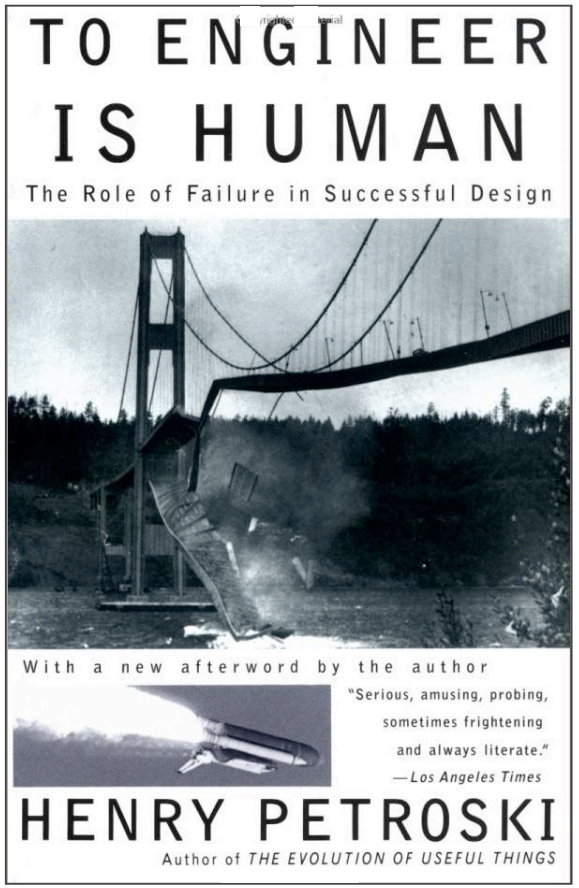
Selling Security to Software Developers

Lessons learned building a commercial static analysis tool

Brian Chess
Founder/Chief Scientist
4/12/07

Success is foreseeing failure.

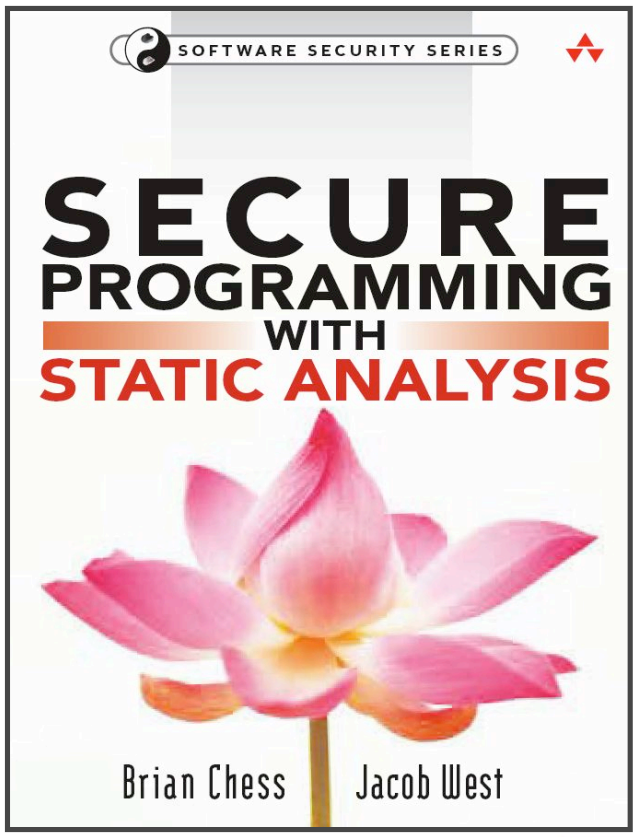
– Henry Petroski



FORTIFY SOFTWARE

Overview

- **Static analysis tools**
  - What makes a commercial tool tick
  - What to search for?
- **Selling static analysis**
  - Customers
  - Competitors
- **Hard problems, real and imagined**



FORTIFY SOFTWARE

Static Analysis

Static Analysis

- **Bugs-to-breaches abound:**
  - Stop playing catch-up: fix security problems before deployment.
  - As a group, developers make the same mistakes over and over.
- **Static analysis is good for security:**
  - Easy to explore boundary conditions and states that may be hard to achieve through testing.
  - Catch common mistakes automatically.

Classic fingerd buffer overflow

```
char line[512];
gets(line);
```

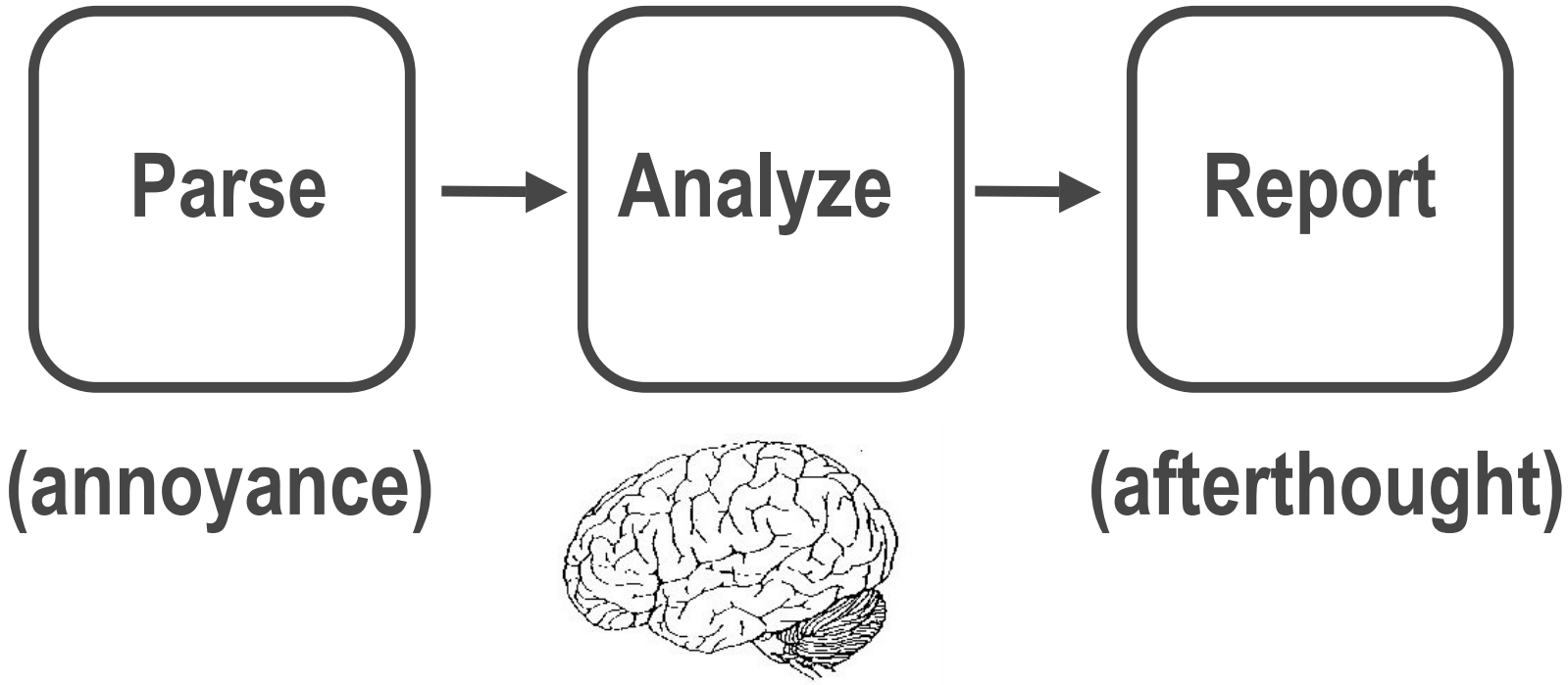
Common Errors

MSDN sample code for function DirSpec:

```
int main(int argc, char *argv[]) {
  ...
  char DirSpec[MAX_PATH + 1];
  printf ("Target dir is %s.\n", argv[1]);
  strncpy (DirSpec, argv[1], strlen(argv[1])+1);
```

