



Microsoft SDL: Agile Development

Nick Coblentz, CISSP, Microsoft MVP
Senior Security Consultant
AT&T Consulting

Nick.Coblentz@gmail.com

<http://nickcoblentz.blogspot.com>

<http://twitter.com/sekhmetn>

OWASP

November 11, 2010

Copyright © The OWASP Foundation

Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.

The OWASP Foundation

<http://www.owasp.org>

Bio

■ AT&T Consulting:

- ▶ Application Security
 - Penetration testing
 - Code review
 - Architecture and design reviews
 - Application security program development
 - Secure development methodology improvement

■ Research & Presentations

- ▶ **OWASP AppSec Research & Ireland 2010 Conferences**
- ▶ **ISSA Journal: Web Application Security Portfolios**
- ▶ **SAMM Interview Template**
- ▶ **Reducing Info Disclosure in ASP.NET Web Services and WCF Data Services**
- ▶ Turn Application Assessment Reports into Training Classes
- ▶ Observed Secure Software Development Stages
- ▶ Vulnerability Tracking, Workflow, and Metrics with Redmine
- ▶ Using Microsoft's AntiXSS Library 3.1

FireStarter: Agile Development and Security

I am a big fan of the **Agile** project development methodology, especially **Agile with Scrum**. I love the granularity and focus the approach requires. I love that at any given point in time you are working on the most important feature or function. I love the derivative value of communication and subtle form of peer pressure that Scrum

“...Agile hurts secure code development.”

But it comes with one *huge* caveat: **Agile hurts secure code development**. There, I said it. Someone had to. The Agile process, and even the scrum leadership model, hamstring development in the area of building secure products. Security is **not** a freakin' task card. Logic flaws are **not** well documented, discreet tasks to be assigned. Project managers (and unfortunately most ScrumMasters) learned security by skimming a 'For Dummies' book at Barnes & Noble while waiting for their lattes, but these are the folks making the choices as to what security should make it into the iterations. Just like general IT security, we end up wrapping the Agile process in a security blanket or bolting on security after the code is complete, because the process as we know it is not well suited to secure development.

Adrian Lane:

<http://securosis.com/blog/agile-development-and-security/>

OWASP



Microsoft SDL For Agile Released

Welcome to MSDN Blogs [Sign in](#) | [Join](#) | [Help](#)

 Microsoft®
Security Development Lifecycle

• HOME • EMAIL • RSS 2.0 • ATOM 1.0

Recent Posts

- [Announcing SDL for Agile Development Methodologies](#)
- [SDL at TechEd Europe and Platforma](#)
- [SIR Volume 7 Released](#)
- [Ninjas are cool, but engineers build bridges](#)
- [MS09-050, SMBv2 and the SDL](#)

Tags

Common Criteria [Crawl Walk Run](#) [Privacy](#) [SDL](#) [SDL Pro](#) [Network](#) [Security Assurance](#) [Security Blackhat](#) [SDL](#) [threat modeling](#)

Announcing SDL for Agile Development Methodologies

Hi everyone, Bryan here. There is a common misconception that because the SDL was originally created for Microsoft's big showcase box products like Windows and SQL Server, that it only works for those kinds of products. This is of course patently false: virtually every Microsoft product and online service, large or small, follows the SDL. Many other organizations outside of Microsoft are also successfully implementing the SDL. However, while the *content* of the SDL – its requirements and recommendations – may be universal, the *structure* of the SDL as originally designed is more suited to long-running waterfall- or spiral-style development methodologies. Consider the classic "chevron" SDL graphic:

Training	Requirements	Design	Implementation	Verification	Release	Response
• Core training	• Define quality gates/bug bar • Analyze security and privacy risk	• Attack surface analysis • Threat modeling	• Specify tools • Enforce banned functions • Static analysis	• Dynamic/fuzz testing • Verify threat models/attack surface	• Response plan • Final security review • Release secure	• Response execution

Source: <http://blogs.msdn.com/sdl/archive/2009/11/10/announcing-sdl-for-agile-development-methodologies.aspx>



Microsoft SDL



Microsoft Security Development Lifecycle (SDL)

Components:

- ▶ Best Practices
- ▶ Processes
- ▶ Standards
- ▶ Security Activities
- ▶ Tools

Goal:

“minimize security-related vulnerabilities in the design, code, and documentation and to detect and eliminate vulnerabilities as early as possible in the development life cycle.”



Microsoft®
Security Development Lifecycle

Which Software?

SDL applies to software that:

- Is used in Business environments
- Stores or transmits PII
- Communicates over the Internet or other networks



Source: Microsoft's Product Website

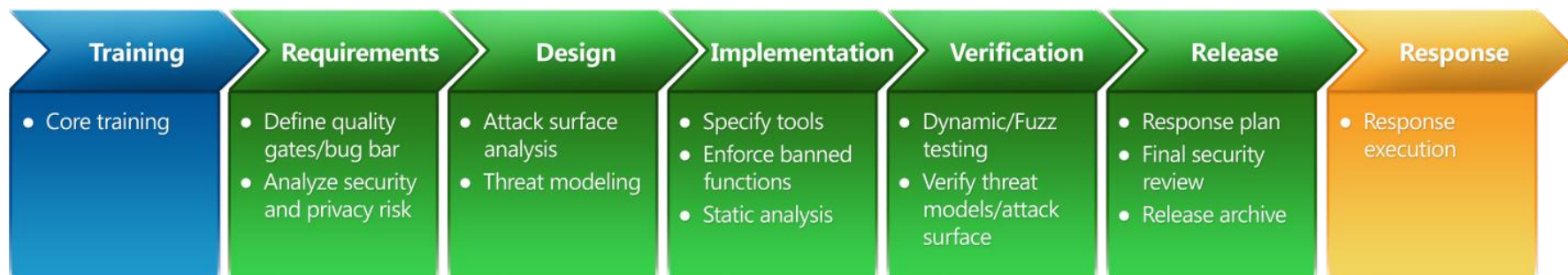
SDL Principles and Process

SD3+C

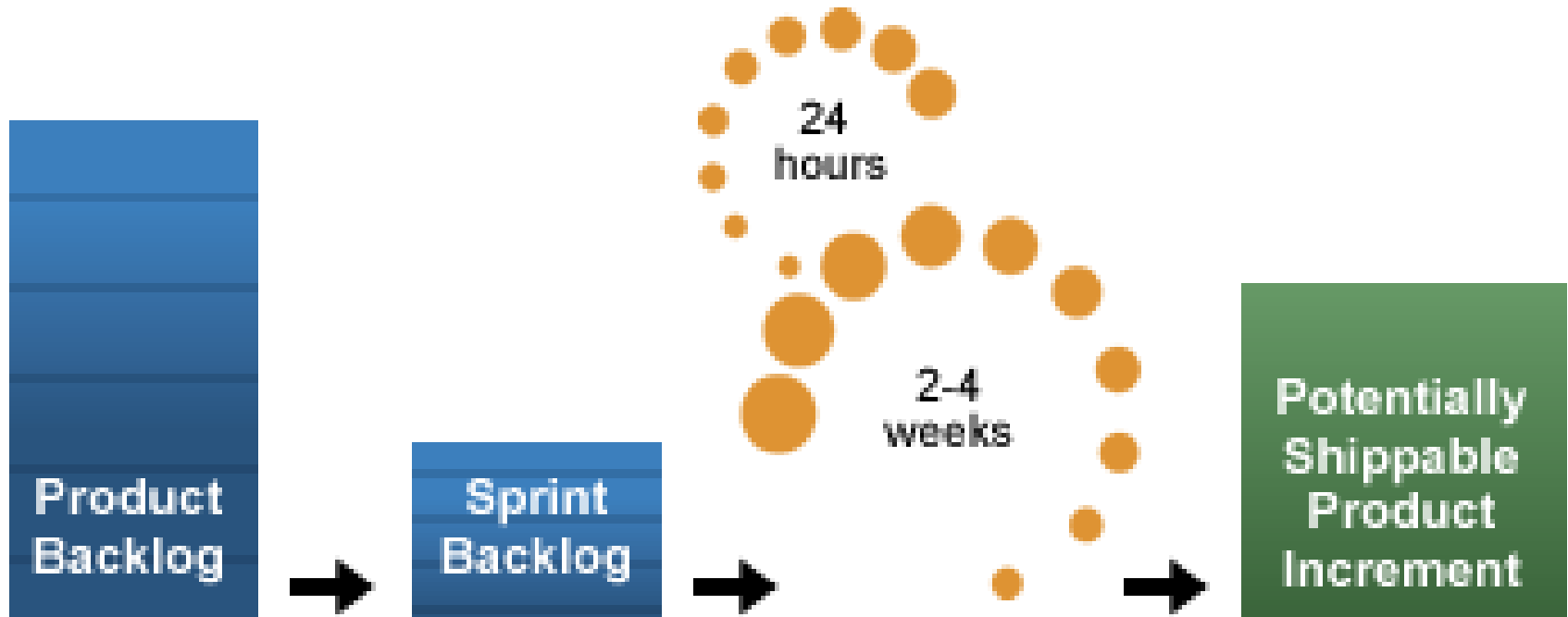
- Secure by Design
- Secure by Default
- Secure in Deployment
- Communications

