1000 Projects later

Security Code Scans at SAP

OWASP
The Open Web Application Security Project

About Us

OWASP
The Open Web Application Security Project

- Ruediger Bachmann
  is a Development Architect at SAP AG working, as member of the central code analyses team, in the areas application security and code analysis.

- *Dr. Achim D. Brucker*
  is a Senior Researcher in the Security & Trust Program of SAP Research as well as a member of the central code analysis team of SAP AG. His research interests includes static and dynamic security testing.

SAP

Agenda

OWASP
The Open Web Application Security Project

Why is SAP using Static Code Analysis?

Secure Development Lifecycle at SAP

Static Code Analysis at SAP

Challenges and Outlook

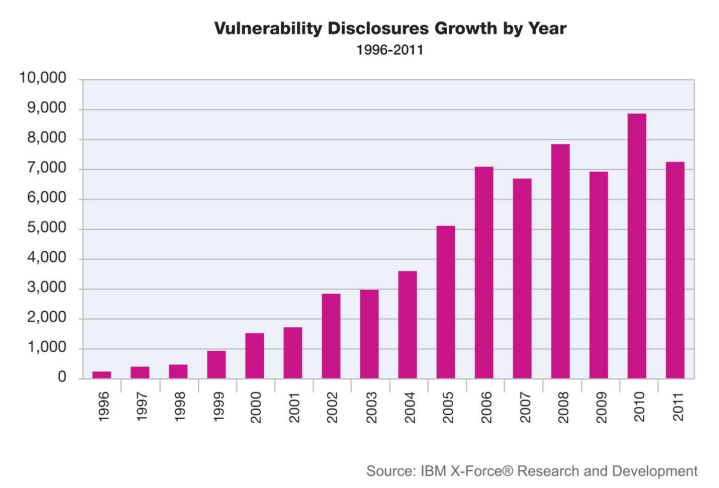
Security Code Scans at SAP Overview

OWASP
The Open Web Application Security Project

- Started rollout in June 2010
- Centrally guided by a project team
  - Definition of Security Requirements
  - Establishment of Scan Infrastructure
- Support of the most important languages
- SAP development and third party code

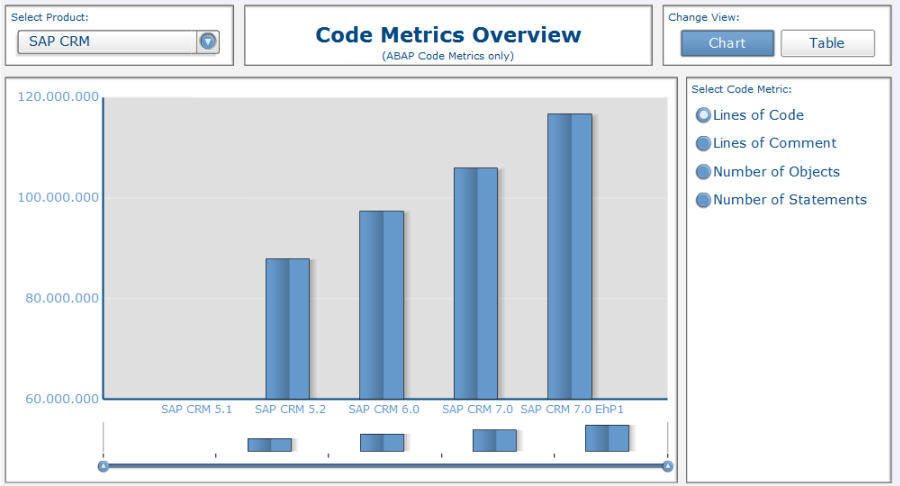
Insecure Software

OWASP
The Open Web Application Security Project



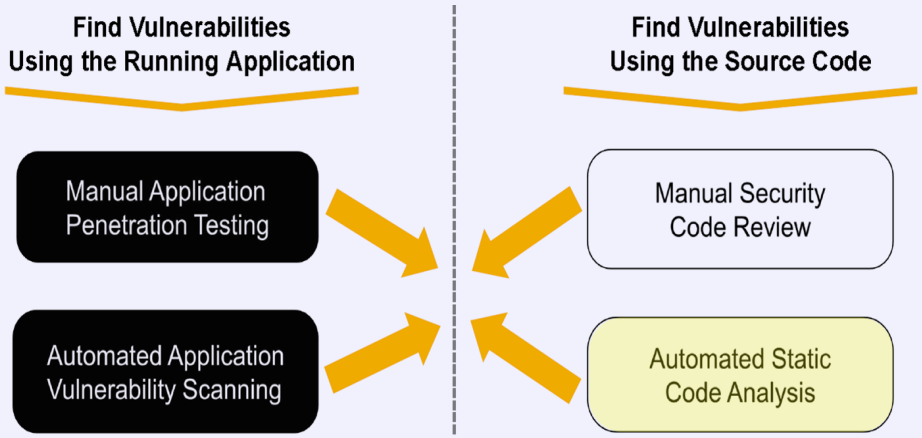
Evolution of Code

OWASP
The Open Web Application Security Project



Security Testing

OWASP
The Open Web Application Security Project



Dynamic Security Testing

OWASP
The Open Web Application Security Project

- Characteristics
  - Black box approach
  - Sends input to applications and analyses response
- Advantages
  - Provides concrete examples (attacks)
  - Analyze dataflows accross multiple components
- Disadvantages
  - Coverage unclear
  - Requires test system

