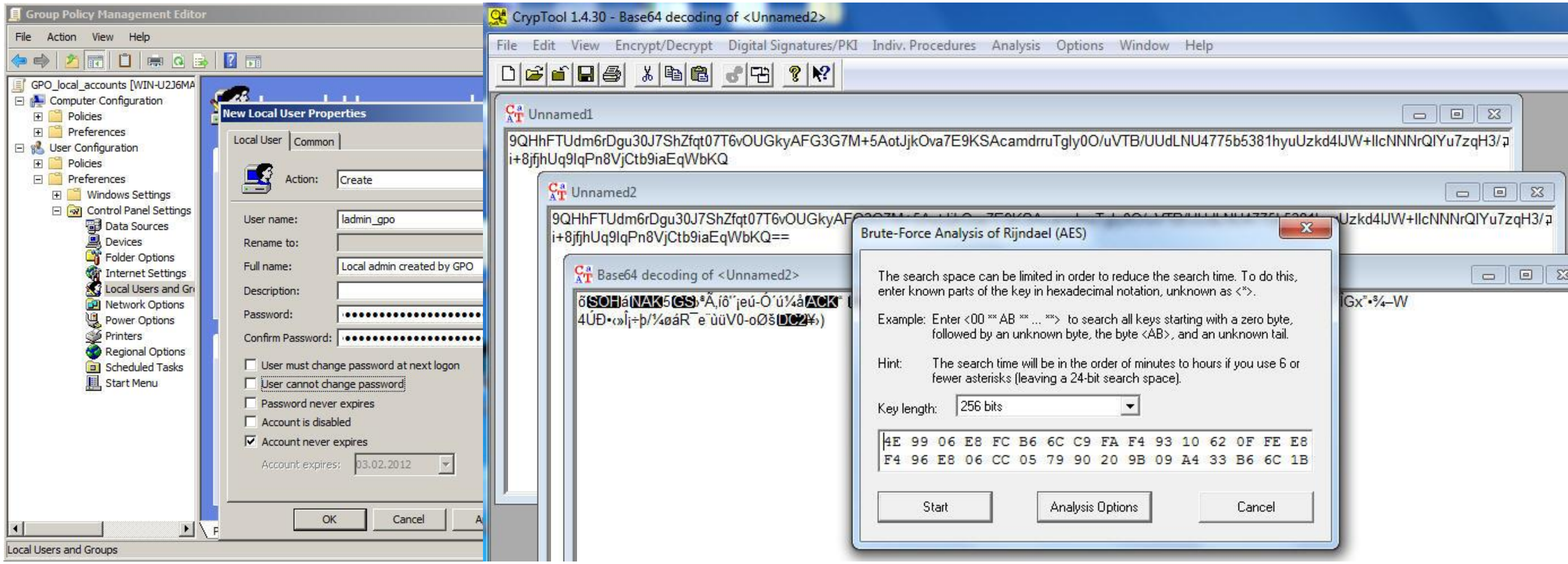
Config of (ASP).NET applications

- 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



New Local User Properties

Local User | Common

Action: Create

User name: ladmin_gpo

Rename to:

Full name: Local admin created by GPO

Description:

Password: ••••••••••••••••••••

Confirm Password: ••••••••••••••••••••

□ User must change password at next logon

□ User cannot change password

□ Password never expires

□ Account is disabled

☑ Account never expires

Account expires: 03.02.2012

OK | Cancel

```
9QHhFTUdm6rDgu30J7ShZfqt07T6vOUGkyAFG3G7M+5AotJjkOva7E9KSAcamdrruTgly0O/uVTB/UUdLNU4775b5381hyuUzkd4lJW+IlcNNNrQIYu7zqH3/
i+8jfjhUq9lqPn8VjCtb9iaEqWbKQ
```

```
9QHhFTUdm6rDgu30J7ShZfqt07T6vOUGkyAF
i+8jfjhUq9lqPn8VjCtb9iaEqWbKQ==
```

```
õSOHáNAK5GS¸ªÃ‚íô¨¡eú-Ó´ú¼åACK
4ÚÐ•‹»Ï¡÷þ/¼øáR¯e¨ùüV0-oØšDC2¥›)
```

Brute-Force Analysis of Rijndael (AES)

The search space can be limited in order to reduce the search time. To do this, enter known parts of the key in hexadecimal notation, unknown as <*>.

