

# Deploying SAST on a Large Scale

Achim D. Brucker

joint work with Uwe Sodan

{achim.brucker, uwe.sodan}@sap.com

SAP AG, Germany

# Has Sony been Hacked this Week?

<http://hassonybeenhackedthisweek.com/>

---

Time-line of the Sony Hack(s) (excerpt):

2011-04-20 Sony PSN goes down

2011-05-21 Sony BMG Greece: data 8300 users (SQL Injection)

2011-05-23 Sony Japanese database leaked (SQL Injection)

2011-05-24 Sony Canada: roughly 2,000 leaked (SQL Injection)

2011-06-05 Sony Pictures Russia (SQL Injection)

2011-06-06 Sony Portugal: SQL injection, iFrame injection and XSS

2011-06-20 20th breach within 2 months

177k email addresses were grabbed via a SQL injection

(<http://hassonybeenhackedthisweek.com/history>)

## Consequences:

- account data of close to 100 million individuals exposed
- over 12 million credit and debit cards compromised
- more than 55 class-action lawsuits
- costs of \$ 170 million only in 2011

# A Bluffers Guide to SQL Injection

---

- **Assume an SQL Statement for**

---

*selecting all users with "userName" from table "user"*

---

# A Bluffers Guide to SQL Injection

---

- **Assume an SQL Statement for**

```
stmt = "SELECT * FROM 'users' WHERE 'name' = '" + userName + "';"
```

# A Bluffers Guide to SQL Injection

---

- **Assume an SQL Statement for**

```
stmt = "SELECT * FROM 'users' WHERE 'name' = '" + userName + "';"
```

- **What happens if we choose the following *userName*:**

```
userName = "' or '1'='1"
```

# A Bluffers Guide to SQL Injection

---

- **Assume an SQL Statement for**

```
stmt = "SELECT * FROM 'users' WHERE 'name' = '" + userName + "';"
```

---

- **What happens if we choose the following *userName*:**

```
userName = "' or '1'='1"
```

---

- **Resulting in the following statement:**

```
stmt = "SELECT * FROM 'users' WHERE 'name' = '' or '1'='1';"
```

---

# A Bluffers Guide to SQL Injection

- **Assume an SQL Statement for**

```
stmt = "SELECT * FROM 'users' WHERE 'name' = '" + userName + "';"
```

- **What happens if we choose the following *userName*:**

```
userName = "' or '1'='1"
```

- **Resulting in the following statement:**

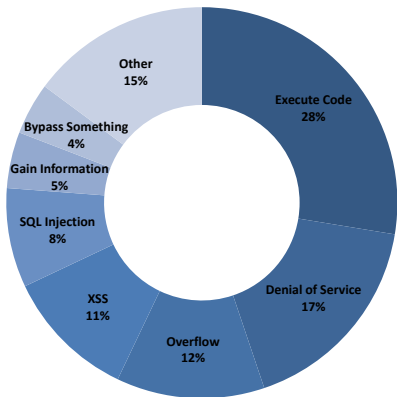
```
stmt = "SELECT * FROM 'users' WHERE 'name' = '' or '1'='1';"
```

- **Which is equivalent to**

```
stmt = "SELECT * FROM 'users';"
```

selecting the information of **all users** stored in the table 'users'!

# Vulnerability types of CVE reports since 1999

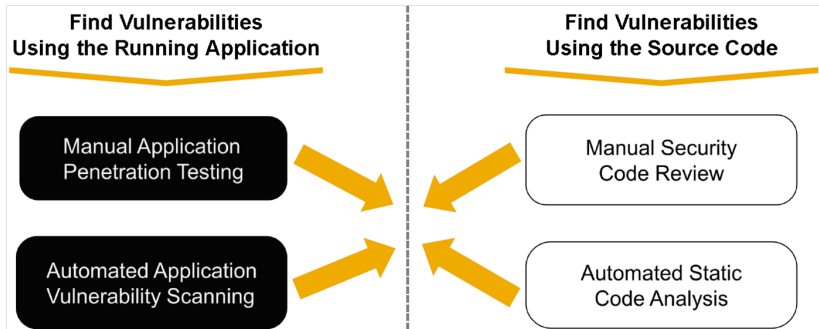


- Causes for most vulnerabilities are
  - programming errors
  - configuration errors
- Patching is
  - expensive
  - may introduce new bugs