PD3+C

- Privacy by Design
- Privacy by Default
- Privacy in Deployment
- Communications

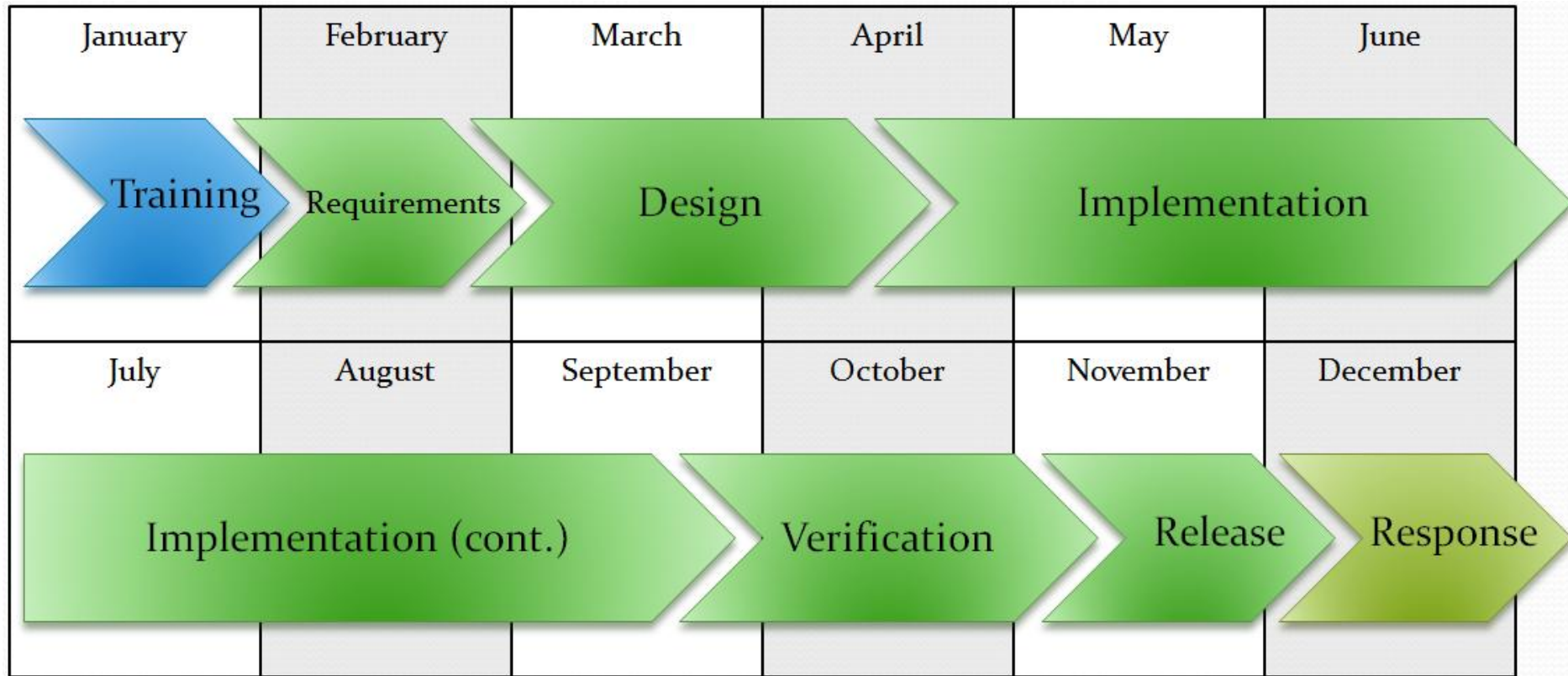


What is Agile Development?



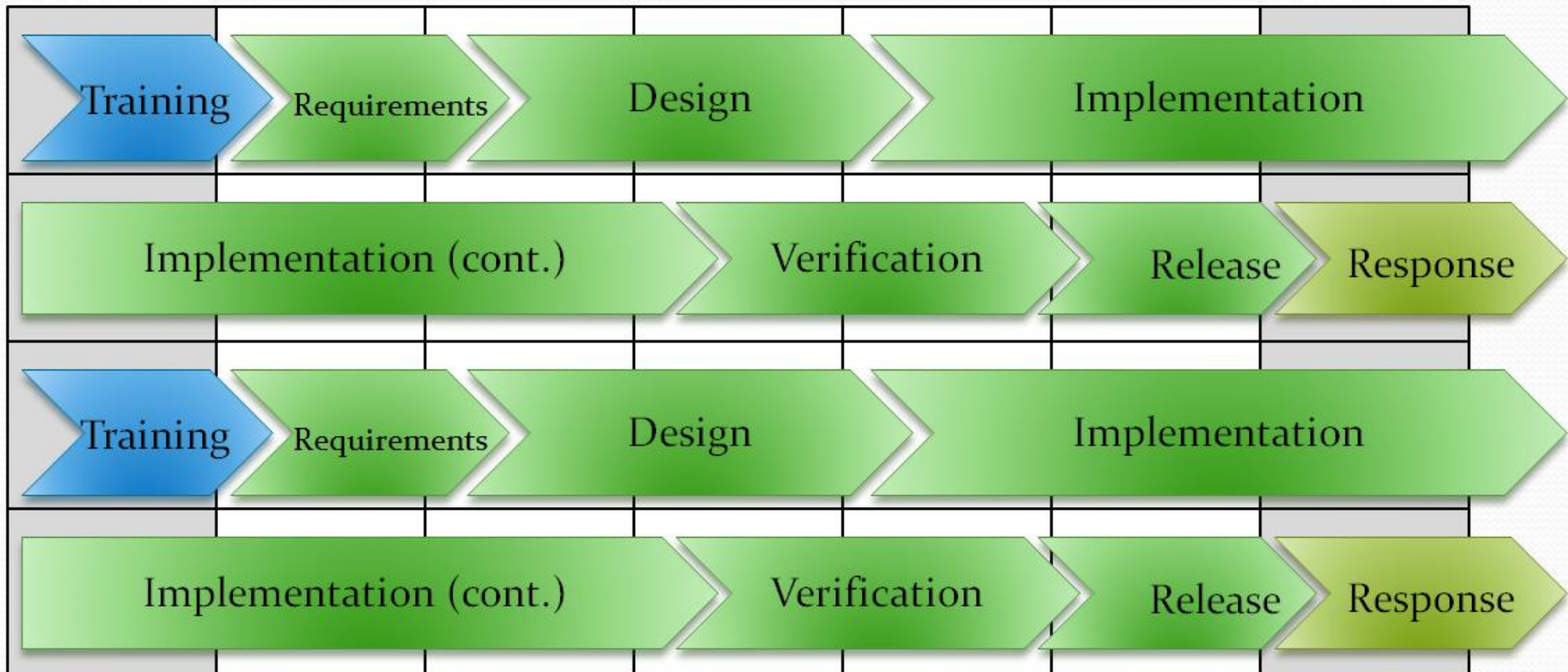
Source: http://www.scrumalliance.org/pages/what_is_scrum