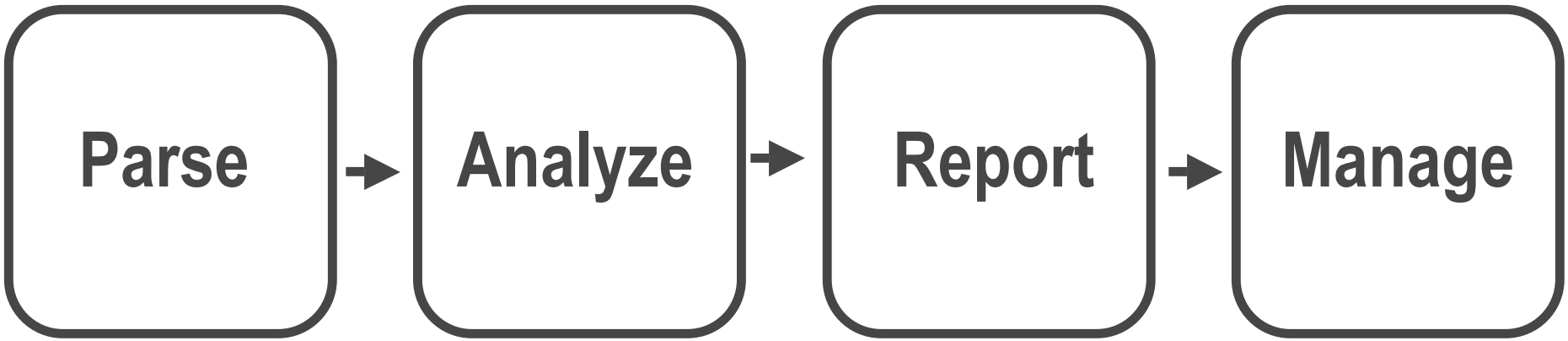
A peek inside a static analysis tool

The academic perspective



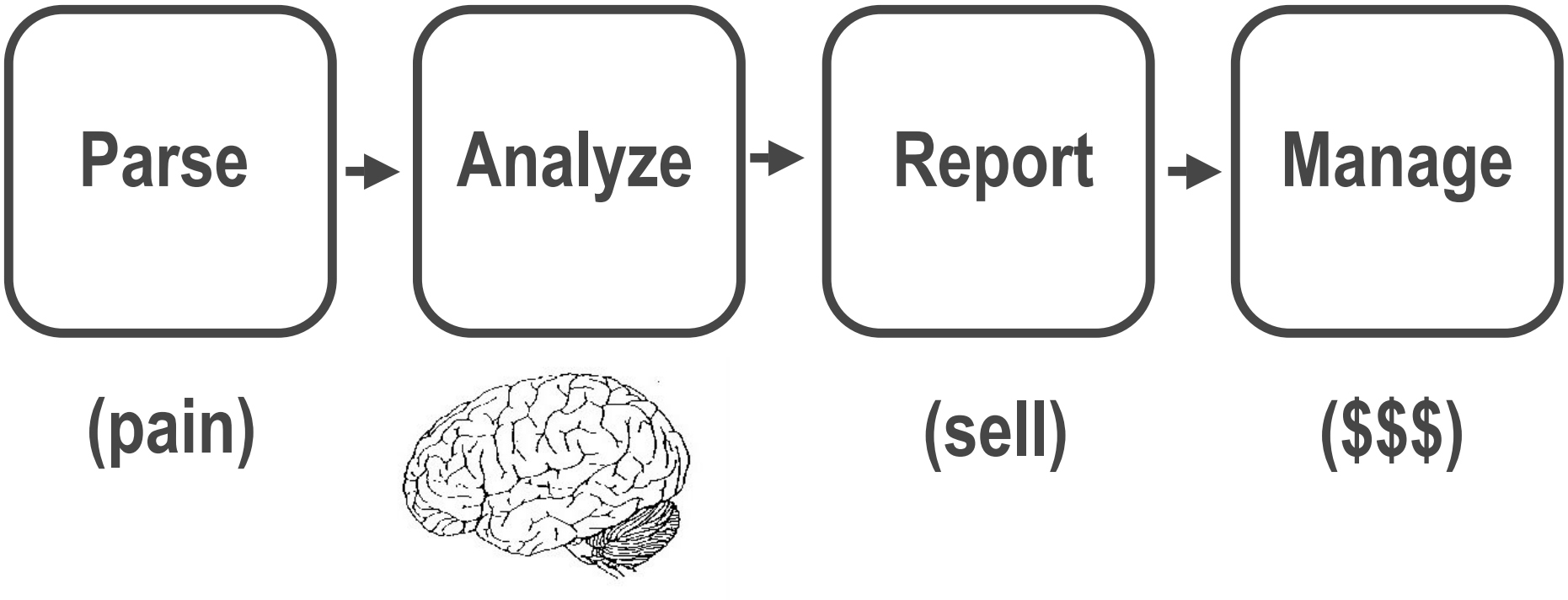
A peek inside a static analysis tool

The industrial perspective



A peek inside a static analysis tool

The industrial perspective



Manage

SSM - Reports - Microsoft Internet Explorer

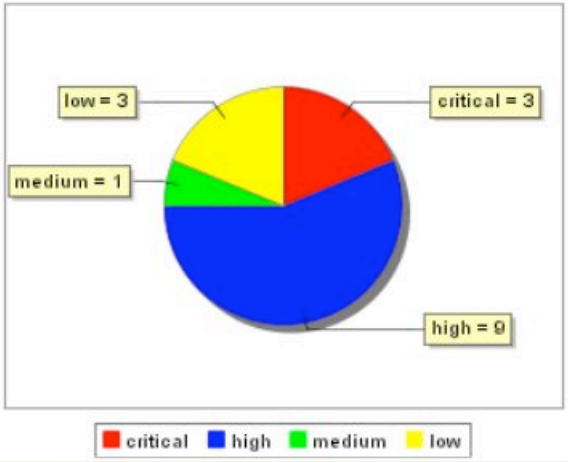
File Edit View Favorites Tools Help

Vulnerability Severity Summary

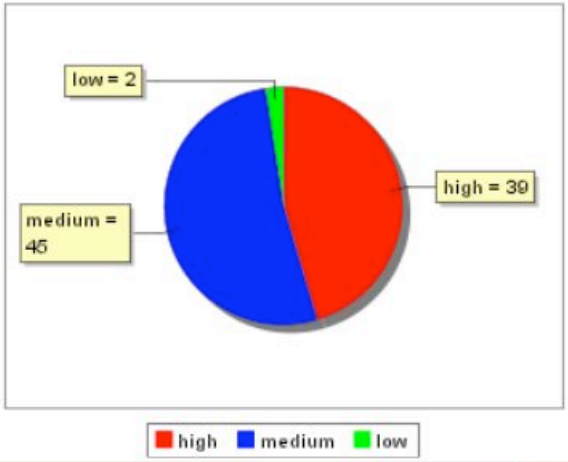
Report Date: February 10, 2005 @ 08:01 AM

Project Name	Most Recent Build Date	Severity	Vulnerabilities	Percent of Total
splc	January 07, 2005 @ 08:21 AM	All Severities	16	100%
		critical	3	18.8%
		high	9	56.2%
		medium	1	6.2%
		low	3	18.8%
zlib	January 03, 2005 @ 10:12 AM	All Severities	86	100%
		high	39	45.3%
		medium	45	52.3%
		low	2	2.3%

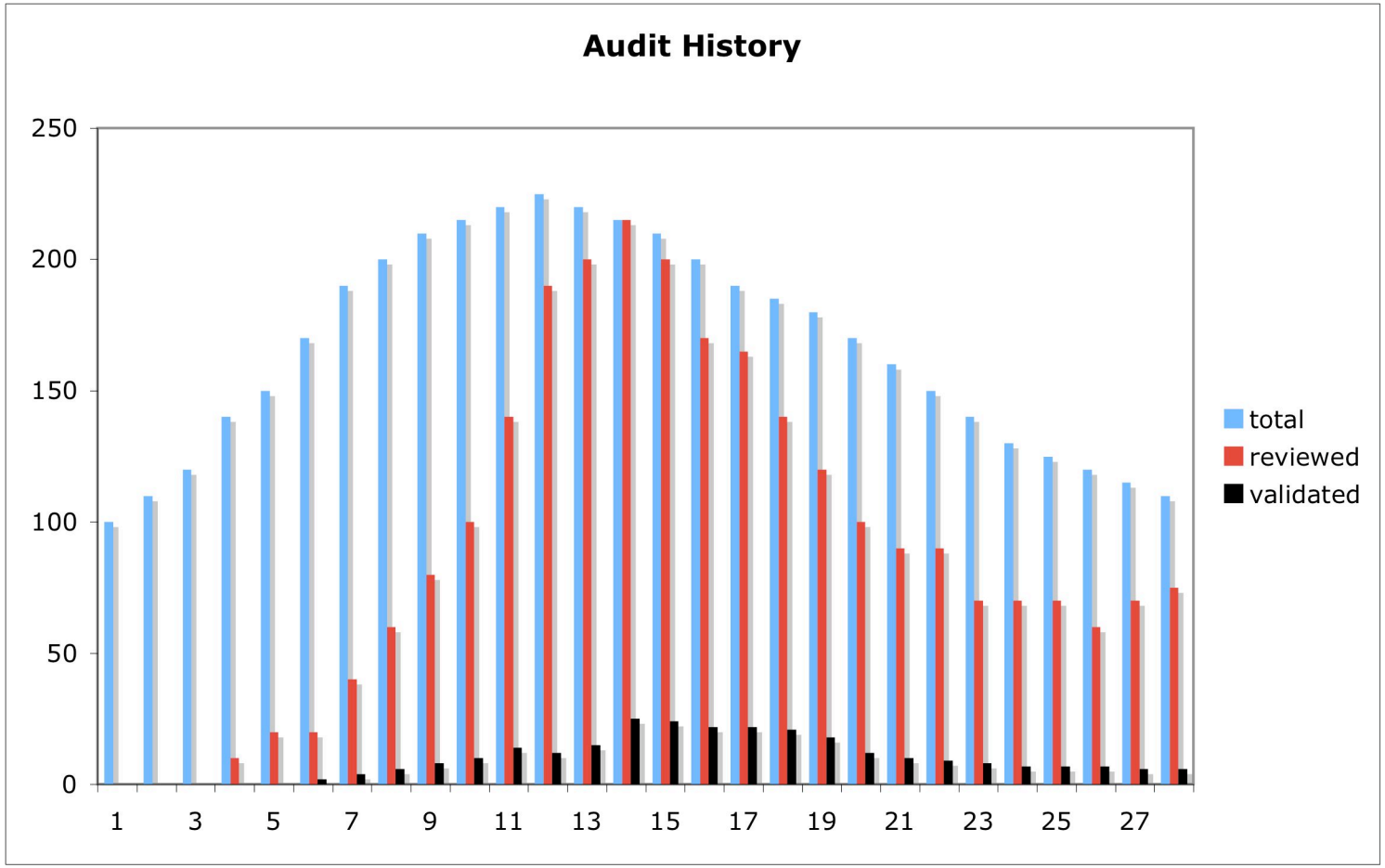
Vulnerability Severities for Project splc