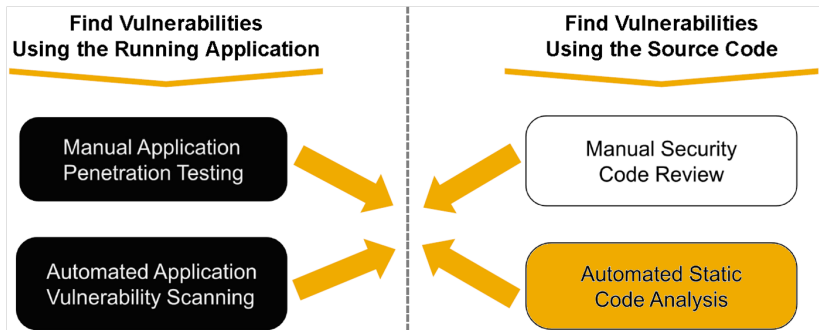
How can we ensure that no vulnerable code is shipped?



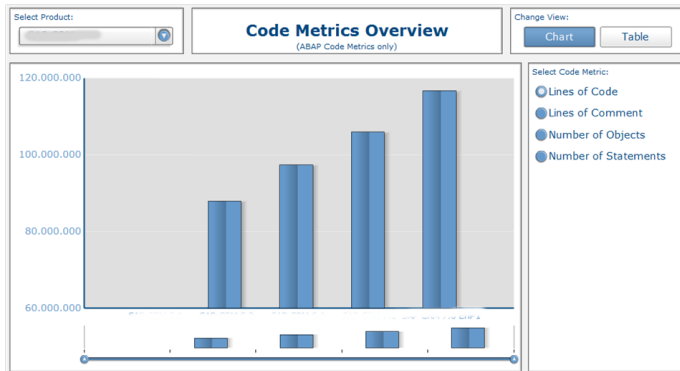
# Finding Security Vulnerabilities



# Finding Security Vulnerabilities



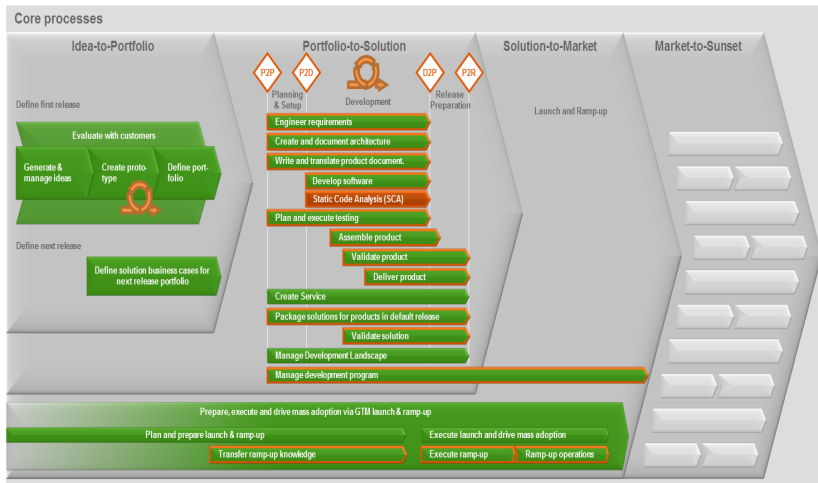
# Evolution of Source Code



- Increase in

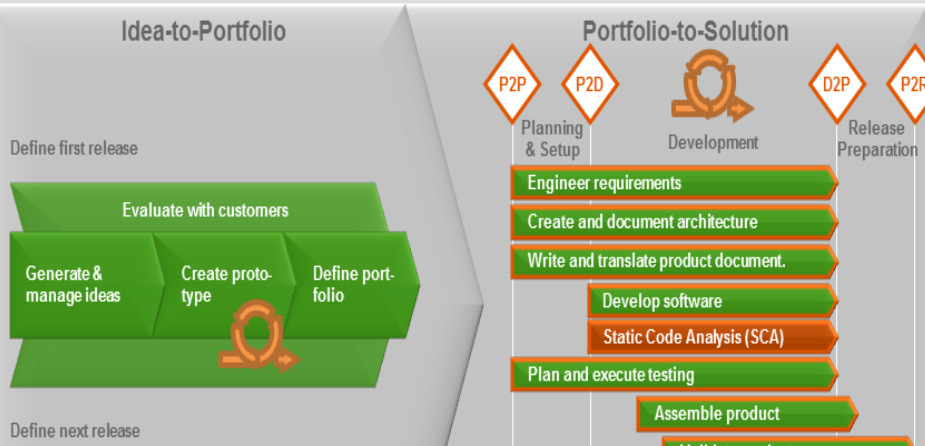
- code size
- code complexity
- number of products
- product versions

# SAST as Part of SAP's SDL



# SAST as Part of SAP's SDL

## Core processes



# So Everything is Secure Now, Right?

---

“

Our tool reports all vulnerabilities in your software – you only need to fix them and you are secure.