SDLC (Waterfall Methodology)



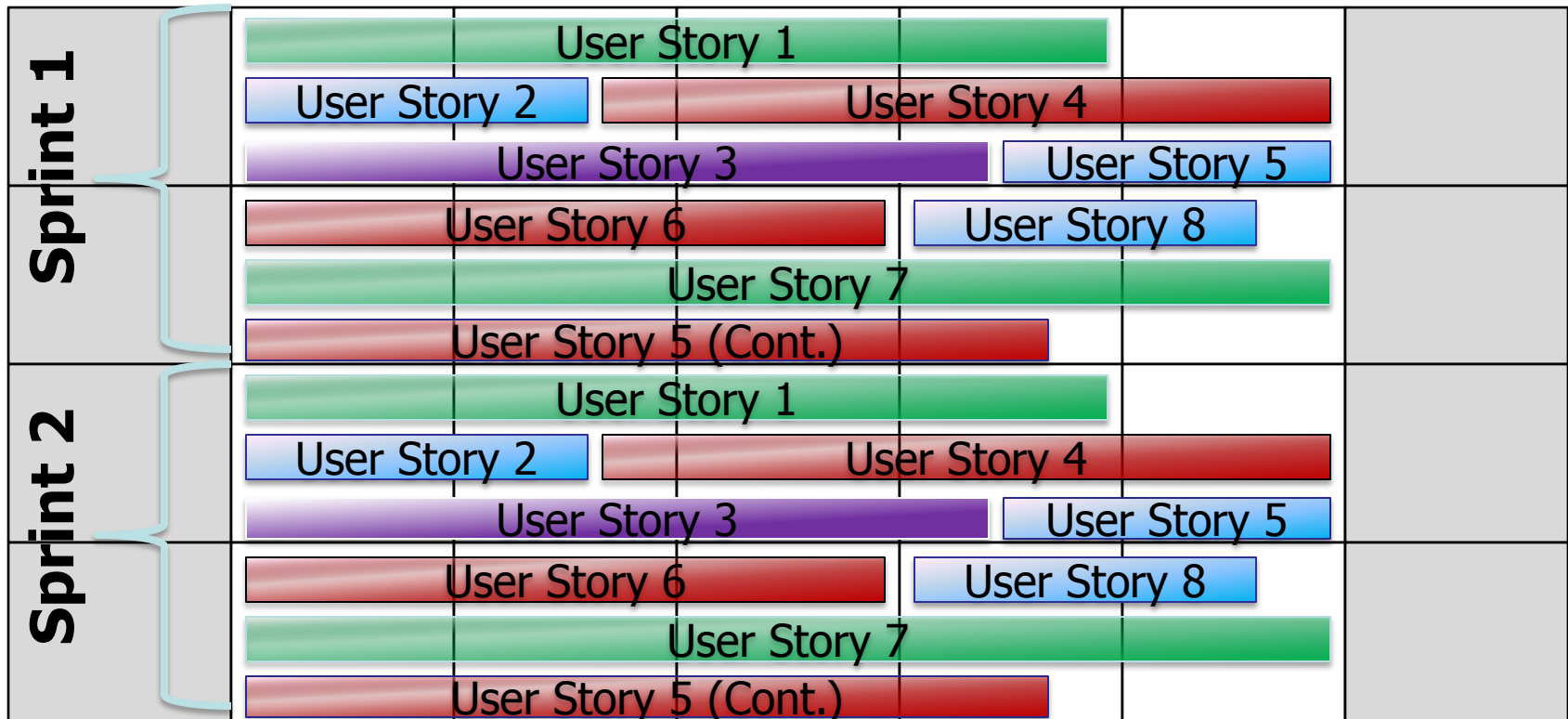
This Is NOT Agile Development

January



Agile Development

January



Agile Development



Source: http://www.scrumalliance.org/pages/what_is_scrum

- Cross-functional, self-organizing teams
- Short, time-boxed development iterations
- Delivery of small functional stories
- No *extensive* up front design or documentation

Planning and Design

<http://www.flickr.com/photos/acarlos1000>

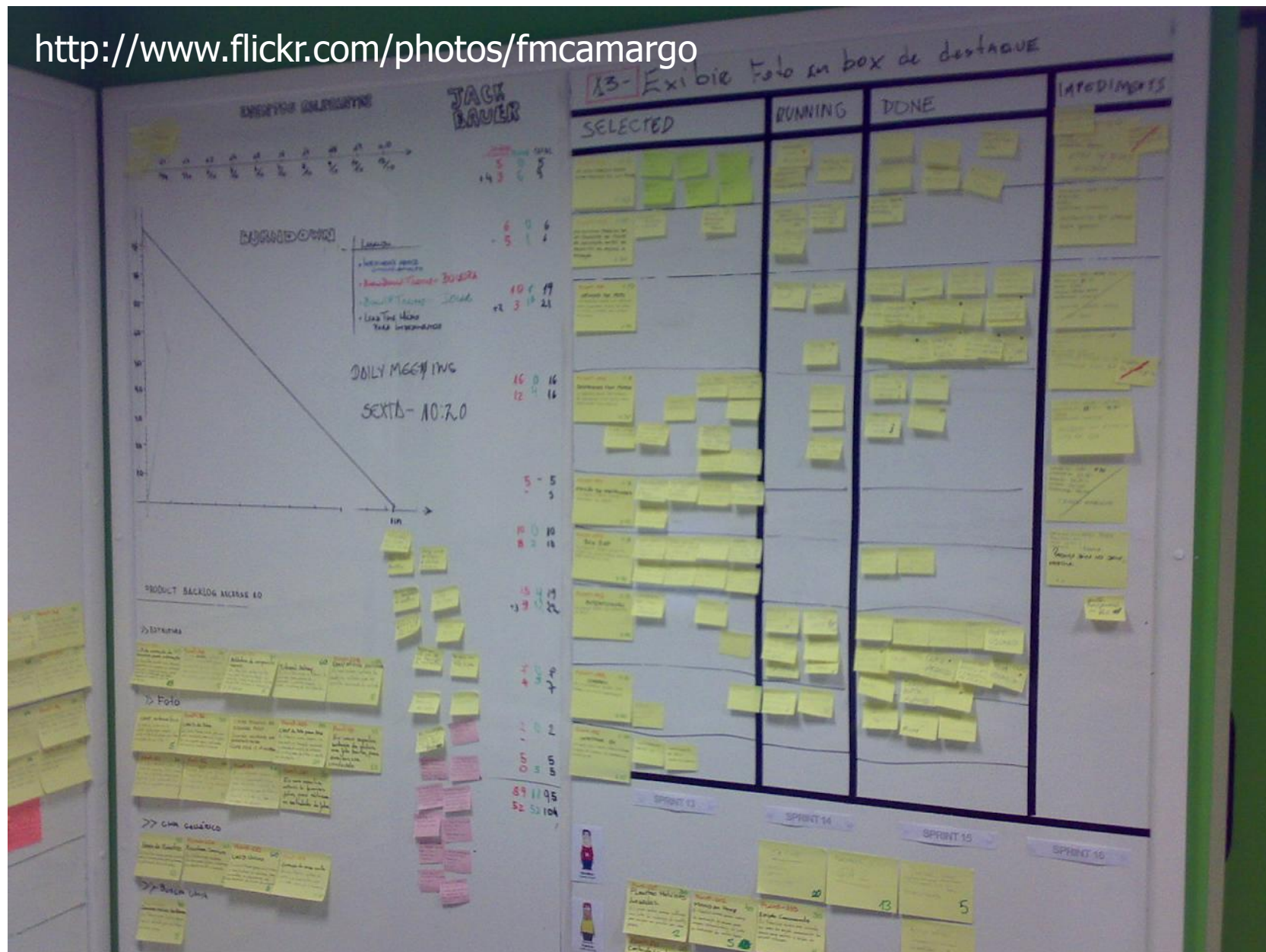


Planning and Design (cont.)