Vulnerability Severities for Project zlib



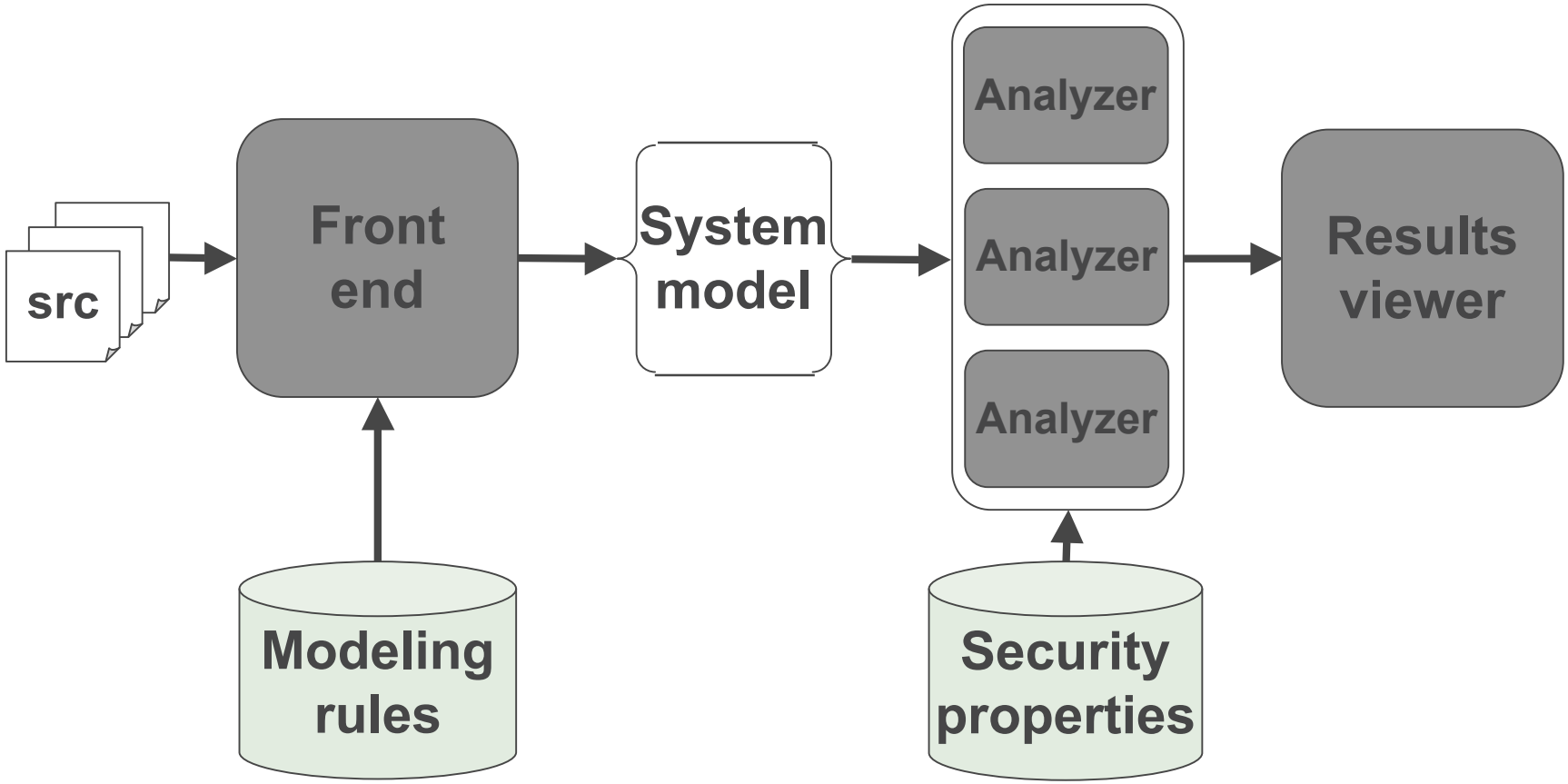
Manage



critical feature: track defects over time

FORTIFY SOFTWARE

A peek inside a static analysis tool



Software security problems

C/C++

- **Buffer overflow**
- **Format string vulnerabilities**
- **Integer overflow**
- **Multi-byte character conversion**
- **Signal handling errors**

Java/C#

- **?**

Java security circa 1996

Mobile code security:

```
public String importantData;
private int[] keysToKingdom;

public int[] getKeysToKingdom() {
  return keysToKingdom;
}
```

Malicious code can alter public member variable.

Getter method leaks mutable array to malicious caller.

Java security circa 2007

Mobile code security:

```
public String importantData;
private int[] keysToKingdom;

public int[] getKeysToKingdom() {
  return keysToKingdom;
}
```

Malicious code can alter public member variable.

Getter method leaks mutable array to malicious caller.

Java security circa 2007

Modern vulnerabilities

- Cross-site scripting
- SQL injection
- Bad error handling
- Poor session management
- Data race conditions

- Not Mobile code
- Not the Java sandbox

What's wrong?

```
Statement stmt;
String q = "select * from users "
           "where uname = '" + uName + "'";
stmt = conn.createStatement();
rs = stmt.executeQuery(q);
```

What's wrong?

```
PreparedStatement stmt;
String q = "select * from users "
           "where uname = '" + uName + "'");
stmt = conn.prepareStatement(q);
ResultSet results = stmt.executeQuery();
```

Injection attacks

- **SQL Injection**
- **Don't stop there!**
  - SQL Injection
  - Command injection
  - File system traversal
  - XML injection
- **Defense**
  - Prepared statements (bind variables)
  - Whitelist good
  - Blacklist bad

#1 cause of security problems:

bad/missing input validation

A blacklist from Apache Tomcat 4.1

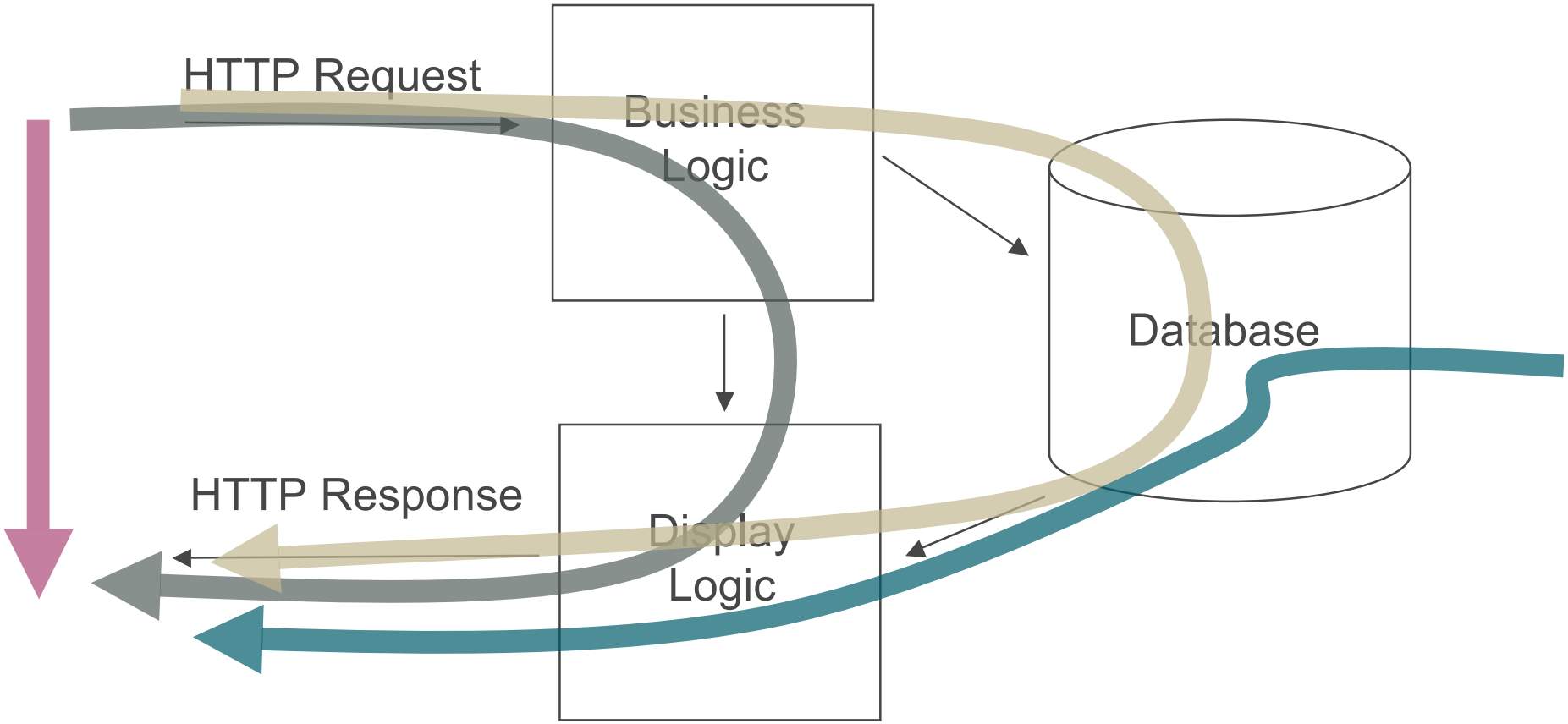
```
for (int i = 0; i < content.length; i++) {
        switch (content[i]) {
            case '<':
                result.append("&lt;");
                break;
            case '>':
                result.append("&gt;");
                break;
           case '&':
               result.append("&amp;");
               break;
           case '"':
               result.append("&quot;");
               break;
           default:
               result.append(content[i]);
        }
```

What about injecting into a CSS stylesheet or into dynamically generated javascript?