Undisclosed sales engineer from a SAST tool vendor.

# So Everything is Secure Now, Right?

“

Our tool reports all vulnerabilities in your software – you only need to fix them and you are secure.

Undisclosed sales engineer from a SAST tool vendor.

Yes, this tools exists! It is called Code Assurance Tool (cat):

# So Everything is Secure Now, Right?

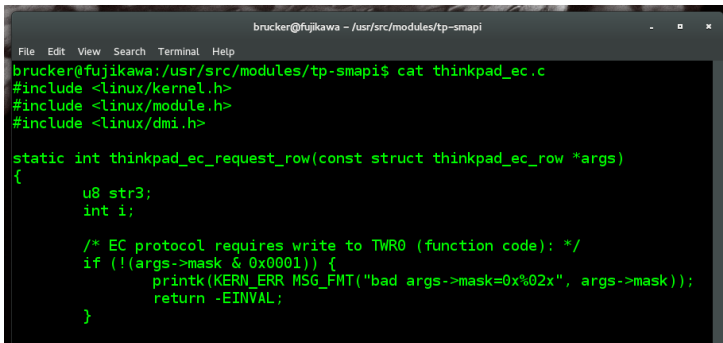
“

Our tool reports all vulnerabilities in your software – you only need to fix them and you are secure.

Undisclosed sales engineer from a SAST tool vendor.

Yes, this tool exists! It is called Code Assurance Tool (cat):

- The cat tool reports each line, that might contain a vulnerability:

A screenshot of a terminal window with a dark background and light green text. The window title is 'brucker@fujikawa - /usr/src/modules/tp-smapi'. The terminal shows the command 'cat thinkpad\_ec.c' and its output, which is the content of the file. The code includes headers for kernel, module, and dmi, and defines a function 'thinkpad\_ec\_request\_row' with a buffer overflow vulnerability. The vulnerability is located in the 'if' statement where 'args->mask' is used without bounds checking.

```
brucker@fujikawa: /usr/src/modules/tp-smapi
File Edit View Search Terminal Help
brucker@fujikawa:/usr/src/modules/tp-smapi$ cat thinkpad_ec.c
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/dmi.h>

static int thinkpad_ec_request_row(const struct thinkpad_ec_row *args)
{
    u8 str3;
    int i;

    /* EC protocol requires write to TWR0 (function code): */
    if (!(args->mask & 0x0001)) {
        printk(KERN_ERR MSG_FMT("bad args->mask=0x%02x", args->mask));
        return -EINVAL;
    }
}
```



# So Everything is Secure Now, Right?

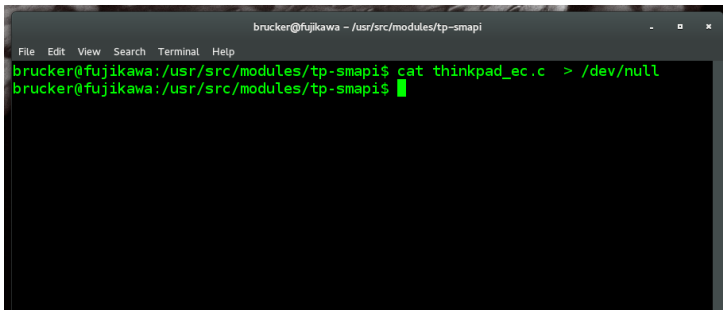
“

Our tool reports all vulnerabilities in your software – you only need to fix them and you are secure.

Undisclosed sales engineer from a SAST tool vendor.

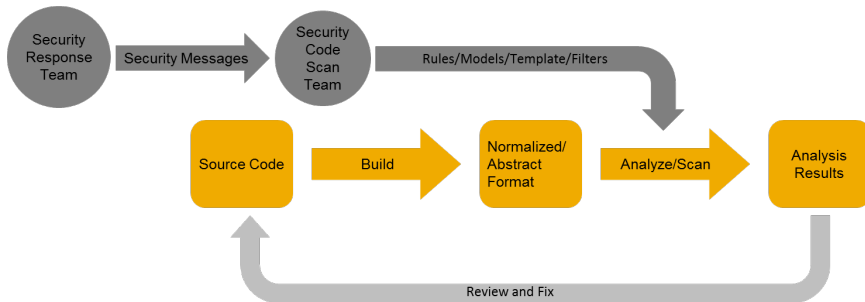
Yes, this tool exists! It is called Code Assurance Tool (cat):