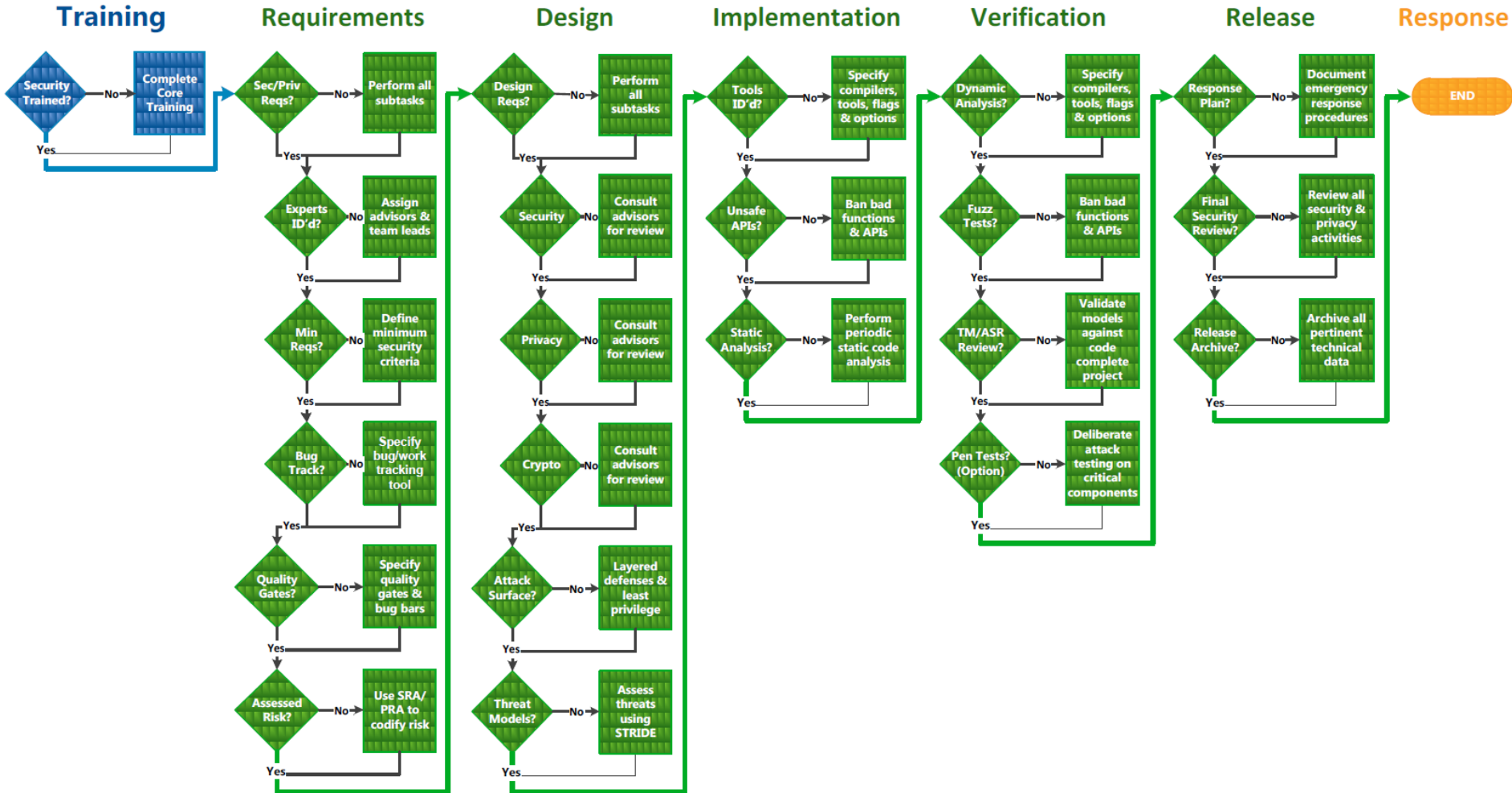


User Stories and Documentation

<http://www.flickr.com/photos/fmcamargo>



SDL SECURITY ACTIVITIES



Source: Simplified Implementation of the Microsoft SDL 



SDL Security Activities

■ Training

■ Requirements

- ▶ Security Requirements
- ▶ Quality Gates/Bug Bars
- ▶ Security and Privacy Risk Assessment

■ Design

- ▶ Design Requirements
- ▶ Attack Surface Reduction
- ▶ Threat Modeling

■ Implementation

- ▶ Use Approved Tools
- ▶ Deprecate Unsafe Functions
- ▶ Static Analysis

■ Verification

- ▶ Dynamic Program Analysis
- ▶ Fuzz Testing
- ▶ Threat Model and Attack Surface Review

■ Release

- ▶ Incident Response Plan
- ▶ Final Security Review
- ▶ Release/Archive

■ Optional Activities

- ▶ Manual Code Review
- ▶ Penetration Testing
- ▶ Vulnerability Analysis of Similar Applications

Traditional SDL Pain Points for Agile

- Can't complete all SDL activities in each sprint
- Requirements, architecture, and design evolves over time
- Threat model/documentation becomes dated quickly
- Data sensitivity, protection, and connections to third parties may not be immediately known
- Teams don't include application security specialists



Microsoft SDL For Agile Development

SDL Requirement Categories:

■ Every-Sprint

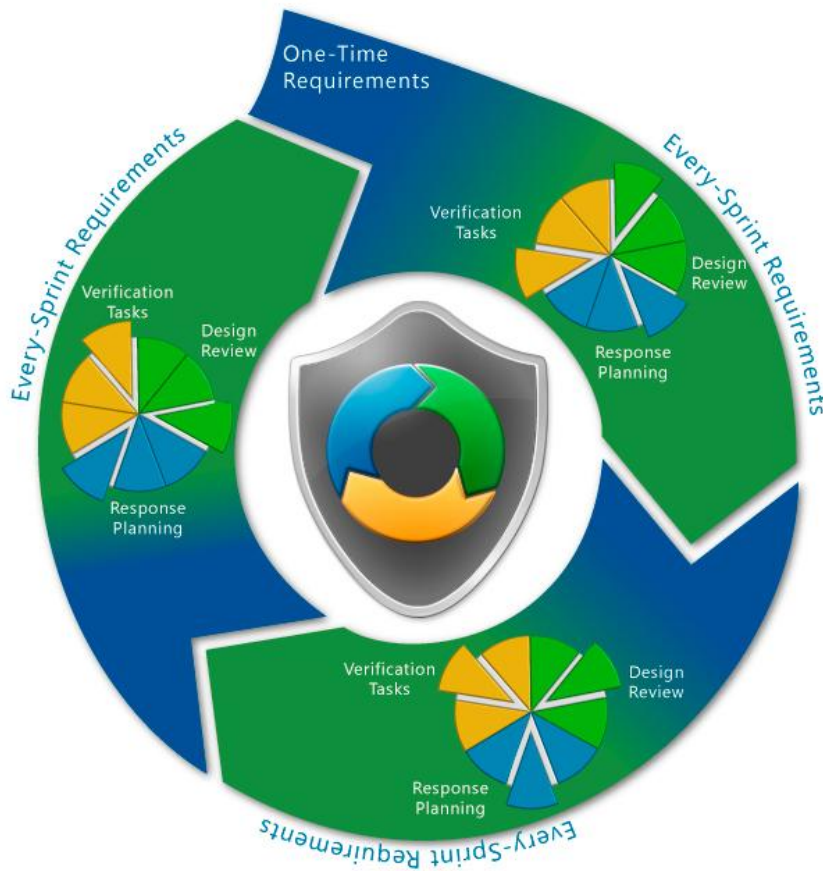
■ Bucket

- ▶ Verification Tasks

- ▶ Design Review Tasks

- ▶ Response Planning Tasks

■ One-Time



Every-Sprint SDL Requirements

“...so essential to security that no software should ever be released without these requirements being met.”

