Secure Development initiative

An effort to empower development teams

Jenny Marie Ellingsæter
Sopra Steria Cyber Security Centre
Why a Secure Development initiative?

Some observations from the InfoSec department

- Development teams expected InfoSec team to take care of security.
- Pentesting as a last resort before release – causing delays.
  - Vulnerabilities not fixed before production
- Pentesting results revealed obvious security flaws and bugs.
  - Some teams did much better than others.
- Pentesting was effective for finding bugs, but not necessarily design flaws.
  - Pentesters did not have sufficient time to learn the product of domain.
  - Development teams have solid product insight and domain knowledge.
- Challenging for the development teams to get time to fix vulnerabilities.
  - Increased focus on time to market – shorter iterations and quicker deliverables.
- Fixing vulnerabilities late in the development process is expensive:
  - There may be multiple dependencies at this point
  - Other tasks will be delayed – causing the project to be delayed.
  - Decreasing the likelihood of vulnerabilities getting fixed.
A major incident occurs

The InfoSec department gets funding.
The Secure Development initiative roadmap

1. Survey amongst development teams
2. Workshops with development teams
3. Created Secure SDLC framework based on Microsoft SDL
4. Created guidelines to support the Secure SDLC
5. Conducted benchmark surveys
6. On-site training in risk assessment and threat modelling
7. Invested in Software Security eLearning platform
8. Appointed & trained Security Champions
Fact finding – surveys and workshops

**Surveys**
- Structured feedback
- Challenging to understand the answers
  - Asking the right questions in the right way
  - No context

**Workshops**
- Got better insight
  - Context and more details
- Unstructured
  - Some shared a lot
  - Others barely said a word.
  - Easy to be influenced by the loud ones...

---

**Some of our findings**

**Environment**
- No common development methodology
- Multiple programming languages
- Wide range of technologies

**Security practices**
- Security training was scarce
- All did code reviews – but not necessarily with security in mind
- Security requirements were left up to the development teams to define
- Apart from pen testing by InfoSec, few other security activities took place.
The Secure Development initiative roadmap

1. Survey amongst development teams
2. Interviews of development teams
3. Created Secure SDLC framework based on Microsoft SDL
4. Created guidelines to support the Secure SDLC
5. Conducted benchmark surveys
6. On-site training in risk assessment and threat modelling
7. Invested in Software Security eLearning platform
8. Appointed & trained Security Champions
Secure SDLC activities

Predict
Understand your risks, know your attack surface, uncover weak spots

Prevent
Minimise attack surface, prevent incidents

React
React to breaches, mitigate the damage, analyse and learn

Detect
Recognise incidents and threats, isolate and contain them.

• Core Security training
• Establish Security Requirements
• Threat Modelling
• Risk Assessment

• Security Tool Assessment
• Secure Coding Standard
• Define Security Acceptance Criteria
• Security testing
• Code review
• Security Review Gate
• Assurance of Third Party Code
• Remediation plan

• Incident Response Plan

• Operational enablement
Secure SDLC activity structure

• Description of the activity

• Trigger – e.g. changes in architecture, new functionality added, time

• Objective – e.g. ensuring development team has necessary competence

• Deliverable – e.g. documented security requirements, proof of training

• Maturity levels, based on a baseline model:

<table>
<thead>
<tr>
<th>Level 0</th>
<th>The activity is neither implemented or executed on a regular basis.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>The activity is to some extent implemented, executed and maintained.</td>
</tr>
<tr>
<td>Level 2</td>
<td>The activity is implemented, executed on a regular basis and continuously maintained.</td>
</tr>
<tr>
<td>Level 3</td>
<td>Results from executing the activity provides improvement feedback to other relevant activities. Knowledge and good practices are shared with the organisation.</td>
</tr>
</tbody>
</table>
Benchmarking

Objective

Visibility into current maturity state, and way to measure progress at:

- Team level
- Department level
- Organisation level
- Vendors/partners

Maturity model inspired by OpenSAMM and BSIMM
The Secure Development initiative roadmap

Survey amongst development teams

Interviews of development teams

Created Secure SDLC framework based on Microsoft SDL

Created guidelines to support the Secure SDLC

Conducted benchmark surveys

On-site training in risk assessment and threat modelling

Invested in Software Security eLearning platform

Appointed & trained Security Champions
Security Champions

Security to developer ratio: 1:100

Our objectives:
• Build security competence in development teams
• Ensure continuous focus on security (and privacy)
• InfoSec “satellites”
• Participants in the Software Security community

Security Champion:
• Interested in security
• Eager to learn
• Good communicator and promoter
• With a bit of backbone

Training:
• Primarily pentesting
The Secure Development initiative roadmap

Survey amongst development teams
- Interviews of development teams
- Created Secure SDLC framework based on Microsoft SDL
- Created guidelines to support the Secure SDLC

Conducted benchmark surveys
- On-site training in risk assessment and threat modelling
- Invested in Software Security eLearning platform
- Appointed & trained Security Champions
Then what?

Nothing much happened.

- Benchmarking surveys showed little progress
- Software Security Community was held alive by InfoSec
- Surprisingly few requests for assistance related to the S-SDLC
- eLearning platform used primarily to achieve compliance with training requirements in PCI
  - (Which was part of the objective)
What’s going on?

Culture and management commitment takes time.

- Business priorities are measured (KPIs).
- Those KPIs do not include security.
A major incident

Seen from the trenches of incident response
- cartoon style
Livet, Det
Forbannede
A few days later

Support? I'm having some trouble with my pwd...
What went wrong?

- Logs existed allowing forensic investigation
- System designed based on complete trust of staff.
- System probably not security assessed on a regular basis
- There was no monitoring of the logs, or triggers defined to detect the malicious/abnormal behaviour
- Users received no receipt when their passwords were changed.
Discussion

What security activities might have helped prevent the incident?

- Training and awareness?
- Identification of Security requirements?
- Risk assessment?
- Threat modelling?
- Security testing?
- Incident response plan?
- Code review?
Last words

• Software security is a cultural thing and management commitment is key.

• Focusing on the security in the product is not enough
  • Someone needs to manage the product
  • The infrastructure and development tools we use may be our weak points

• Competence trumps tools.
Thanks for your attention!