Example: Enter <00 ** AB ** ... **> to search all keys starting with a zero byte, followed by an unknown byte, the byte <AB>, and an unknown tail.

Hint: The search time will be in the order of minutes to hours if you use 6 or fewer asterisks (leaving a 24-bit search space).

Key length: 256 bits

```
4E 99 06 E8 FC B6 6C C9 FA F4 93 10 62 0F FE E8
F4 96 E8 06 CC 05 79 90 20 9B 09 A4 33 B6 6C 1B
```

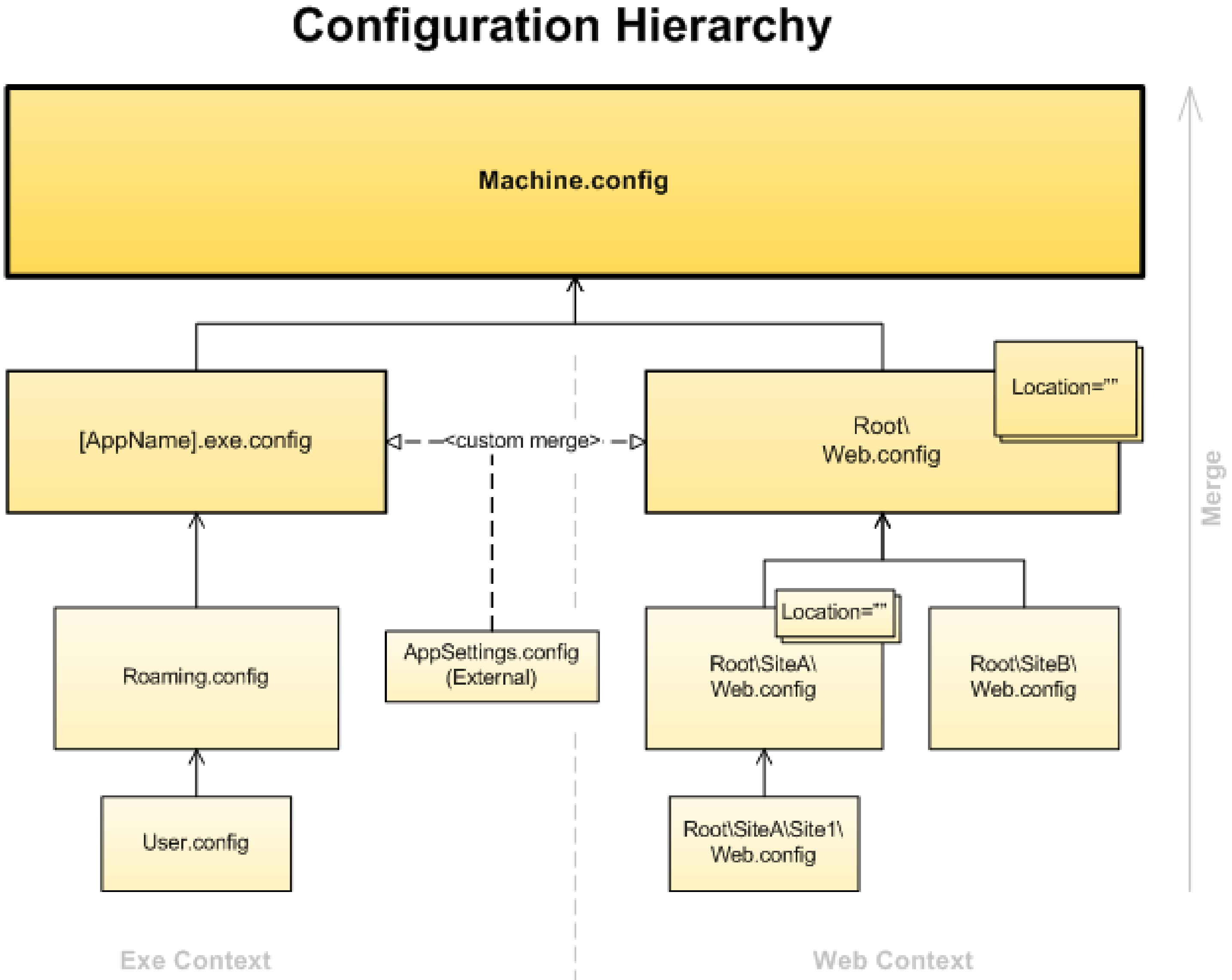
Start | Analysis Options | Cancel

- 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



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]