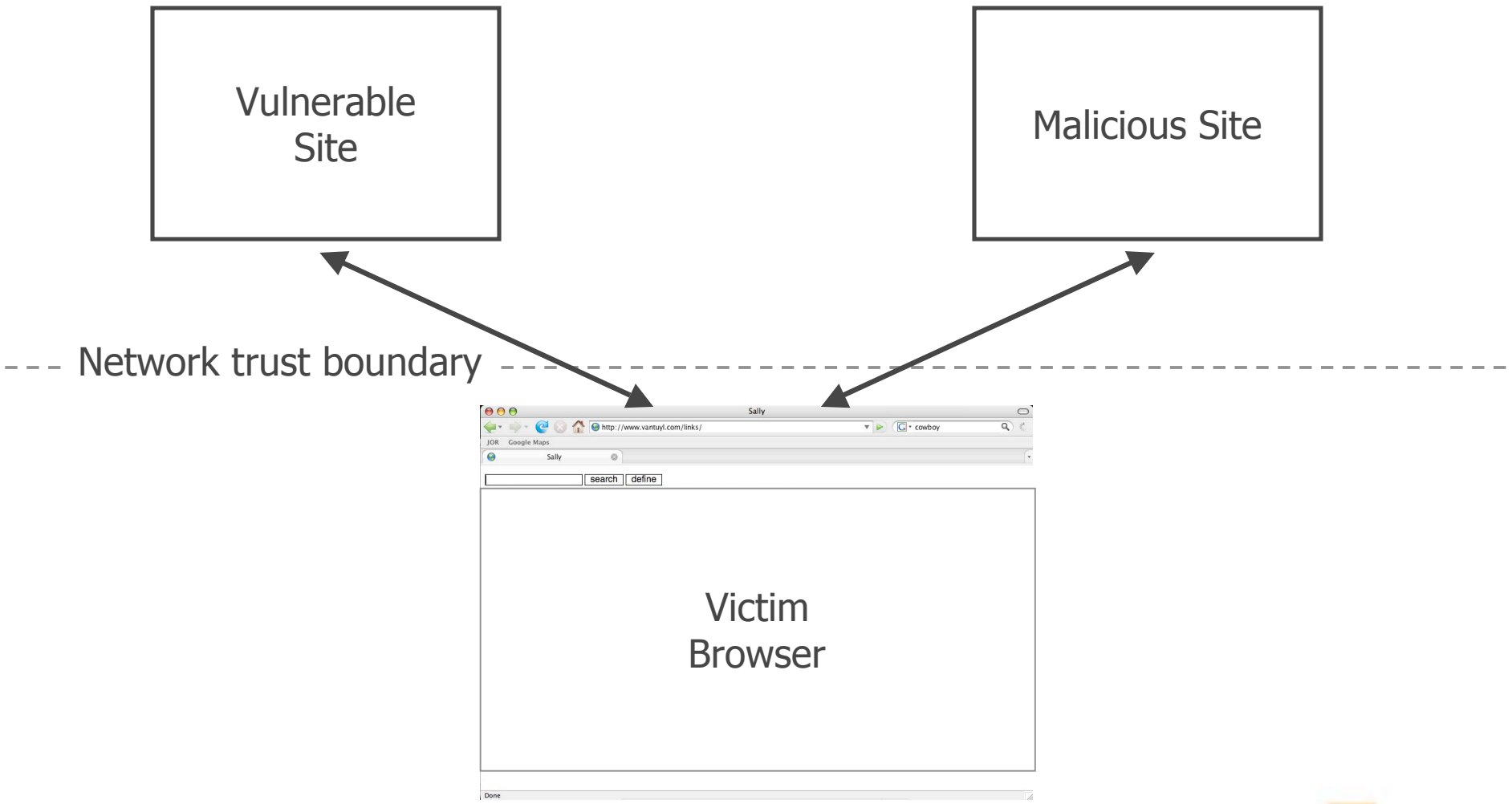
What's wrong?

```
out.println("malformed input: " +
            queryParameter);
```

Cross-site Scripting (XSS)



Cross-site Scripting



More from Tomcat: generating session identifiers

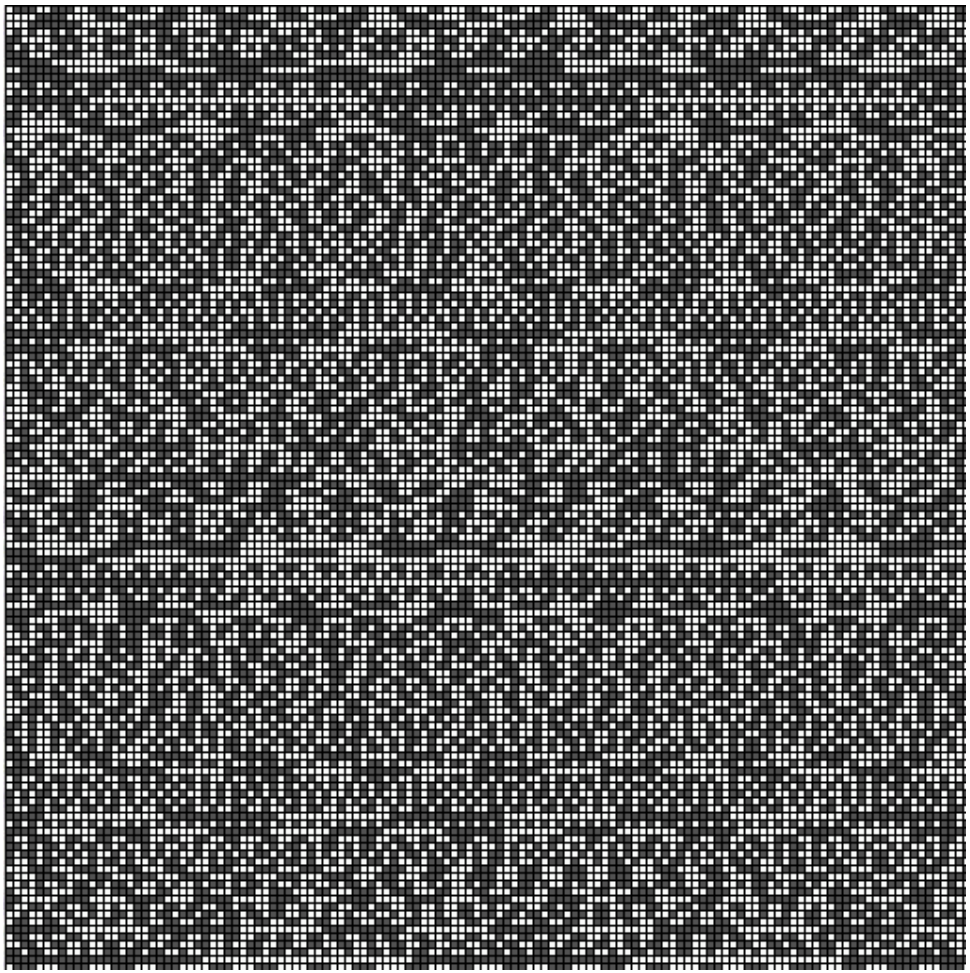
```
protected synchronized Random getRandom() {
 if (this.random == null) {
     try {
       Class clazz = Class.forName(randomClass);
       this.random = (Random) clazz.newInstance();
       long seed = System.currentTimeMillis();
       char entropy[] = getEntropy().toCharArray();
       for (int i = 0; i < entropy.length; i++) {
           long update = ((byte) entropy[i]) << ((i % 8)*8);
           seed ^= update;
       }
       this.random.setSeed(seed);
     } catch (Exception e) {
         this.random = new java.util.Random();
     }
  }
  return (this.random);
}
```

What, me worry?

FORTIFY SOFTWARE

Bad random numbers

Output from java.util.Random



More from Tomcat: generating session identifiers

```
protected synchronized Random getRandom() {
 if (this.random == null) {
     try {
       Class clazz = Class.forName(randomClass);
       this.random = (Random) clazz.newInstance();
       long seed = System.currentTimeMillis();
       char entropy[] = getEntropy().toCharArray();
       for (int i = 0; i < entropy.length; i++) {
           long update = ((byte) entropy[i]) << ((i % 8)*8);
           seed ^= update;
       }
       this.random.setSeed(seed);
     } catch (Exception e) {
         this.random = new java.util.Random();
     }
  }
  return (this.random);
}
```

Bad error handling:

- **insecure**
- **no notification!**

FORTIFY SOFTWARE

What's wrong?