Static Security Testing

OWASP The Open Web Application Security Project

- Characteristics
  - White box approach
  - Analyses abstraction of the source (binary)
- Advantages
  - Explores all data paths / control flows
  - Can analyse single modules (unit test)
- Disadvantages
  - High false positive rate (not exploitable findings)
  - Does not consider application environment

Agenda

OWASP The Open Web Application Security Project

Why is SAP using Static Code Analysis?

Secure Development Lifecycle at SAP

Static Code Analysis at SAP

Challenges and Outlook

First Step: Security Training

OWASP The Open Web Application Security Project

- Education:
  The prerequisite for achieving a high security quality
- Security awareness:
  Reducing the number of “built-in” security problems
- Trained persons:
  Analyze and fix vulnerabilities much more efficiently
- Trainings:
  Secure Programming, Build & Scan, Auditing, ….

Secure Development Lifecycle (SDLC) at SAP

OWASP The Open Web Application Security Project

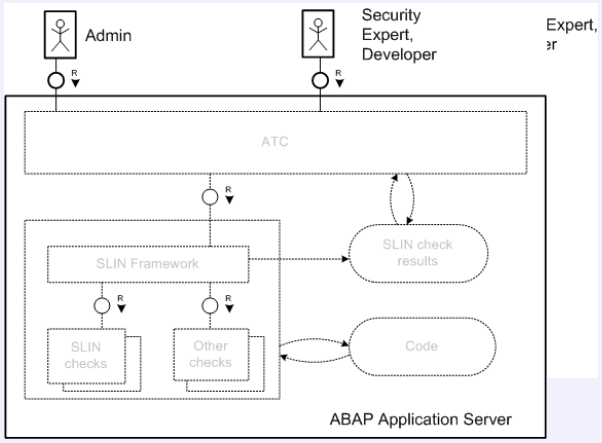
- Structure the investment of time and resources
  - to safeguard a high level of security
  - to ensure security standards across all areas
- Security requirements
  - are taken into account and
  - are implemented

  in all phases of product development

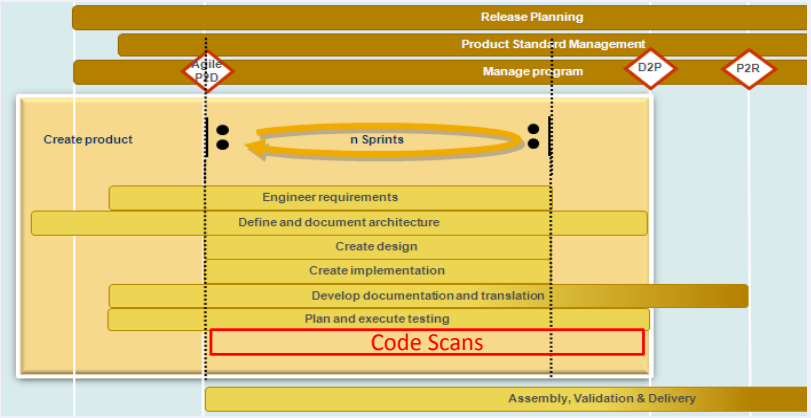
The Different Roles

- Developer
  - fixes software security issues
- Security Expert
  - review scan results, decides on fixes
- Build Master
  - scans the source code, manages results
- Scrum Master
  - requests scan, assigns vulnerabilities to developers

Infrastructure



SAP Secure Software Development Life Cycle



For passing D2P Q-gate, evidence has to be provided that the source code has been scanned and exploitables have been fixed.

P2D: Planning to Development. / D2P: Development to Production. /

P2R: Production to Ramp-up (gradual roll-out to customers).

Third Party Code

- Third party code
  - Open Source libaries and frameworks
  - Freeware
  - other third party components
- Different approaches
  - SAST analysis by SAP
  - Certificate from vendor
  - SLA with vendor

Agenda

OWASP The Open Web Application Security Project

Why is SAP using Static Code Analysis?