- The cat tool reports each line, that might contain a vulnerability:
- It supports also a mode that reports **no false positives**:



```
brucker@fujikawa - /usr/src/modules/tp-smapi
File Edit View Search Terminal Help
brucker@fujikawa:/usr/src/modules/tp-smapi$ cat thinkpad_ec.c > /dev/null
brucker@fujikawa:/usr/src/modules/tp-smapi$
```

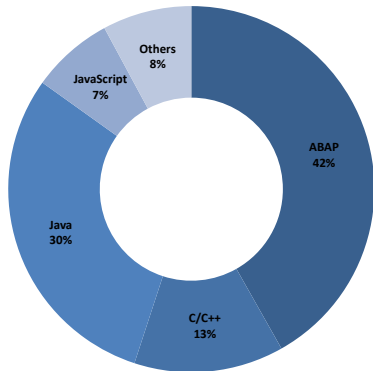
# Continuous Improvements



Further input channels:

- Development teams
- Internal research
- Scan reviews
- Code reviews
- ...

# SAST Solutions Applied at SAP



- Mandatory for all products
- Multiple billions lines analyzed

Language	Tool	Vendor
ABAP	CVA (SLIN_SEC)	SAP
C/C++	Coverity	Coverity
Others	Fortify	HP

Other important pillars of SAP's SDL:

- Secure programming training's
- Pen tests on the final product

In addition:

- Own research (e.g., JS, Mobile)
- Ongoing evaluation of
  - alternative tools and
  - complementary techniques.

# Open Issues

---

- Estimating the risk of not fixing security issues is hard
  - How to prioritize security vs. functionality
  - In case of doubt, functionality wins
- Pushing SAST across the software supply chain
  - Consumed software (OSS, third-party products)
  - SAP Customers, partners, and OEM products
- Huge and hybrid multi-language applications
  - Client-server applications
  - Web-frameworks
- Dynamic programming paradigms and languages
  - JavaScript, Ruby, etc.
- Lack of standardized regression test suites
  - Different tools
  - Different versions of the same tool

# Lessons Learned: Recommendations (1/3)

---

Follow the recommendations given by Chandra et al:

- Start small
  - Start with one pilot
  - Succeed with pilot before larger roll-out
- Go for the throat
  - Start with the main security threat
- Appoint a champion
  - Identify a developer that knows all parts of the application
  - Make this developer your tool champion
- Measure the outcome
  - Track and measure the generated data
- Make it your own
  - Adapt the tool to your needs
  - SAST tools are not “off-the-shelf” products

## Lessons Learned: Recommendations (2/3)

---

Based on our experiences, we add:

- Plan and invest enough resources
  - Introducing SAST requires significant resources
  - Integration, Analysis, Education, . . .
- Plan and invest enough infrastructure
  - If the tools are slow, nobody will use them
- Do understand your developers as your friends
  - Do not follow the “security review” model
  - SAST tools should be understood as “debug tool”
- Execute scans regularly
  - SAST is not a one-time effort

## Lessons Learned: Recommendations (3/3)

---

- Plan your changes and updates
  - All changes to the tools might change the results
- Do get support (and commitment) from your management
  - Introducing SAST will cost money and effort
  - Minimize the risk of discussing “security vs. features”
- Do not stop here.
  - Introducing SAST is only the first step
  - Use complementary techniques, e.g.,
    - Threat modeling
    - Dynamic testing tools
    - Penetration tests
    - ...

# Conclusion

---

“ You cannot pay people well enough, to do proper code audits.  
I tried it.

Yaron Minsky, Jane Street Capital

- We can confirm the results of Scandariato et al that show that SAST is the most effective and efficient security testing method
- Embed your SAST efforts into a **holistic security testing** strategy



# Thank you!



<http://xkcd.com/327/>

# Bibliography I

---



Ruediger Bachmann and Achim D. Brucker.

Developing secure software: A holistic approach to security testing.

*Datenschutz und Datensicherheit*, March 2014.



Achim D. Brucker and Uwe Sodan.

Deploying static application security testing on a large scale.

In *gi Sicherheit 2014*, Lecture Notes in Informatics. gi, March 2014.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice. Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation. IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries. Oracle is a registered trademark of Oracle Corporation. UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group. Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc. HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology. Java is a registered trademark of Sun Microsystems, Inc. JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape. SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company. Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary. The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence. The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.