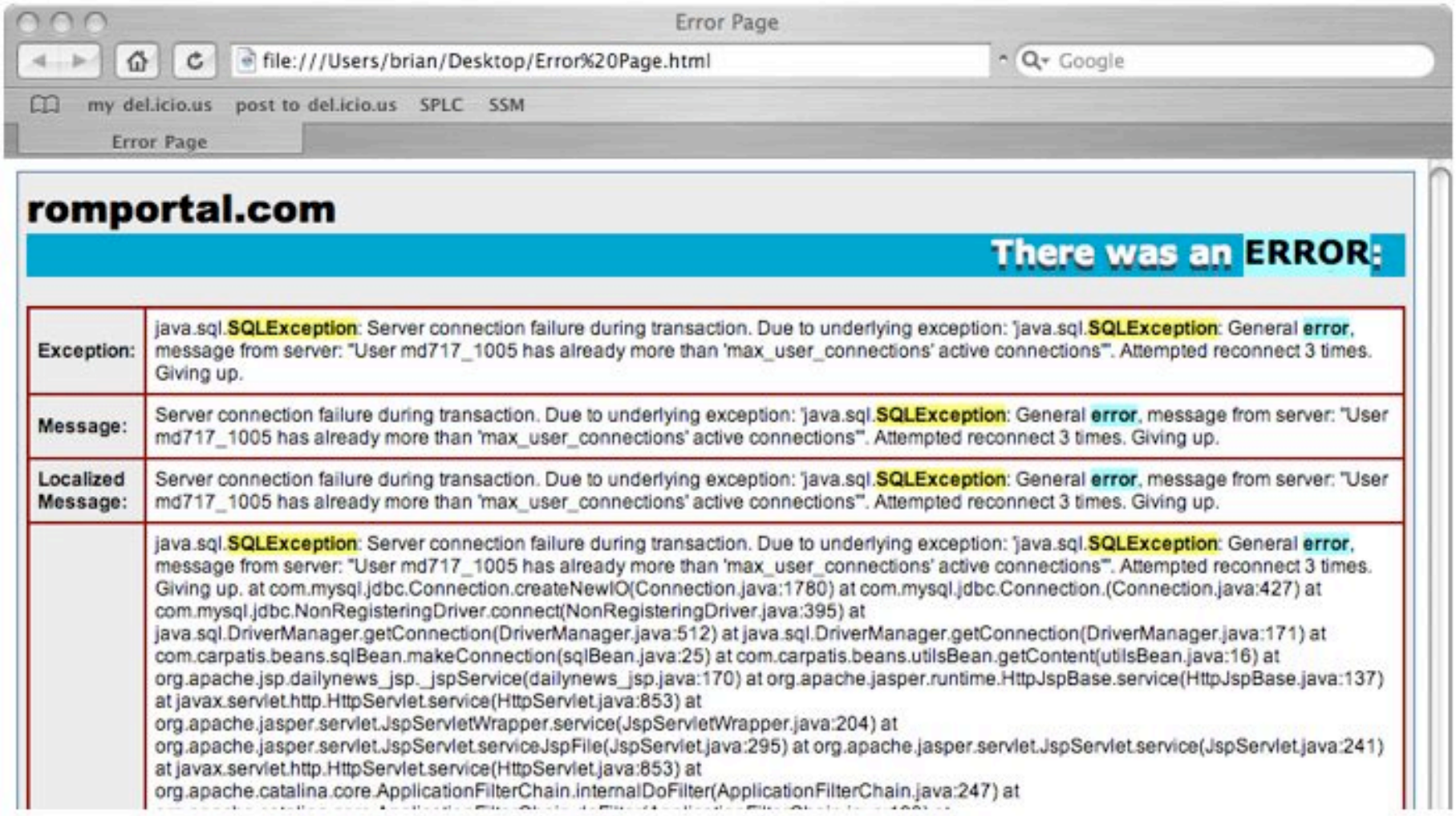
```
private boolean doAuth(String usr, String passwd)
{
  if (checkPasswd(usr, passwd)) {
    session = req.getSession();
    session.setAttribute(USER, usr);
    return true;
  }
}
```

Session Fixation

```
private boolean doAuth(String usr, String passwd)
{
  if (checkPasswd(usr, passwd)) {
    session = req.getSession();
    session.setAttribute(USER, usr);
    return true;
  }
}
```

- Session fixation: attacker forces session ID on user.
- Tomcat (again):
  - Attacker can send link that includes jsessionid

What's wrong?


Error Page
file:///Users/brian/Desktop/Error%20Page.html
Google
my del.icio.us
post to del.icio.us
SPLC
SSM
Error Page
romportal.com
There was an ERROR:
Exception:
java.sql.SQLException: Server connection failure during transaction. Due to underlying exception: 'java.sql.SQLException: General error, message from server: "User md717_1005 has already more than 'max_user_connections' active connections"'. Attempted reconnect 3 times. Giving up.
Message:
Server connection failure during transaction. Due to underlying exception: 'java.sql.SQLException: General error, message from server: "User md717_1005 has already more than 'max_user_connections' active connections"'. Attempted reconnect 3 times. Giving up.
Localized Message:
Server connection failure during transaction. Due to underlying exception: 'java.sql.SQLException: General error, message from server: "User md717_1005 has already more than 'max_user_connections' active connections"'. Attempted reconnect 3 times. Giving up.
java.sql.SQLException: Server connection failure during transaction. Due to underlying exception: 'java.sql.SQLException: General error, message from server: "User md717_1005 has already more than 'max_user_connections' active connections"'. Attempted reconnect 3 times. Giving up. at com.mysql.jdbc.Connection.createNewIO(Connection.java:1780) at com.mysql.jdbc.Connection.(Connection.java:427) at com.mysql.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:395) at java.sql.DriverManager.getConnection(DriverManager.java:512) at java.sql.DriverManager.getConnection(DriverManager.java:171) at com.carpatis.beans.sqlBean.makeConnection(sqlBean.java:25) at com.carpatis.beans.utilsBean.getContent(utilsBean.java:16) at org.apache.jsp.dailynews_jsp._jspService(dailynews_jsp.java:170) at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:137) at javax.servlet.http.HttpServlet.service(HttpServlet.java:853) at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:204) at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:295) at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:241) at javax.servlet.http.HttpServlet.service(HttpServlet.java:853) at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:247) at

Bad Error Handling

- Lack of top-level (global) error handling
- Lack of understanding about how valuable an error message or system information is to an attacker

What's wrong?

```
public class SimpleServlet extends HttpServlet {
    public String acct;
    public Receipt rcpt;
    …
}
```

Concurrency/Synchronization/Race conditions

"Hidden" concurrency errors through singletons. Single object shared between all users:

- HttpServlet
- Struts Action
- Spring Bean

Software security problems

C/C++

- **Buffer overflow**
- **Format string vulnerability**
- **Integer overflow**
- **Multi-byte character conversion**
- **Signal handling errors**

Java/C#

- **SQL injection**
- **Cross-site scripting**
- **Bad error handling**
- **Poor session management**
- **Data race conditions**
- **Buffer overflow**

Native Methods / Unmanaged code

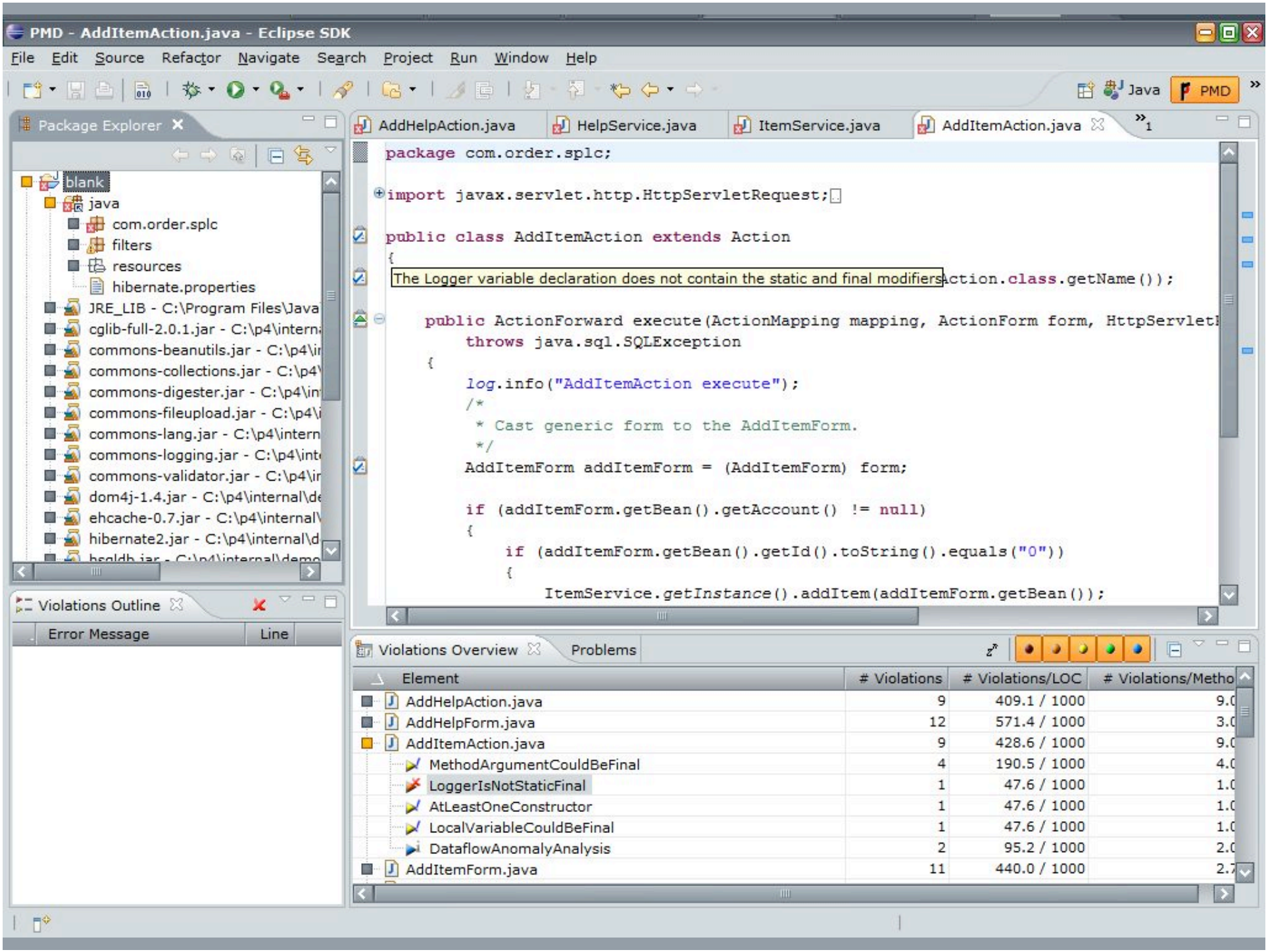
- **All the memory safety promises that Java makes?**
  - Gone
- **All of the type safety promises that Java makes?**
  - Gone
- **Cross-language boundaries == bugs**

