Eoin Fitzpatrick
@Celtikill | celtikill@celtikill.io

- Autodidact 4 lyfe.
- Software Security
- Startups . . . Fortune 500s
- Likes Hackers, Brazilian Jiu Jitsu
Credits

- **Gary McGraw**
  - Silver Bullet podcast ([https://www.cigital.com/podcast/](https://www.cigital.com/podcast/))

- **Pravir Chandra**
  - youtubes.

- **Sunny Wear**
  - Local training

- **US CERT**
  - [https://buildsecurityin.us-cert.gov/](https://buildsecurityin.us-cert.gov/)
Software Assurance

The OWASP way

Note the license:

● The OpenSAMM is licensed under CC Attribution-ShareAlike v. 3.0.
● This work is licensed under the 4.0 version of the same license.

All images used were marked for reuse, none were modified from their original form.
Objective

Narrative overview.

Quick comprehension.

Introduce the model, then briefly illustrate each practice.

Goals:

● Enable quick understanding of Software Assurance for a wide audience.
● Inspire the adoption of assurance activity.
Why Care?
Why care?

Compliance has noticed.

Sadly, this is important to the adoption of Assurance in the SDLC:

- A group of buzzards is called a “wake”.
- The buzzards stick around when the herd’s experiencing significant loss

PCI-DSS 3.1 section 6.3:
“Without the inclusion of security during the requirements definition, design, analysis, and testing phases of software development, security vulnerabilities can be inadvertently or maliciously introduced into the production environment.”

HIPAA Security Rule:
- Speaks specifically to NIST.
- NIST SP 800-64 reads “To be most effective, information security must be integrated into the SDLC from its inception.“
- It’s said so since 2003.
Why care?

No clear perimeter.

Not simple.

Trends in software:
- Perimeter dissolves
- Complexity increases
- Extensibility increases
Why care?

Risk Convergence.

Software assurance in general addresses:

● Behavior
● Development of software

There are other places where this convergence exists . . .
Why OpenSAMM?
“It Depends.”

- ISO, IEEE are not freely available.
- The freely-available consensus standards evolve faster, and do map up.

Options:

- **BSIMM6**
  - Descriptive
  - Answers “weakest zebra” question (McGraw)
  - More detailed, software focused

- **OpenSAMM**
  - Prescriptive
  - Great marketing collateral . . .
  - Easy to consume at all levels
Model Intro
Open Software Assurance Maturity Model

Assurance

Deeds not words. Deeds not tools.

Assurance = **Integrity**.

Saying “We care about security” . . . and proving it with our actions.
Maturity

Quantify progress.

Apply flexibly.

How much assurance do we really have?
“All models are wrong . . .”

“. . . but some are useful.”

- George Box

Zen moment: models are not reality.

George Box has been called "one of the great statistical minds of the 20th century". Use a Bayesean classifier?

https://en.wikipedia.org/wiki/George_E._P._Box
OpenSAMM

Look at the back of the handouts...
Activity Overview
Governance
Difference between fear and risk?

Have we described the demons? Do we have a plan to escape them?

- Record your fears (define risk profile).
- Have a roadmap from fear to assurance (OpenSAMM roadmap).
- Inform with data (internal and external).

“The Triumph of Death” by Pieter Bruegel the Elder. Depicts 16th century security fears.

Who are your risk managers?

- Senior, Strategic Leadership.
Policy & Compliance
Understand the vision, make it your own, ensure it’s not a fantasy.

- Understand and meet external requirements.
- Drive internal security standards.
- Maintain integrity through audit.

Regulatory framework is an imperfect vision of security, set forth by well meaning people. They are a place to begin, not a place to end.
- Policy and Standard for SECURITY and regulation.
Education & Guidance

- Offer help mitigating security bugs and flaws
  - Resources, guides advice
  - Coaching
- Continue education

When venturing into new territory, it’s easy to feel stranded. **Don’t drop a development team in the wilderness without a guide.**
- Do Technical Security Training
- Build Library of Technical Guidance

- Do Role-Specific Training
- Provide Expert Consulting Service

- Create Support Portal
- Do Role-based Certification
Remember the demons? The Soviet Union was one of them post WWII (arguably). **U2 was commissioned by CIA to better understand Soviet capability and intention. This info was applied to national security.**

Note: Threat assessment is NOT threat intelligence(TM). It is threat intelligence applied to the development of a system. [Deeds not tools!]