Examples:

- Update the threat model
- Communicate privacy-impacting design changes to the team's privacy advisor
- Fix all issues identified by code analysis tools for unmanaged code
- Follow input validation and output encoding guidelines to defend against cross-site scripting attacks

Bucket SDL Requirements

- Teams prioritize the pool of tasks over many sprints
- Each sprint, one task from each bucket completed
- Each tasks must be completed at least every 6 months

Examples:

- Security Verification Tasks
 - ▶ Run fuzzing tools
 - ▶ Manual and automated code review
- Design Review Tasks
 - ▶ Conduct privacy review
 - ▶ In-depth threat model
- Response Planning Tasks
 - ▶ Define security/privacy bug bar
 - ▶ Create support documents

One-Time Requirements

Why?

- Repetition not necessary
- Must occur at the beginning of the project
- Not possible at the beginning of the project

Examples:

- Configure bug tracking system (3 months)
- Identify security/privacy experts (1 month)
- Baseline threat model (3 months)
- Establish a security response plan (6 months)

SDL-Agile Appendix

Appendix Q: SDL-Agile Bucket Requirements

Bucket A: Security Verification

Title	Requirement/ Recommendation	Applies to Online Services	Applies to Managed Code	Applies to Native Code
Debug the application with the Application Verifier enabled	Requirement			X
Disable tracing and debugging in ASP.NET applications	Requirement	X	X	
Investigate and service any reported /GS crashes	Requirement			X
Perform ActiveX control fuzzing	Requirement	X		X
Perform attack surface analysis	Requirement	X	X	X
Perform binary analysis (BinScope)	Requirement	X	X	X
Perform COM object testing	Requirement			X
Perform cross-domain scripting testing	Requirement	X	X	X
Perform file fuzz testing	Requirement	X		X
Perform RPC fuzz testing	Requirement	X		X
Conduct in-depth manual and automated code review for high-risk code	Recommendation	X	X	X
Perform data flow testing	Recommendation	X	X	X

SDL-Agile Appendix: Deadlines

Appendix R: SDL-Agile One-Time Requirements

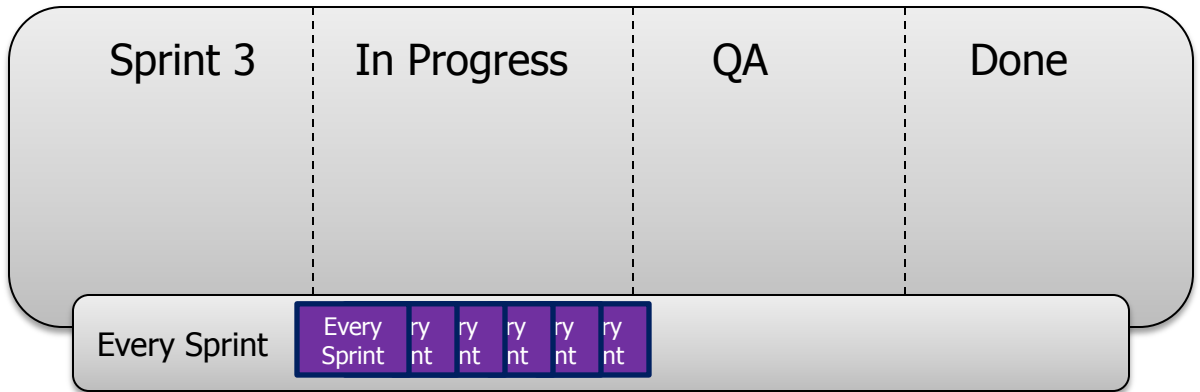
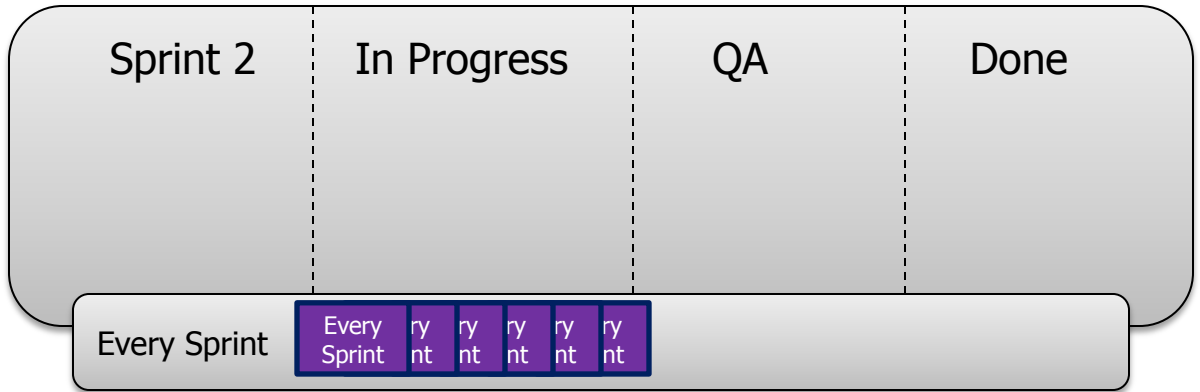
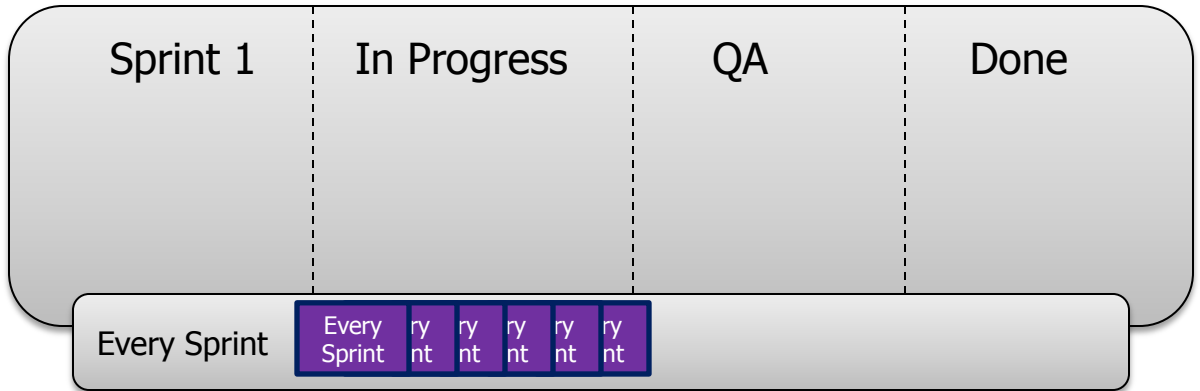
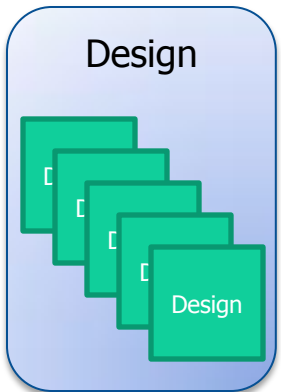
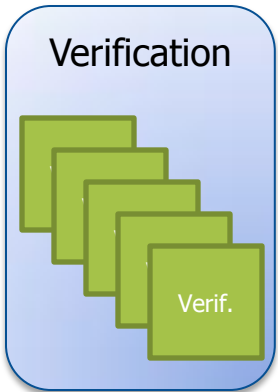
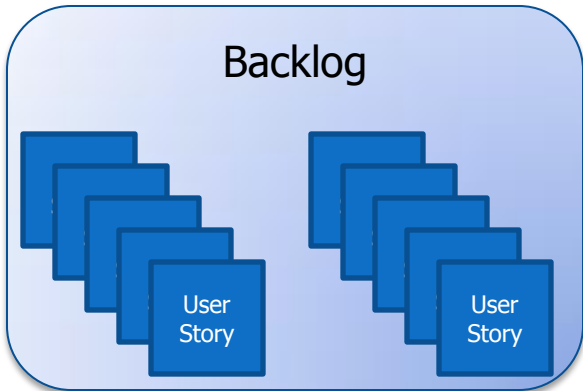
Title	Requirement/ Recommendation	Completion Deadline (months)	Applies to Online Services	Applies to Managed Code
Avoid writable PE segments	Requirement	6	X	
Configure bug tracking to track the cause and effect of security vulnerabilities	Requirement	3	X	X
Create a baseline threat model	Requirement	3	X	X
Determine security response standards	Requirement	6	X	X
Establish a security response plan	Requirement	6	X	X
Identify primary	Requirement	1	X	X