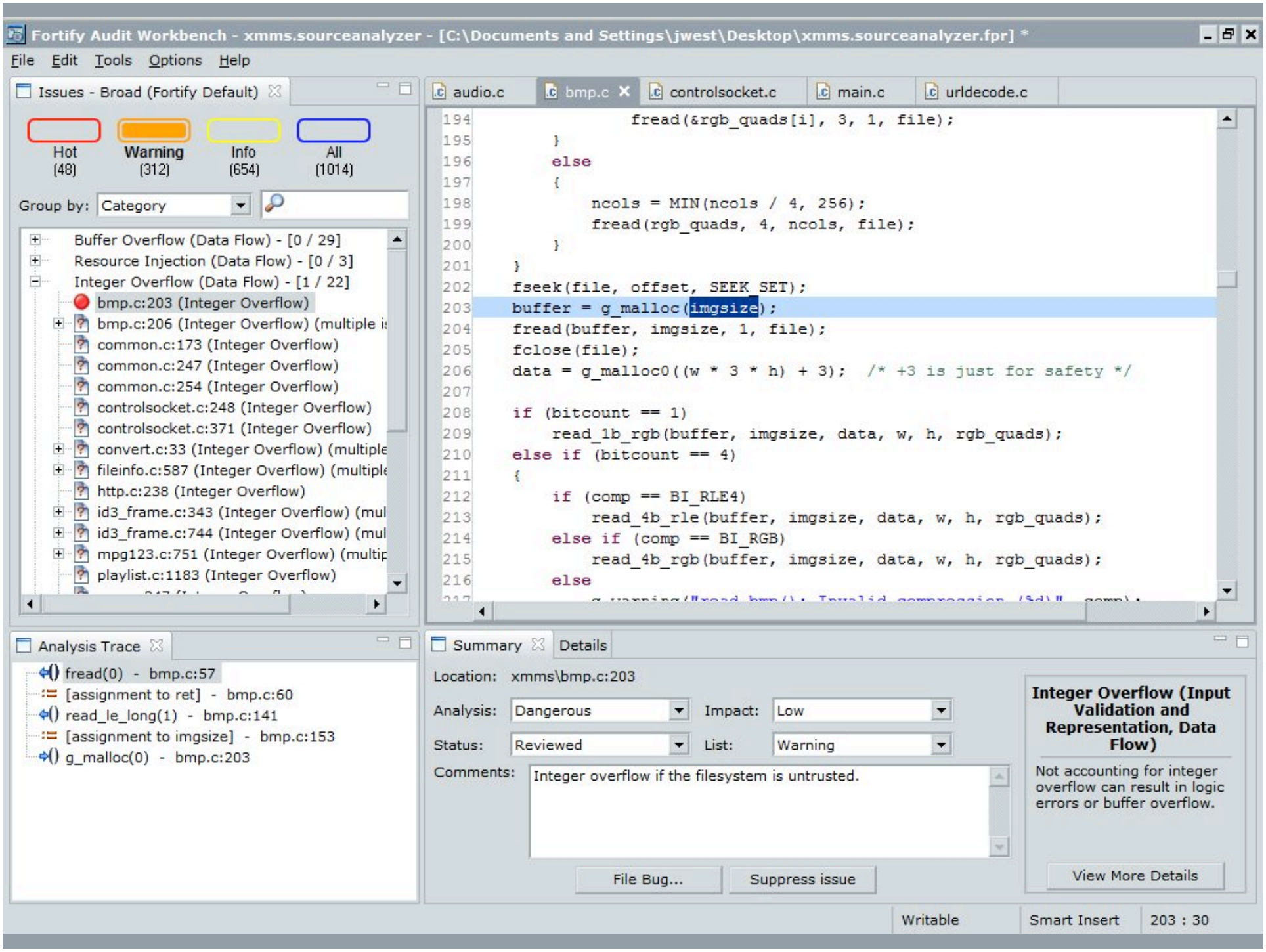
Native methods / Unmanaged code

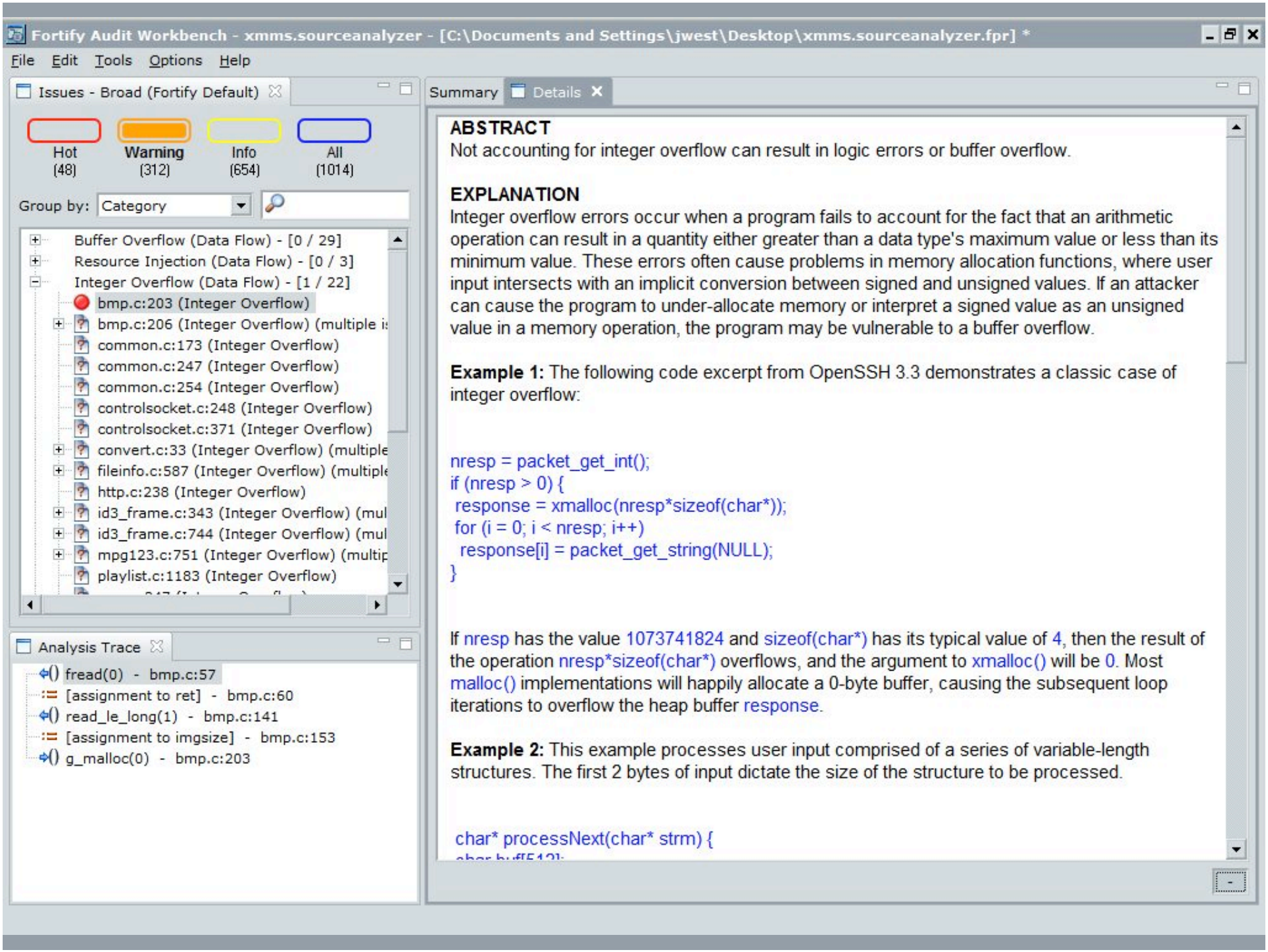
- Date Jan 18, 2007
- Name CVE-2007-0243 (under review)
- Status Candidate
- Description **Buffer overflow in Sun JDK** and Java Runtime Environment (JRE) 5.0 Update 9 and earlier, SDK and JRE 1.4.2_12 and earlier, and SDK and JRE 1.3.1_18 and earlier allows applets to gain privileges via a GIF image with a block with a 0 width field, which triggers memory corruption.