Secure Development Lifecycle at SAP

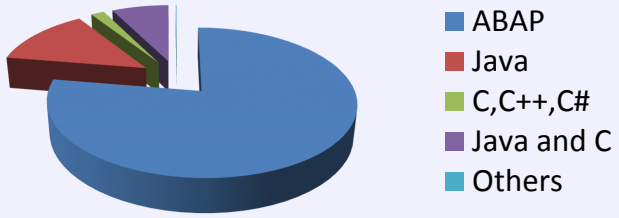
Static Code Analysis at SAP

Challenges and Outlook

Code Scan Facts

OWASP The Open Web Application Security Project

- Over 2000 developers are using SAST tools
- Over 500 MLOC scanned



Statistics Jan 2012

Security Scan Tools used at SAP

OWASP The Open Web Application Security Project

Language	Scan Application
ABAP	SAP
C/C++	Coverity
Others	HP/Fortify

Security Requirements

OWASP The Open Web Application Security Project

- SAP on Corporate Security Requirements
  - SAP Applications shall be free of backdoors
  - SQL injection vulnerabilities shall be avoided
  - Cross-Site Scripting vulnerabilities shall be prevented
  - Directory traversal vulnerabilities shall be prevented
  - The system shall be protected against buffer overflow vulnerabilities
- OWASP Top 10
- CWE/SANS Top 25 2011
- CVE

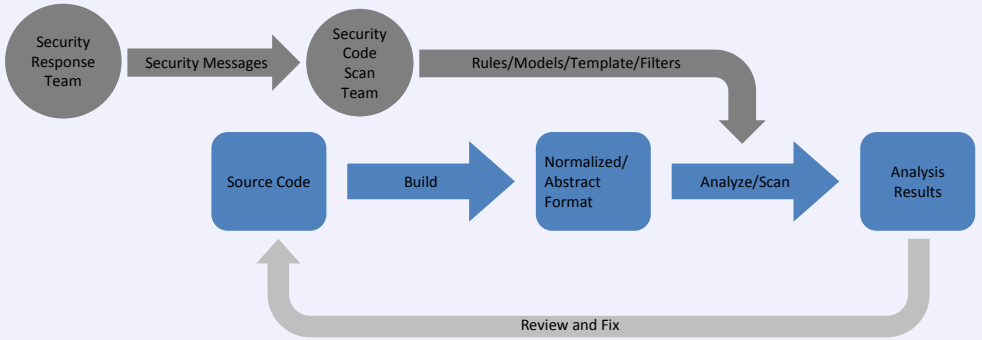
Continuous Improvement

OWASP The Open Web Application Security Project

- Collect feedback from the
  - Product Security Response Team
  - Development Teams
- Develop rules/models to improve the scans
- Continuously improve the infrastructure
- Continuously improve the rollout process

Input to Improve Code Scans

OWASP The Open Web Application Security Project



•Further input channels: Development teams, internal research, scan reviews, code reviews

Lessons Learned

OWASP The Open Web Application Security Project

- Scans have to be obligatory
  but not introduced 'brute force'
- Establish Secure Development Life Cycle
  make scans a natural part of development
- Plan carefully
  – Do not start with scans right before Dev. Close
  – Do it regularly (nightly)
- Do not introduce changes during development

Agenda

OWASP The Open Web Application Security Project

Why is SAP using Static Code Analysis?

Secure Development Lifecycle at SAP

Static Code Analysis at SAP

Challenges and Outlook

Challenges

JavaScript I
Unerstand the DOM

- Assume the following index.html:

```
<TITLE>Welcome!</TITLE>
Hi
  <SCRIPT>
     var pos=document.URL.indexOf("name=")+5;
     document.write(document.URL.substring(pos,document.URL.length));
  </SCRIPT>
Welcome to our system
```

and a call

```
index.html?name=<script>alert(document.cookie)</script>
```

- resulting in a DOM-based XSS attack
- DOM implementations are Browser specific

JavaScript II
Dynamic Evaluation

- A simple script statement

```
<script language="javascript">
   document.write("<script>src='other.js'></script>");
</script>
```

- Dynamically creating script tags

```
<script>
  var oHead = document.getElementsByTagName('HEAD').item(0);
  var oScript= document.createElement("script");
  oScript.type = "text/javascript";
  oScript.src="other.js";
  oHead.appendChild( oScript);
</script>
```

- Or using eval() directly (not shown here)

JavaScript III
Server-Side JavaScript

- Combining the comlexity of two worlds

```
var entry=JSON.parse(data);
query = "insert into \"FOO(".NAME")\";
var conn = $.db.getConnection();
conn.execute(query);
```

Challenges: Current Trends

OWASP
The Open Web Application Security Project

- SAST works very well for
  - "traditional" programming languages
  - Analyzing data paths within one technology
- Many new development uses JavaScript
  - HTML5/JavaScript UIs
  - Server-side JavaScript
- JavaScript
  - Untyped
  - Dynamic programming model

Conclusion

OWASP
The Open Web Application Security Project

You cannot pay people well enough, to do proper code audits.

I tried.

(Yaron Minsky, Jane Street Capital)

OWASP
The Open Web Application Security Project

Thank you!