Final Security Review

- Occurs at the end of every sprint

- Checklist:

- All every-sprint requirements have been completed
- No one-time requirements have exceeded deadline
- At least one requirement from each bucket category has been completed
- No bucket requirements exceed the six month deadline
- No security or privacy bugs are open that exceed the severity threshold



Backlog

Sprint 1 | In Progress | QA | Done

User Story	er y	er y		
One-Time	f. gn	p		

Final Security Review

✓ ✓ ✓ ✓ ✓

Every Sprint | Every Sprint | nt | nt | nt | nt | nt

One-Time

Verification

Sprint 2 | In Progress | QA | Done

User Story	e e e	e e e		
Verif.	sign	p		

Final Security Review

✓ ✓ ✓ ✓ ✓

Every Sprint | Every Sprint | nt | nt | nt | nt | nt

Design

Resp. Plan

Sprint 3 | In Progress | QA | Done

User Story	r y	r y	r y	
Verif.	n n			

Final Security Review

✓ ✓ ✓ ✓ ✓

Every Sprint | Every Sprint | nt | nt | nt | nt | nt

Making SDL-Agile Manageable

- Documented standards
- Security training
- Automation
 - ▶ Continuous Integration
 - Secure Configuration
 - Security Unit Tests
 - Automated Secure Code Analysis
 - Automated Deployment and Vulnerability Scanning
- Process
 - ▶ Continuous updates to the threat model
 - ▶ SDL Process Templates for VSTS
 - ▶ MSF-Agile + SDL Process Template
- Light on security artifacts/documentation

Making SDL-Agile Manageable

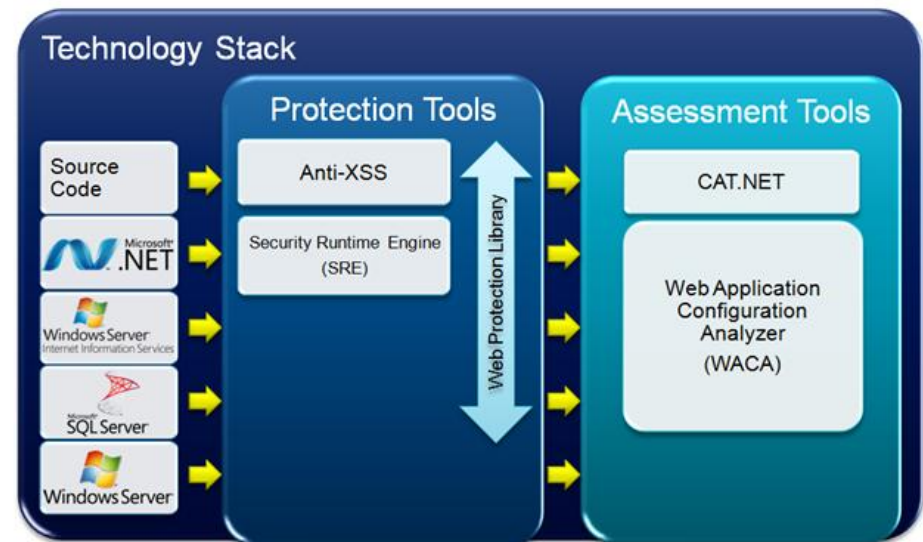
■ Tooling

▶ Code

Analysis/Scanning

- CAT.NET
- MiniFuzz
- BinScope Binary Analyzer
- Fiddler w/ Watcher
- FxCop

▶ MS Threat Modeling Tool



CAT.NET: Cross-site Scripting Vulnerability

Analysis Information

Analysis Engine Version	1.0.3455.36250
Created by	
Start time	Sunday, February 28, 2010 1:34:46 PM
Stop time	Sunday, February 28, 2010 1:34:47 PM
Elapsed time	00:00:00.6100000
Data flow graph	5 nodes, 5 edges
Targets	C:\Users\ \Desktop\UnsignedUnencryptedViewStateExploit\UnsignedUnencryptedViewStateEx

Cross-Site Scripting (ACESEC05)

1 results

Result #1

Summary

Problem	A cross-site scripting vulnerability was found through a user controlled variable that enters the application at Default.aspx.cs:21 through the variable stack1 which eventually leads to a cross-site scripting issue at Default.aspx.cs:21.		
Resolution	Use the Anti-XSS library to properly encode the data before rendering it		
Entry Variable	stack1		
Confidence	High		

Source Context	Line	Input Variable	Statement
Default.aspx.cs	21		lblPayload.Text = txtBox1.Text;
Default.aspx.cs	21	Return from TextBox.get_Text	lblPayload.Text = txtBox1.Text;

Making SDL-Agile Manageable

■ Libraries

▶ Web Protection Library (WPL)

- Encoder/~~Anti-XSS~~
Library
- Security Runtime
Engine (SRE)
- Sanitizer.GetSafeHTML

Web Protection Library - Encoder/~~Anti~~XSS

Encoder Methods

[Encoder Class](#) [See Also](#) [Send Feedback](#)

Microsoft.Security.Application.Encoder

The [Encoder](#) type exposes the following members.

