Proactive risk mitigation within the Software Development Lifecycle (SDLC)

Real world examples that have worked for me, …

Joe White, CISSP, CSSLP
joe@cyberlocksmith.com
@cyberlocksmith

20+ years technical experience

10+ years security experience

Hands-on, real world pen testing experience

Built application security programs at two organizations.

OWASP NYC (2008)
“Web Application Security Roadmap”

SDLC building blocks

What worked and what did not work (+/-)

Details of published SDLC process

<disclaimer>

The examples and references in this presentation are real examples from a real-life security practitioner.

Names have been changed to protect the innocent.

No animals were harmed in the making of this presentation

And of course, YMMV !

</disclaimer>

SDLC building blocks

Supporting quotes and research (+)
Secure Coding Guidelines (-)
Secure Coding checklist (+)
Non Functional Requirements (++)
Static Code Analysis (+)
Security Awareness Training (++)
Threat Modeling (+/-)
Application Security Risk Matrix (++)
Published SDLC (++)

Recommended:
Center of Excellence (++)

NIST (2002): The Economic Impacts of Inadequate Infrastructure for Software Testing

http://www.nist.gov/director/planning/upload/report02-3.pdf
(Section 5, page 4)

"Vulnerabilities within application code are exponentially more expensive to address the farther in the development lifecycle they are found. In fact, the National Institute of Standards and Technology (NIST) estimates that code fixes performed after release can result in 30 times the cost of fixes performed during the coding/development phase into the report."

Aberdeen Group (2010): Security and the Software Development Lifecycle: Secure at the Source

http://www.microsoft.com/download/en/details.aspx?id=6968

"companies adopting the secure at the source strategy – i.e., the integration of secure application development tools and practices into the software development lifecycle, to increase the elimination of security vulnerabilities before applications are deployed – found that they realized a very strong 4.0-times return on their annual investments in application security, higher than that of the Industry Average and higher than that of both the find and fix and defend and defer approaches."

Secure Coding Guidelines (-)

overlooked by developers

“static and not helpful”

100+ pages that can be language specific.

most surprising discovery over the last 5 years.

Secure Coding Checklist (+)

Use while NFR still under development

Simple 1-2 page document

Useful if combined with a peer code review prior to code check-in.

Non-Functional Requirements (++)

Most effective of all building blocks

‘Container’ for other SDLC building blocks.

Can include application security guidelines, secure coding checklist, security policies, etc.

Effective NFRs will document the requirement *and* explain why the requirement is necessary.

Static Code Analysis (SCA) (+)

SDLC requires SCA

Must be baked into acceptance criteria for code to leave the SDLC.

Assurance to QA that code is ready for testing

Security awareness training (++)

Instructor-led training

Very useful for educating on attack techniques and unexpected behavior

Rewards for eLearning

Your goal should be to provide anyone that can influence application security, e.g. project managers, development managers, application developers, QA, etc. with the training, awareness and resources they need to be successful.

Threat modeling (+/-)

Hit or miss at most locations

Can be informal process

Combines nicely with NFRs

Discussing NFR may lead to threat modeling discussion

Application Security Risk Matrix (++)

External facing	Data: Non sensitive	Data: sensitive
Internal facing	Data: Non sensitive	Data: sensitive

If you do not have a published SDLC for your organization then you will <u>NOT</u> be successful.

SDLC Phases

Planning
Design
Construct
Accept
Close

Phases will need artifacts

Be flexible

Center of Excellence (++)

COE Steering Committee

COE Drivers

COE Members

Remove barriers between departments

Positively impact change

SDLC building blocks

Supporting quotes and research (+)
Secure Coding Guidelines (-)
Secure Coding checklist (+)
Non Functional Requirements (++)
Static Code Analysis (+)
Security Awareness Training (++)
Threat Modeling (+/-)
Application Security Risk Matrix (++)
Published SDLC (++)

Recommended:
Center of Excellence (++)