Config of (ASP).NET applications

- 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 />
  - <ConnectionStrings />
  - 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

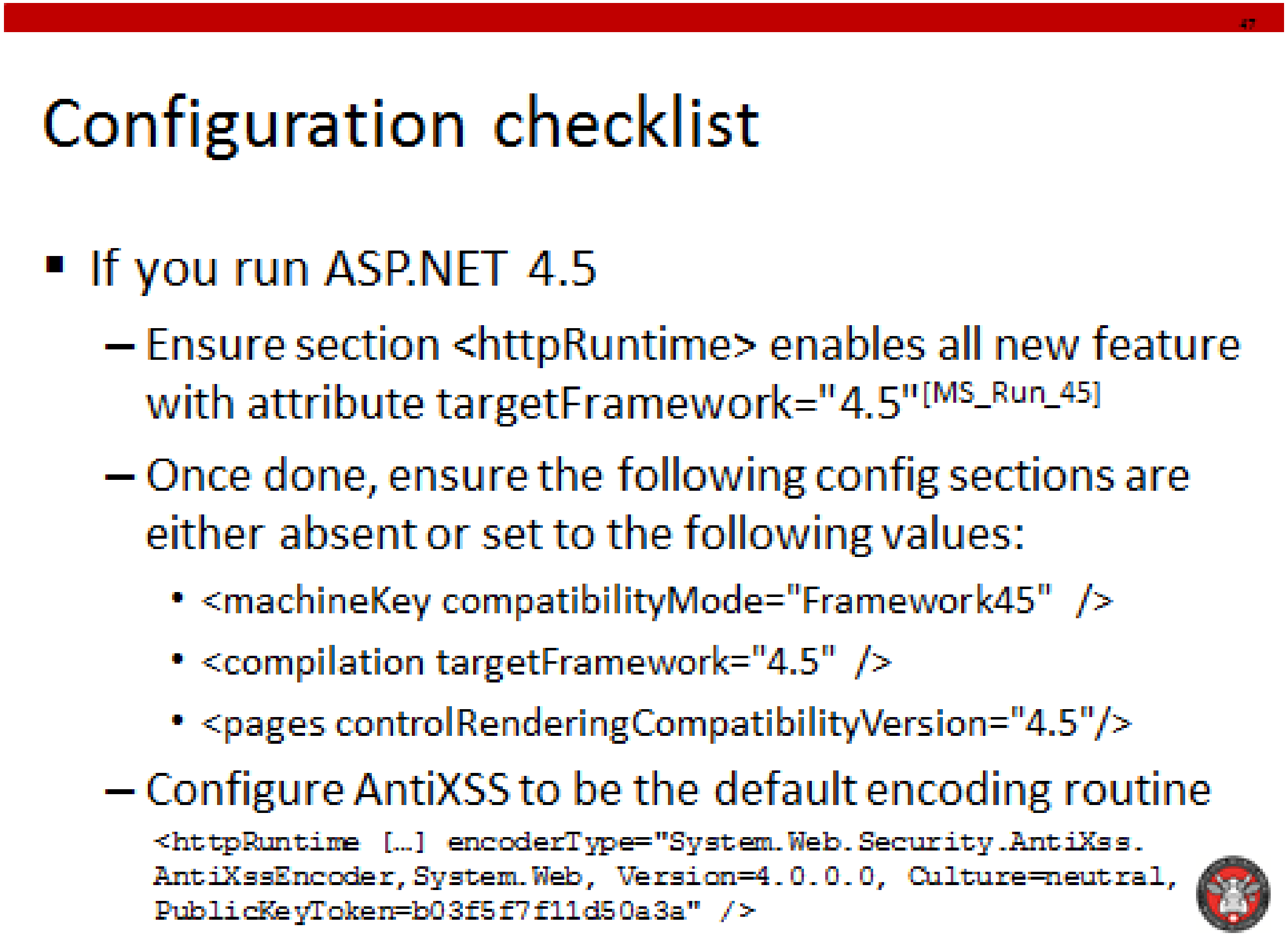
- Why such crypto improvements in version 4.5?
  - *"Cryptography in the Web: The Case of Cryptographic Design Flaws in ASP.NET"* research of [Duong_Rizzo]
  - Padding oracle attack patched by Microsoft for all versions in MS10-070
  - All details of these changes in (ASP).NET 4.5 are described in [MS_Improv_1] 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 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

```
<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

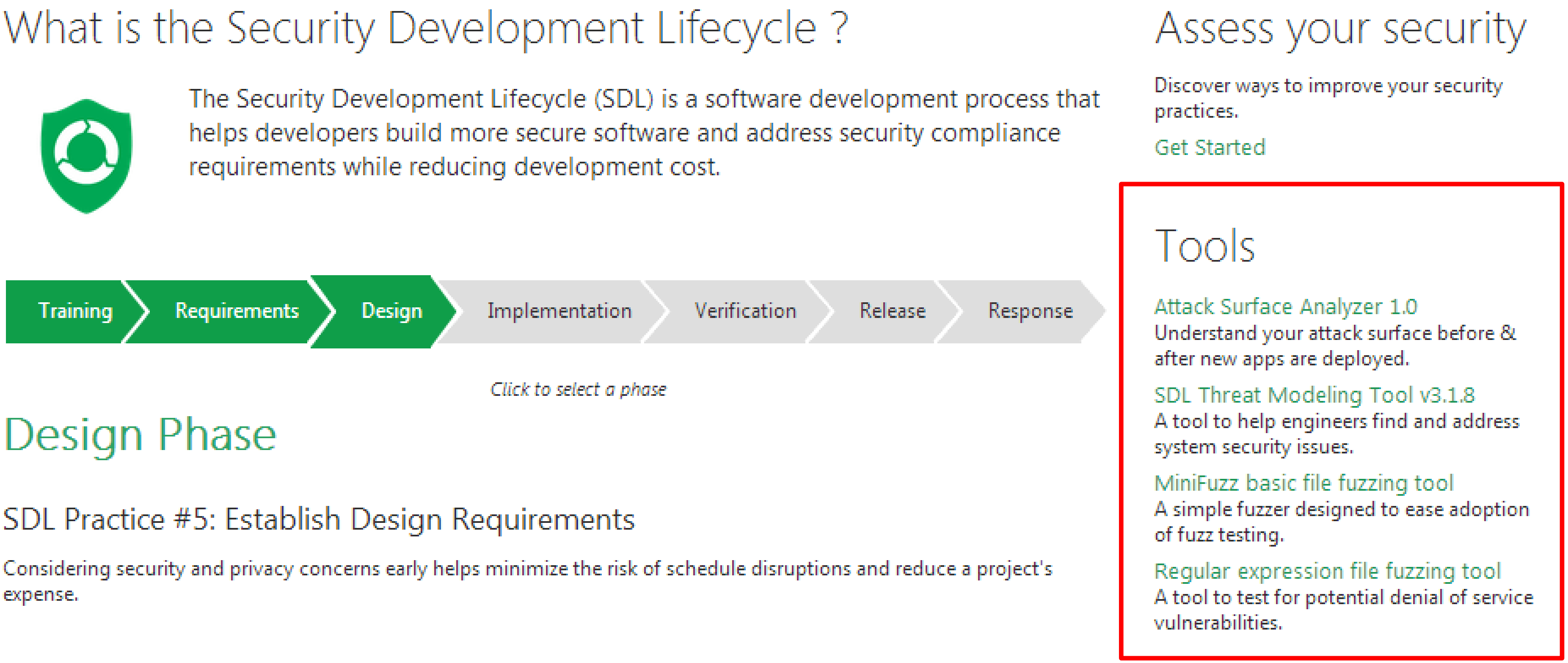
Key security points of app lifecycle >

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.

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.

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.

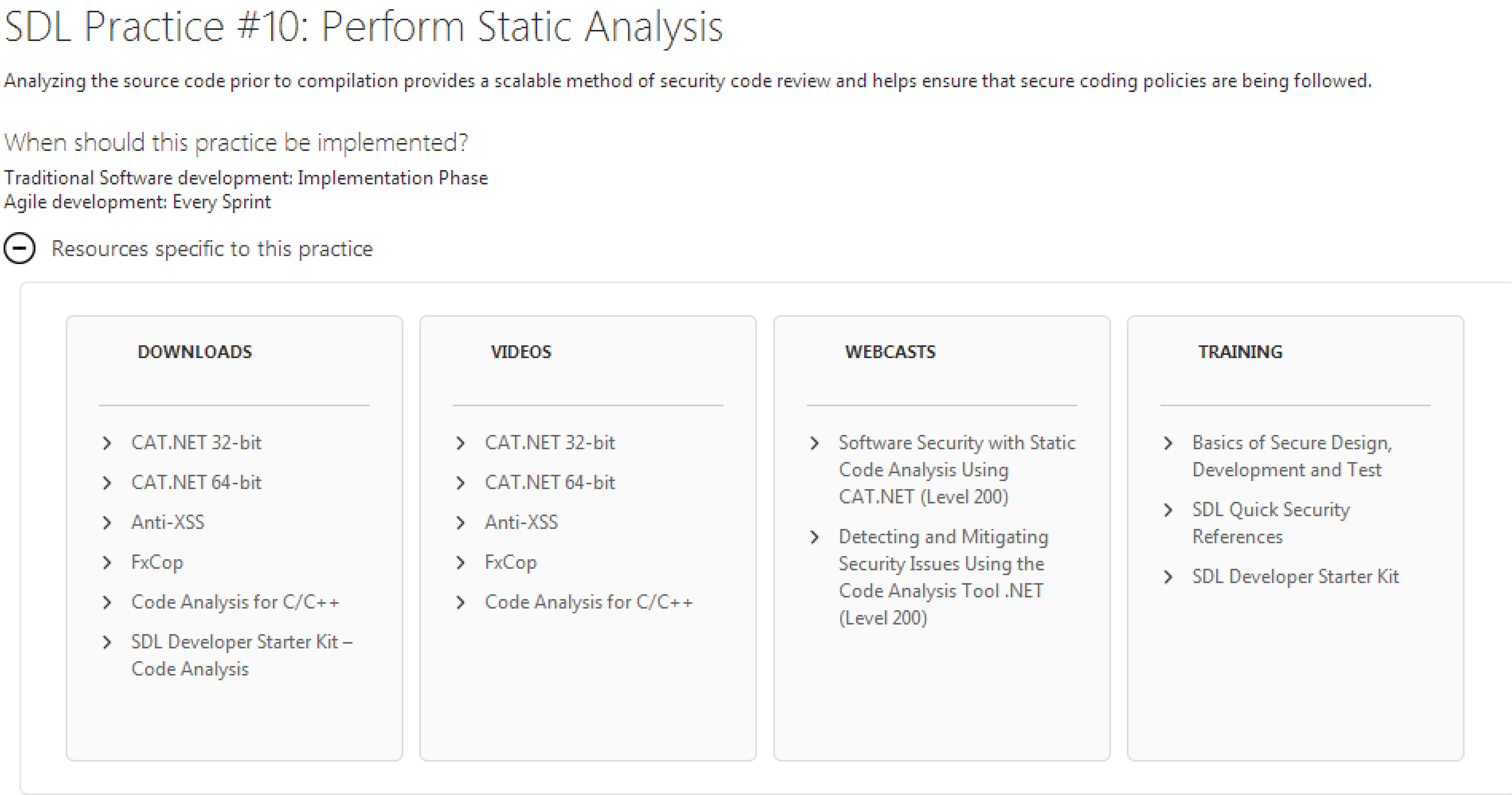
SDL Threat Modeling Tool v3.1.8
A tool to help engineers find and address system security issues.

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.

Source http://www.microsoft.com/security/sdl/

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

DOWNLOADS

- CAT.NET 32-bit
- CAT.NET 64-bit
- Anti-XSS
- FxCop
- Code Analysis for C/C++
- SDL Developer Starter Kit – Code Analysis

VIDEOS

- CAT.NET 32-bit