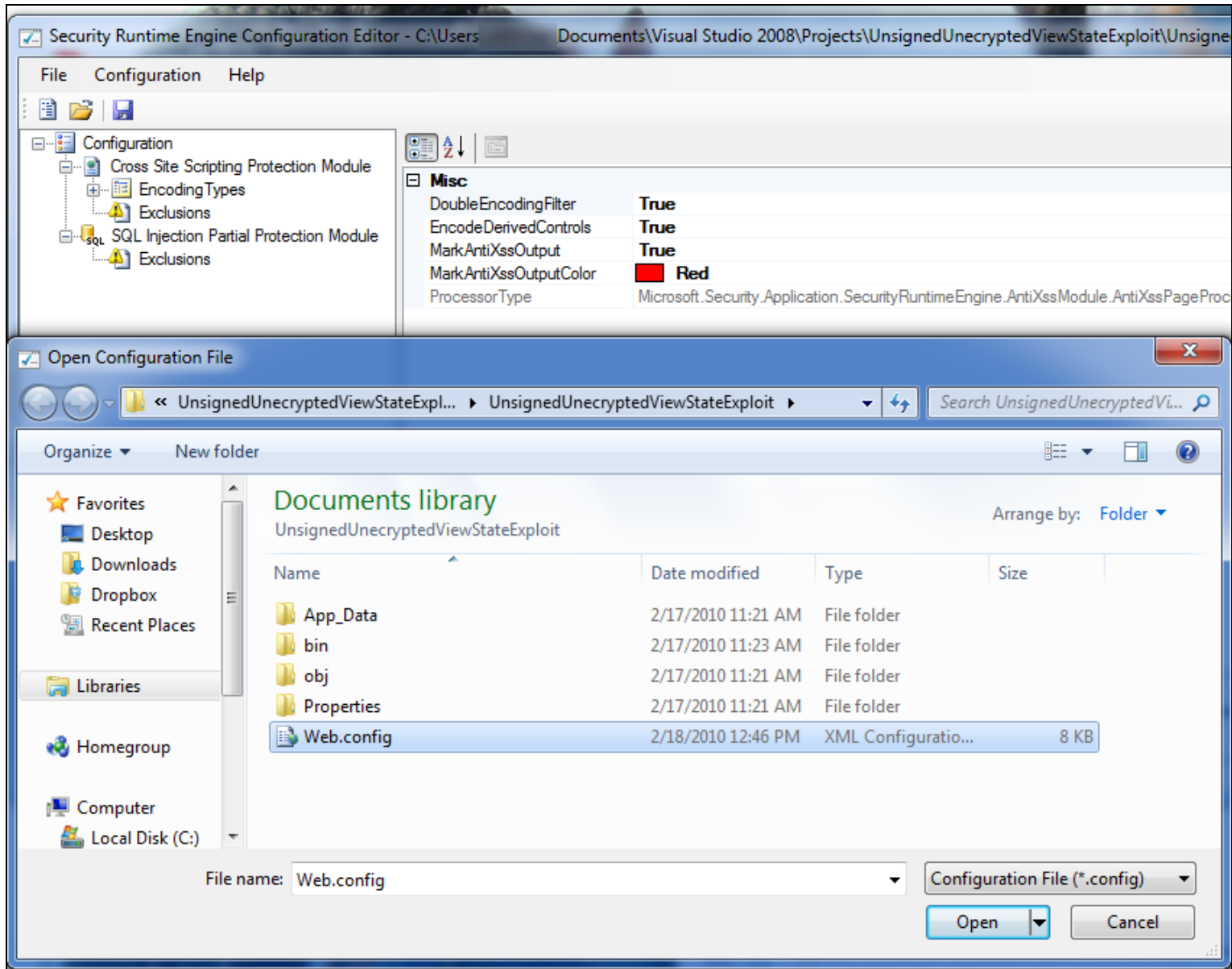
Methods

	Name	Description
⇒ S	CssEncode	Encodes input strings used in Cascading Style Sheet (CSS) elements.
⇒ S	HtmlAttributeEncode	Encodes input strings for use in HTML attributes.
⇒ S	HtmlEncode	Overloaded.
⇒ S	JavaScriptEncode	Overloaded.
⇒ S	LdapEncode	Encodes input strings used in Lightweight Directory Access Protocol (LDAP) search queries.
⇒ S	UrlEncode	Overloaded.
⇒ S	VisualBasicScriptEncode	Encodes input strings for use in Visual Basic Script.
⇒ S	XmlAttributeEncode	Encodes input strings for use in XML attributes.
⇒ S	XmlEncode	Encodes input strings for use in XML.

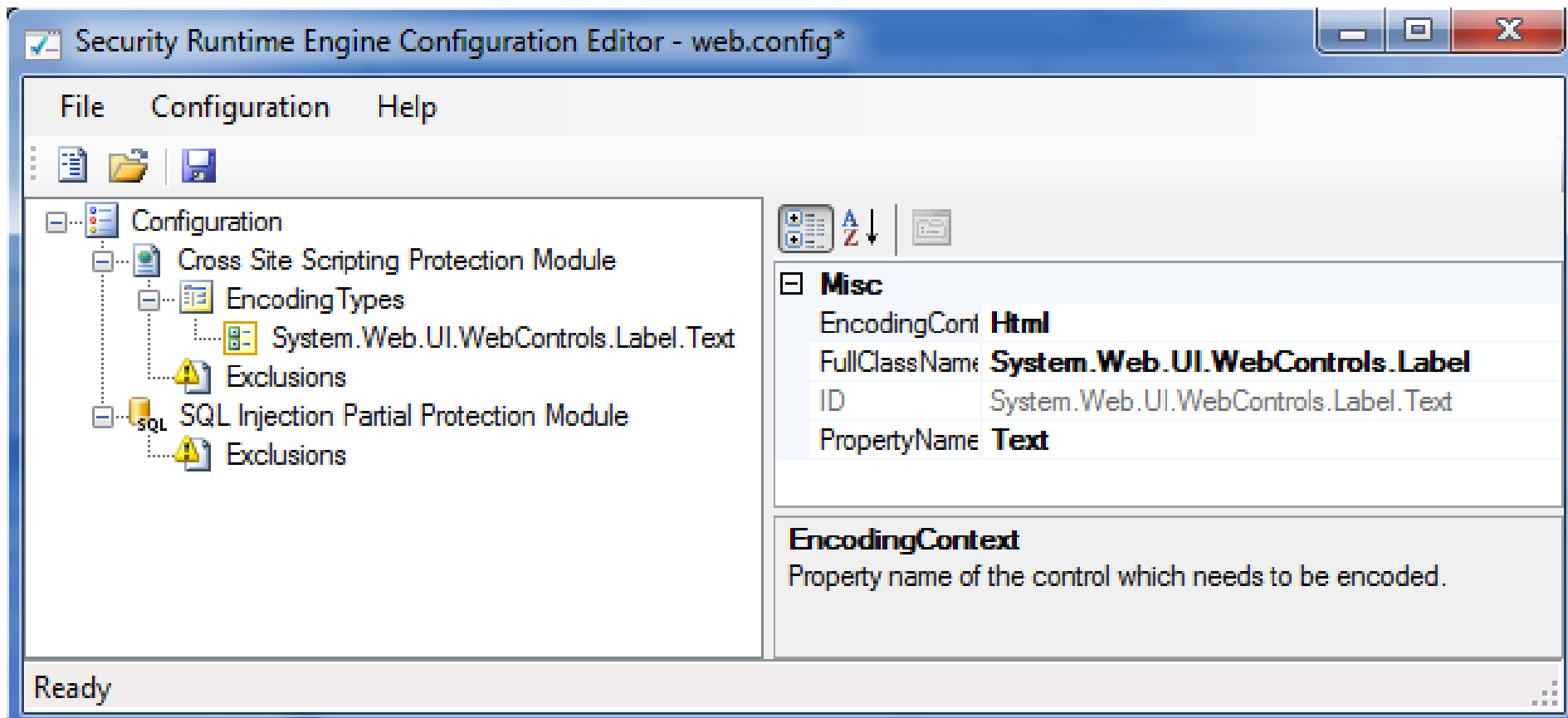
The Security Runtime Engine (SRE)

- “The Security Runtime Engine (SRE) is an HTTP module that acts like a gatekeeper to protect ASP.NET web applications from cross-site scripting (XSS) attacks.”
- “It works by inspecting each control that is being reflected by ASP.NET and then automatically encoding data of vulnerable controls in their appropriate context.”
- SRE Configuration Editor GUI Tool

The Security Runtime Engine (SRE)



The Security Runtime Engine (SRE)



The screenshot shows the Security Runtime Engine Configuration Editor window. The title bar reads "Security Runtime Engine Configuration Editor - web.config*". The menu bar includes "File", "Configuration", and "Help". The left pane shows a tree view of the configuration structure:

- Configuration
 - Cross Site Scripting Protection Module
 - EncodingTypes
 - System.Web.UI.WebControls.Label.Text
 - Exclusions
 - SQL Injection Partial Protection Module
 - Exclusions

The right pane displays the configuration for the selected item, "Misc":

EncodingCont	Html
FullClassName	System.Web.UI.WebControls.Label
ID	System.Web.UI.WebControls.Label.Text
PropertyName	Text

Below the table, the **EncodingContext** is defined as: "Property name of the control which needs to be encoded."