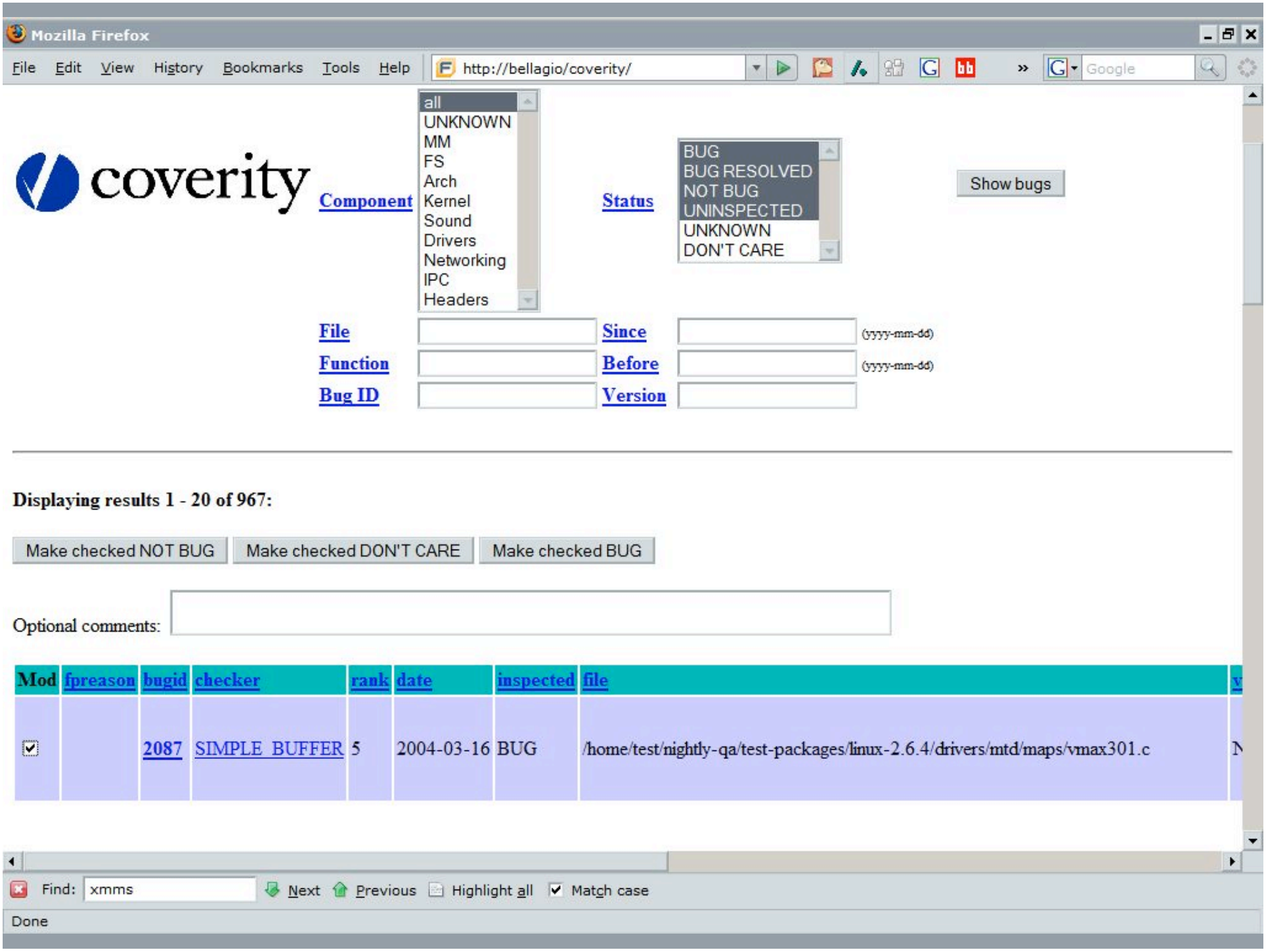
Commercial static analysis keys to success

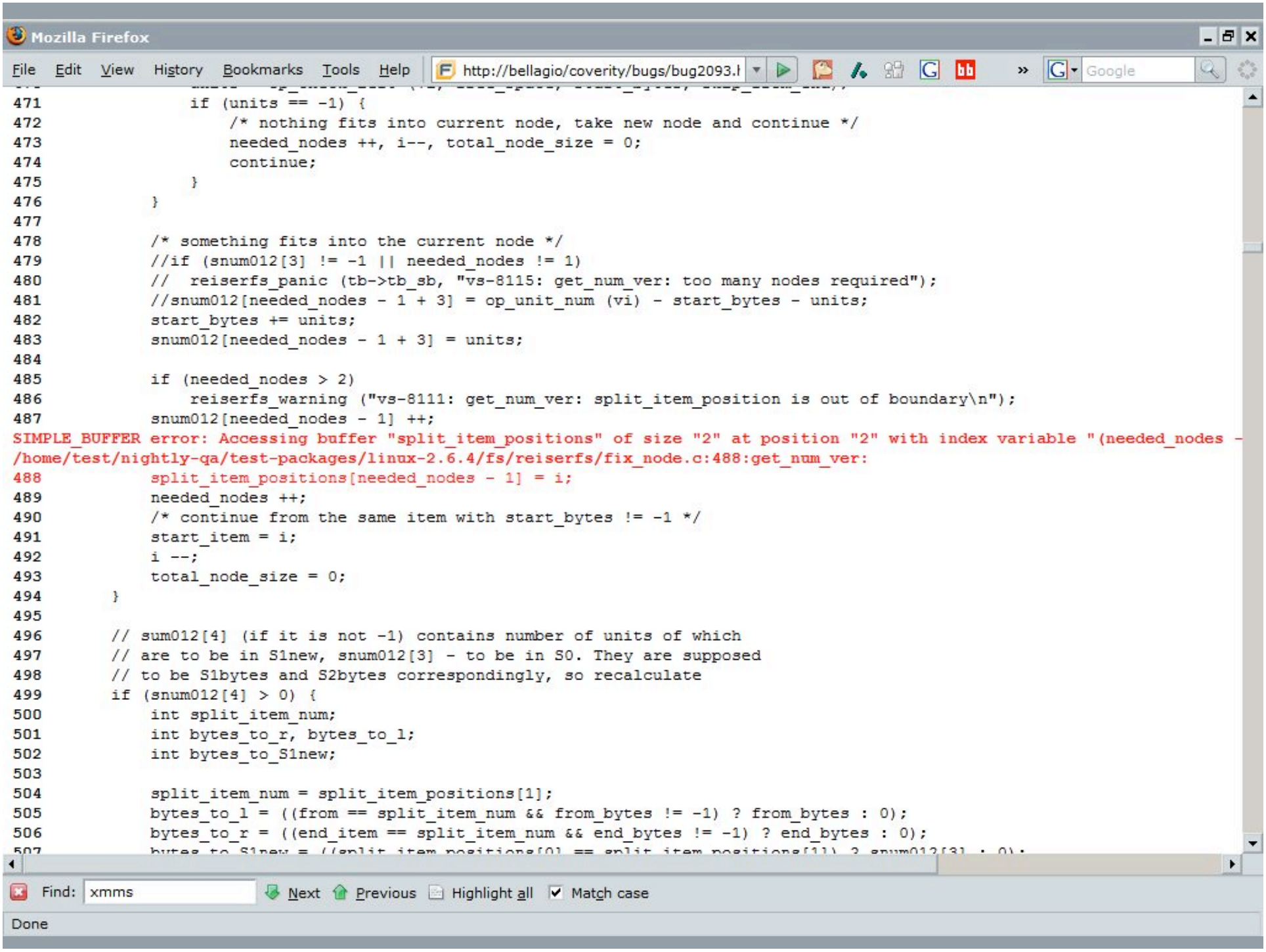
- Don’t crash, don’t hang
- No one cares if your tool is sound
- Run on large code bases (> 1000 kloc → one of everything)
- Explain findings in great detail
- Offer metrics, reporting, management
- A good rule set is crucial
- **Excellent results viewer**