Post-WWII, only visual intelligence was from German spy planes. U2 could fly above fighters, radar and SAM...
- Build Threat Models
- Profile Attackers

- Develop Abuse Cases
- Rate and Weight Threats

- Risk-rate Third-party Components
- Consider Compensating Controls in Models
Specify the expected behaviors within a system.

How should they interact with each other, how should they interact with actors.
- Write Security into Business Requirements
- Require Best Practice

- Understand Access Control
- Match security requirements to risk profile

- Write security requirements into 3rd-party agreements
- Add security requirements to audits
Define how you build the thing . . . Securely. This house had a blueprint, but . . .
“We use this framework”
“We build in this way”

threat model = the cold. Architects said “because cold, 2” extra foundation, special joints for the roof, and R32 minimum insulation ratings. Put your pipes in the middle of the house”

Did the architect then leave the project? No.
<table>
<thead>
<tr>
<th>Secure Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SA 1</strong></td>
</tr>
<tr>
<td>● List recommended software frameworks</td>
</tr>
<tr>
<td>● Apply security principles checklist to design</td>
</tr>
<tr>
<td><strong>SA 2</strong></td>
</tr>
<tr>
<td>● ID &amp; Promote shared security services</td>
</tr>
<tr>
<td>● ID &amp; promote secure design patterns</td>
</tr>
<tr>
<td><strong>SA 3</strong></td>
</tr>
<tr>
<td>● Make Formal reference architecture</td>
</tr>
<tr>
<td>● Validate usage of frameworks, patterns, architecture</td>
</tr>
</tbody>
</table>
Verification
Consider design and architecture of built software.

This is an exercise in detecting flaws:

- does the design reflect the purpose of security?
- Is the vision of architecture represented in the substance of what which is created.

Lee Green, VP at IBM. In quote, he speaks of the Eames, who were credited with inspiring IBM's design philosophy.


**Legs = Security mechanism**
- Identify Attack Surface
- Match Design to Requirements

- Inspect the Application of security mechanisms
- Provide Design Review Service

- Diagram key data flows
- Make release gates for design review
Inspect the code for bugs.

Focus efforts based on risk.
- Create review checklist from requirements
- Do reviews of high-risk code

- Utilize automated code review tools
- Do code review in development

- Tun code review for application specifics
- Make release gates for code review
Security Testing

- Inspect the monster in its runtime environment, living and breathing.
- Question and test assumptions made in building it.

No matter how good the Dr.’s intentions, the monster never quite cooperates...
Security Testing

**ST 1**
- Make test cases from requirements
- Pen test on release

**ST 2**
- Utilize automated testing tools
- Test security in development

**ST 3**
- Tune security tests to application specifics
- Make release gates for security testing
Vulnerability Management

- Define bat-team
- Provide bat-signal
- Do bat-things

Handling vulnerabilities, and handling incidents (two sides, same coin)
- Define Roles (the PM receives the report, the team lead forms the incident response group)
- Define the bat-signal.
Vulnerability Management

**VM 1**
- Identify point of contact for project
- Make loose incident response team

**VM 2**
- Define incident response process
- Adopt a disclosure process

**VM 3**
- Do root cause analysis on incidents
- Measure trends in incidents
Ensure the underlying infrastructure is configured securely.

- Maintain a configuration spec (project group communicates effectively, expressing the configurations needed to run the app securely)
- Manage patches (with communication…)

What happens when Infrastructure and the Project don’t communicate? Patches break apps, and apps confound infrastructure.

Note that infrastructure is much more than this: From the perspective of software projects, they train the horse, and infrastructure puts the shoes on it.
- Specify assumptions of operating environment
- Do security patching

- Establish routine patch process
- Monitor baseline configurations

- ID and deploy protective controls
- Audit baselines for configuration and patching
Communicate best way to use the software.

- Gather information critical to using an app securely.
- Share that information to the appropriate people (users, operators, administrators).
- Record security-related configuration info
- Write procedures for alerts

- Have change management procedures
- Maintain operational security guidelines

- Audit for operational enablement
- Do code signing
Case Study
Fin.