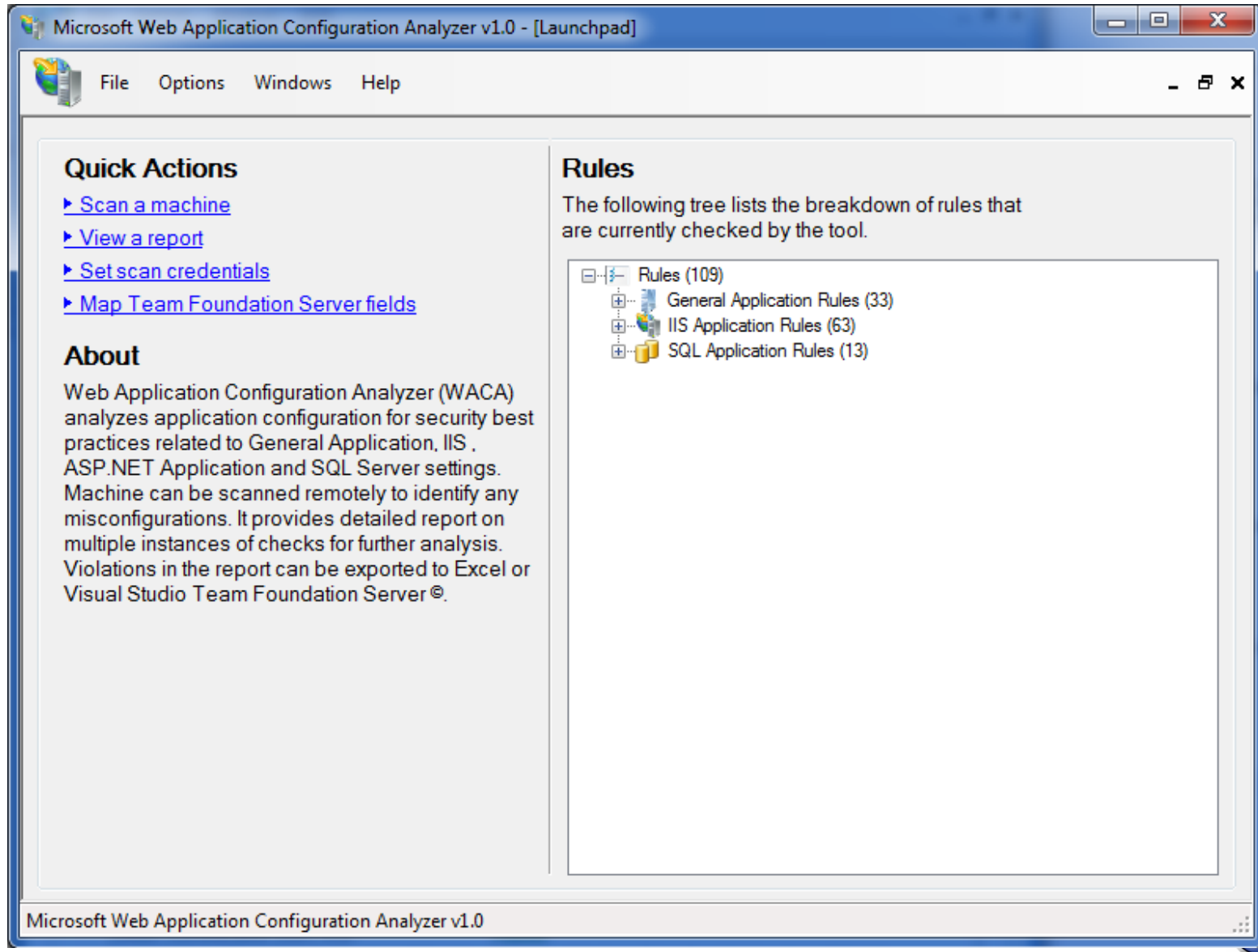
Ready

Making SDL-Agile Manageable

■ Deployment

- ▶ Web Application Configuration Analyzer (WACA)
- ▶ Microsoft Baseline Security Analyzer
- ▶ Web.config Security Analyzer (WCSA)

Web Application Configuration Analyzer



Web Application Configuration Analyzer

Microsoft Web Application Configuration Analyzer v1.0 - [New Scan 3]

File Options Windows Help

New Scan

Enter the fully qualified name of the machine to scan. Ex: server.orgunit.organization. Current user or the specified scan credentials should have administrator access to the machine. Click "Scan" to start scanning, progress of the scan will be displayed in the progress bar.

Machine Name:

SQL Instance Name:

Rule Category:

- General Application Rules
- IIS Application Rules
- SQL Application Rules

Web Application Configuration Analyzer

Microsoft Web Application Configuration Analyzer v1.0 - [Scan Report 2]

Select a scan: 28022010102558 Export to Excel Export To TFS

Scan Report for 28022010102558

Summary of the scan			
Machine Name:		Start Time:	02/28/2010 10:25:58
SQL Server:		End Time:	02/28/2010 10:27:49
General Application Rules:	55% (11 of 20) Passed		45% (9 of 20) Failed
IS Application Rules:	% (0 of 1) Passed		100% (1 of 1) Failed
SQL Application Rules:	NaN% (0 of 0) Passed		NaN% (0 of 0) Failed
All Rules:	52.38% (11 of 21) Passed		47.62% (10 of 21) Failed

General Application Rules

1	Is Drive (C:\) Partition a Non-NTFS File System?	No, Passed	
	Is Drive (D:\) Partition a Non-NTFS File System?	Yes, Failed	System partition is a Non-NTFS partition.
	Is Drive (E:\) Partition a Non-NTFS File System?	No, Passed	
	Is Drive (N:\) Partition a Non-NTFS File System?	Yes, Failed	System partition is a Non-NTFS partition.
2	Is guest account enabled on the server?	No, Passed	
3	Does Windows Drive Root Folder (A:\) NTFS ACL contain Everyone ACE?	No, Passed	
	Does Windows Drive Root Folder (C:\) NTFS ACL contain Everyone ACE?	No, Passed	
	Does Windows Drive Root Folder (E:\) NTFS ACL contain Everyone ACE?	No, Passed	
	Does Windows Drive Root Folder (N:\) NTFS ACL contain Everyone ACE?	No, Passed	
4	Does Windows Drive Root Folder (A:\) NTFS ACL contain Authenticated Users ACE?	No, Passed	
	Does Windows Drive Root Folder (C:\) NTFS ACL contain Authenticated Users ACE?	Yes, Failed	Windows Drive Root Folder (C:\) NTFS ACL contains ACE for Authenticated Users
	Does Windows Drive Root Folder (E:\) NTFS ACL contain Authenticated Users ACE?	Yes, Failed	Windows Drive Root Folder (E:\) NTFS ACL contains ACE for Authenticated Users
	Does Windows Drive Root Folder (N:\) NTFS ACL contain Authenticated Users ACE?	No, Passed	

Close


Microsoft Web Application Configuration Analyzer v1.0



Web.config Security Analyzer (WCSA)

Web.config Security Analyzer v1.0.2

File Help External Resources Exit

 **H-Security Labs**
Mesut Timur
<http://www.h-labs.org>
<http://www.webguvenligi.org>

web.config Security Report

There are 6 security issues found.

Path : C:\Users\... Documents\Visual Studio 2008\Projects\UnsignedUnencryptedViewStateExploit\UnsignedUnencryptedViewStateExploit\Web.config

1. Debugging Enabled

You have enabled debugging on your application. This could be used by attackers to extract useful information such as detailed information and stack trace, etc production environment.

Secure Configuration

```
<configuration>  
<system.web>  
  <compilation debug="false">
```

References

- <http://msdn.microsoft.com/en-us/library/s10awwz0.aspx>
- <http://www.developerfusion.com/article/6745/top-10-application-security-vulnerabilities-in-webconfig-files-part-two/6/>
- <http://www.acunetix.com/vulnerabilities/ASP.NET-debuoaigna-enabled.htm>

2. Web cookies are not HttpOnly

Your configuration allows cookies to be be accessed from client-side scripting technologies. This can lead an attacker to perform Cross-Site Scripting and session

Secure Configuration

```
<configuration>  
<system.web>  
  <httpCookies httpOnlyCookies="true">
```

References

- <http://msdn.microsoft.com/en-us/library/ms228262.aspx>
- <http://www.owasp.org/index.php/HTTPOnly>
- <http://www.developerfusion.com/article/6678/top-10-application-security-vulnerabilities-in-webconfig-files-part-one/5/>

3. Web cookies doesn't require SSL

Your configuration allows web cookies to be transferred in clear-text form. An attacker that can be able to monitor your network, can easily steal your session.

Secure Configuration



Making SDL-Agile Manageable

- **Education, secure coding standards, automation** and **tools** play a significant role in making secure Agile development efficient and economical
- Don't forget:
 - ▶ Periodic manual security activities are also a must
 - ▶ All of this must fit within a **repeatable, mature process**

Summary and Questions

More Information:

<http://www.microsoft.com/sdl>

Nick Coblentz, CISSP

Senior Consultant, AT&T Consulting

Nick.Coblentz@gmail.com

<http://nickcoblentz.blogspot.com>

<http://www.twitter.com/sekhmetn>

- Microsoft releases SDL-Agile Guidance in Nov. 2009
- Treats SDL Activities like team-prioritized User Stories
 - ▶ 3 Categories: One-time, Every-time, and Bucket
- Increased success with the implementation of training, automation, tools, and standards