- CAT.NET 64-bit
- Anti-XSS
- FxCop
- Code Analysis for C/C++

WEBCASTS

- Software Security with Static Code Analysis Using CAT.NET (Level 200)
- Detecting and Mitigating Security Issues Using the Code Analysis Tool .NET (Level 200)

TRAINING

- Basics of Secure Design, Development and Test
- SDL Quick Security References
- SDL Developer Starter Kit

Source http://www.microsoft.com/security/sdl/

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[MS_Trust_expl]
- 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

      Solved by configuration (server or app)
      Static code analysis

    - Secrets stored within the code (executable or Silverlight)

      No secrets in the code!
      Consider an obfuscator

  - Injections
    - SQL injections due to unsafe database requests

      Education of devs, code review & static analysis

  - Unsafe application settings
    - Unencrypted communication

      Education of devs, code and config review

    - Unsafe distribution of credentials

      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 SD^3+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)!

Questions?

Danke/Merci/Thank you!

Contact:

alexandre.herzog@csnc.ch

Company blog	http://blog.csnc.ch/
LinkedIn	http://ch.linkedin.com/in/alexandreherzog/
G+	https://plus.google.com/u/1/109572456864701444940/
Slides	http://slideshare.net/ASF-WS/presentations

References (1/2)

- [MS_DotNet_Def] http://msdn.microsoft.com/en-us/library/w0x726c2.aspx
- [Wiki_Components] http://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/DotNet.svg/250px-DotNet.svg.png
- [Wiki_IL_Lang] http://en.wikipedia.org/wiki/Category:.NET_programming_languages
- [ASPNET_config] http://msdn.microsoft.com/en-Us/library/ms178685.aspx
- [CP_net_config] http://www.codeproject.com/Articles/19675/Cracking-the-Mysteries-of-NET-2-0-Configuration
- [CSNC_GPP] http://blog.csnc.ch/2012/04/exploit-credentials-stored-in-windows-group-policy-preferences/
- [HL_Decompile] http://media.hacking-lab.com/largefiles/7205/Paper_DisassembleDotNetClient_v2.0.pdf
- [MS_lock_config] http://msdn.microsoft.com/en-us/library/ms228167(v=vs.100).aspx
- [MS_AspNet_config] http://msdn.microsoft.com/en-us/library/zeshe0eb(v=vs.85).aspx
- [MS_Net_config] http://msdn.microsoft.com/en-us/library/1fk1t1t0.aspx
- [Duong_Rizzo] http://www.ieee-security.org/TC/SP2011/PAPERS/2011/paper030.pdf

References (2/2)

- [MS_Improv_1] http://blogs.msdn.com/b/webdev/archive/2012/10/22/cryptographic-improvements-in-asp-net-4-5-pt-1.aspx
- [MS_Improv_2] http://blogs.msdn.com/b/webdev/archive/2012/10/23/cryptographic-improvements-in-asp-net-4-5-pt-2.aspx
- [MS_Improv_3] http://blogs.msdn.com/b/webdev/archive/2012/10/24/cryptographic-improvements-in-asp-net-4-5-pt-3.aspx
- [MS_enc_config] http://msdn.microsoft.com/en-us/library/dtkwfdky(v=VS.100).aspx
- [MS_SD3C] http://msdn.microsoft.com/en-us/library/ms995349.aspx
- [MS_SDL] http://www.microsoft.com/security/sdl/default.aspx
- [MS_FxCop] http://www.microsoft.com/en-us/download/details.aspx?id=6544
- [MS_CATNET] http://www.microsoft.com/en-us/download/details.aspx?id=5570
- [MS_Trust] http://msdn.microsoft.com/en-us/library/tkscy493(v=vs.85).aspx
- [MS_Trust_expl] http://msdn.microsoft.com/En-Us/library/wyts434y.aspx
- [MS_Run_45] http://blogs.msdn.com/b/webdev/archive/2012/11/19/all-about-httpruntime-targetframework.aspx
- [MS_Trust_HowTo] http://msdn.microsoft.com/en-us/library/ff648344.aspx

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

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

Configuration checklist

- 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 validateRequest should stay **true**
  - For the <forms> configuration section
    - Property enableCrossAppRedirects should stay **disabled**
    - Property protection should stay **all**

Configuration checklist

- 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" />

Configuration checklist

- 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

```
<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

```
<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)

Restrictions due to high trust level

Additional restrictions due to medium trust level