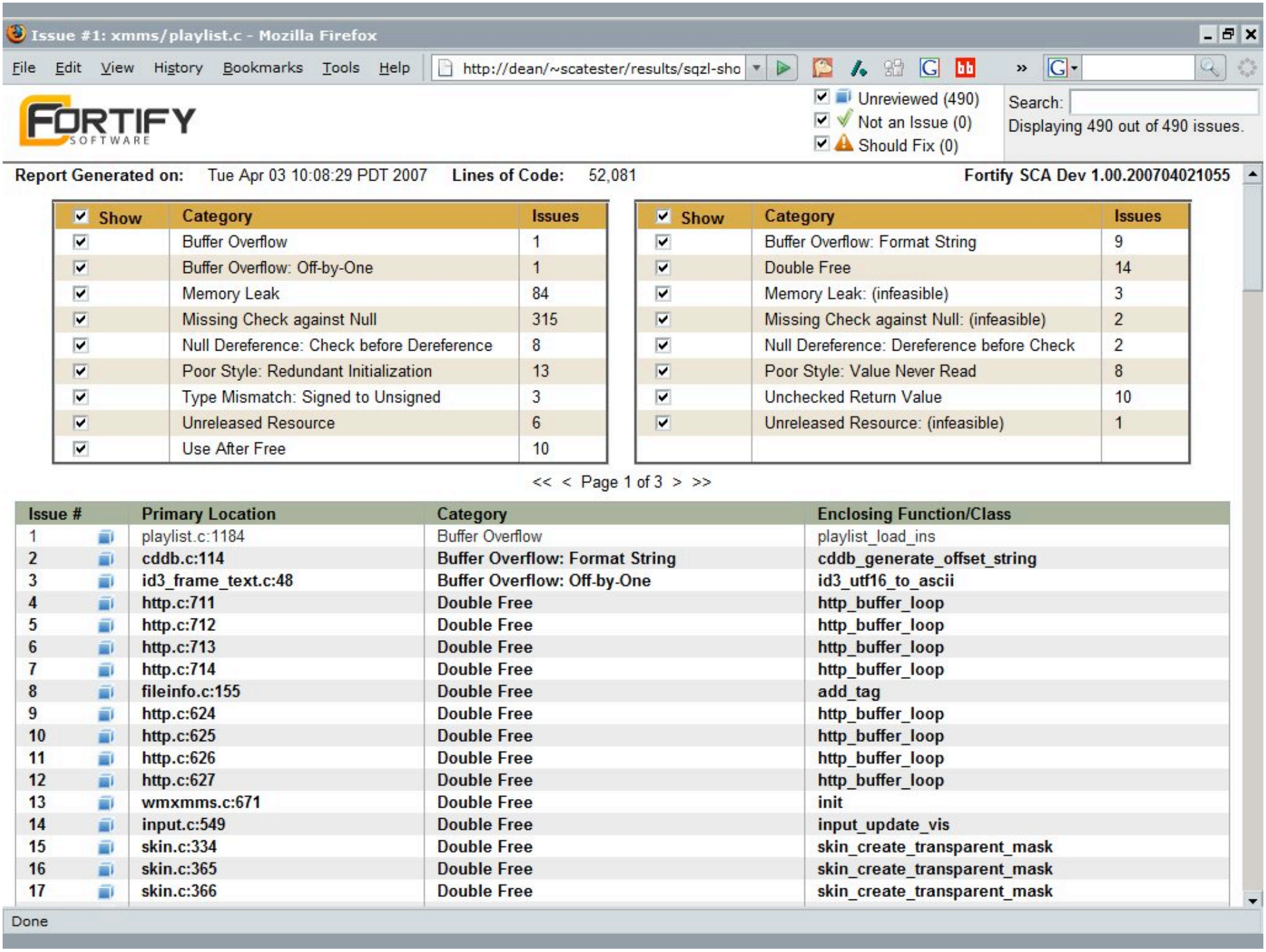

PMD - AddItemAction.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Java
PMD
Package Explorer
blank
java
com.order.splc
filters
resources
hibernate.properties
JRE_LIB - C:\Program Files\Java
cglib-full-2.0.1.jar - C:\p4\intern
commons-beanutils.jar - C:\p4\i
commons-collections.jar - C:\p4
commons-digester.jar - C:\p4\in
commons-fileupload.jar - C:\p4\i
commons-lang.jar - C:\p4\intern
commons-logging.jar - C:\p4\int
commons-validator.jar - C:\p4\i
dom4j-1.4.jar - C:\p4\internal\de
ehcache-0.7.jar - C:\p4\internal
hibernate2.jar - C:\p4\internal\d
AddHelpAction.java
HelpService.java
ItemService.java
AddItemAction.java
package com.order.splc;
import javax.servlet.http.HttpServletRequest;
public class AddItemAction extends Action
{
The Logger variable declaration does not contain the static and final modifiersAction.class.getName());
public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServlet
throws java.sql.SQLException
{
log.info("AddItemAction execute");
/*
* Cast generic form to the AddItemForm.
*/
AddItemForm addItemForm = (AddItemForm) form;
if (addItemForm.getBean().getAccount() != null)
{
if (addItemForm.getBean().getId().toString().equals("0"))
{
ItemService.getInstance().addItem(addItemForm.getBean());
Violations Outline
Error Message
Line
Violations Overview
Problems
Element
Violations
Violations/LOC
Violations/Metho
AddHelpAction.java 9 409.1 / 1000 9.0
AddHelpForm.java 12 571.4 / 1000 3.0
AddItemAction.java 9 428.6 / 1000 9.0
MethodArgumentCouldBeFinal 4 190.5 / 1000 4.0
LoggerIsNotStaticFinal 1 47.6 / 1000 1.0
AtLeastOneConstructor 1 47.6 / 1000 1.0
LocalVariableCouldBeFinal 1 47.6 / 1000 1.0
DataflowAnomalyAnalysis 2 95.2 / 1000 2.0
AddItemForm.java 11 440.0 / 1000 2.7


Fortify Audit Workbench - xmms.sourceanalyzer - [C:\Documents and Settings\jwest\Desktop\xmms.sourceanalyzer.fpr] *
File Edit Tools Options Help
Issues - Broad (Fortify Default)
Hot (48)
Warning (312)
Info (654)
All (1014)
Group by: Category
Buffer Overflow (Data Flow) - [0 / 29]
Resource Injection (Data Flow) - [0 / 3]
Integer Overflow (Data Flow) - [1 / 22]
bmp.c:203 (Integer Overflow)
bmp.c:206 (Integer Overflow) (multiple i
common.c:173 (Integer Overflow)
common.c:247 (Integer Overflow)
common.c:254 (Integer Overflow)
controlsocket.c:248 (Integer Overflow)
controlsocket.c:371 (Integer Overflow)
convert.c:33 (Integer Overflow) (multiple
fileinfo.c:587 (Integer Overflow) (multiple
http.c:238 (Integer Overflow)
id3_frame.c:343 (Integer Overflow) (mul
id3_frame.c:744 (Integer Overflow) (mul
mpg123.c:751 (Integer Overflow) (multip
playlist.c:1183 (Integer Overflow)
audio.c
bmp.c
controlsocket.c
main.c
urldecode.c
fread(&rgb_quads[i], 3, 1, file);
}
else
{
ncols = MIN(ncols / 4, 256);
fread(rgb_quads, 4, ncols, file);
}
}
fseek(file, offset, SEEK_SET);
buffer = g_malloc(imgsize);
fread(buffer, imgsize, 1, file);
fclose(file);
data = g_malloc0((w * 3 * h) + 3); /* +3 is just for safety */
if (bitcount == 1)
read_1b_rgb(buffer, imgsize, data, w, h, rgb_quads);
else if (bitcount == 4)
{
if (comp == BI_RLE4)
read_4b_rle(buffer, imgsize, data, w, h, rgb_quads);
else if (comp == BI_RGB)
read_4b_rgb(buffer, imgsize, data, w, h, rgb_quads);
else
Analysis Trace
fread(0) - bmp.c:57
[assignment to ret] - bmp.c:60
read_le_long(1) - bmp.c:141
[assignment to imgsize] - bmp.c:153
g_malloc(0) - bmp.c:203
Summary
Details
Location: xmms\bmp.c:203
Analysis: Dangerous
Impact: Low
Status: Reviewed
List: Warning
Comments: Integer overflow if the filesystem is untrusted.
File Bug...
Suppress issue
Integer Overflow (Input Validation and Representation, Data Flow)
Not accounting for integer overflow can result in logic errors or buffer overflow.
View More Details
Writable
Smart Insert
203 : 30


Fortify Audit Workbench - xmms.sourceanalyzer - [C:\Documents and Settings\jwest\Desktop\xmms.sourceanalyzer.fpr] *
File Edit Tools Options Help
Issues - Broad (Fortify Default)
Hot (48)
Warning (312)
Info (654)
All (1014)
Group by: Category
Buffer Overflow (Data Flow) - [0 / 29]
Resource Injection (Data Flow) - [0 / 3]
Integer Overflow (Data Flow) - [1 / 22]
bmp.c:203 (Integer Overflow)
bmp.c:206 (Integer Overflow) (multiple i
common.c:173 (Integer Overflow)
common.c:247 (Integer Overflow)
common.c:254 (Integer Overflow)
controlsocket.c:248 (Integer Overflow)
controlsocket.c:371 (Integer Overflow)
convert.c:33 (Integer Overflow) (multiple
fileinfo.c:587 (Integer Overflow) (multiple
http.c:238 (Integer Overflow)
id3_frame.c:343 (Integer Overflow) (mul
id3_frame.c:744 (Integer Overflow) (mul
mpg123.c:751 (Integer Overflow) (multip
playlist.c:1183 (Integer Overflow)
Analysis Trace
fread(0) - bmp.c:57
[assignment to ret] - bmp.c:60
read_le_long(1) - bmp.c:141
[assignment to imgsize] - bmp.c:153
g_malloc(0) - bmp.c:203
Summary
Details
ABSTRACT
Not accounting for integer overflow can result in logic errors or buffer overflow.
EXPLANATION
Integer overflow errors occur when a program fails to account for the fact that an arithmetic operation can result in a quantity either greater than a data type's maximum value or less than its minimum value. These errors often cause problems in memory allocation functions, where user input intersects with an implicit conversion between signed and unsigned values. If an attacker can cause the program to under-allocate memory or interpret a signed value as an unsigned value in a memory operation, the program may be vulnerable to a buffer overflow.
Example 1: The following code excerpt from OpenSSH 3.3 demonstrates a classic case of integer overflow:
nresp = packet_get_int();
if (nresp > 0) {
response = xmalloc(nresp*sizeof(char*));
for (i = 0; i < nresp; i++)
response[i] = packet_get_string(NULL);
}
If nresp has the value 1073741824 and sizeof(char*) has its typical value of 4, then the result of the operation nresp*sizeof(char*) overflows, and the argument to xmalloc() will be 0. Most malloc() implementations will happily allocate a 0-byte buffer, causing the subsequent loop iterations to overflow the heap buffer response.
Example 2: This example processes user input comprised of a series of variable-length structures. The first 2 bytes of input dictate the size of the structure to be processed.
char* processNext(char* strm) {


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://bellagio/coverity/
Google
coverity
Component
all
UNKNOWN
MM
FS
Arch
Kernel
Sound
Drivers
Networking
IPC
Headers
Status
BUG
BUG RESOLVED
NOT BUG
UNINSPECTED
UNKNOWN
DON'T CARE
Show bugs
File
Function
Bug ID
Since
Before
Version
(yyyy-mm-dd)
(yyyy-mm-dd)
Displaying results 1 - 20 of 967:
Make checked NOT BUG
Make checked DON'T CARE
Make checked BUG
Optional comments:
Mod
fpreason
bugid
checker
rank
date
inspected
file
2087
SIMPLE_BUFFER
5
2004-03-16
BUG
/home/test/nightly-qa/test-packages/linux-2.6.4/drivers/mtd/maps/vmax301.c
Find:
xmms
Next
Previous
Highlight all
Match case
Done


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://bellagio/coverity/bugs/bug2093.l
Google
471 if (units == -1) {
472 /* nothing fits into current node, take new node and continue */
473 needed_nodes ++, i--, total_node_size = 0;
474 continue;
475 }
476 }
477
478 /* something fits into the current node */
479 //if (snum012[3] != -1 || needed_nodes != 1)
480 // reiserfs_panic (tb->tb_sb, "vs-8115: get_num_ver: too many nodes required");
481 //snum012[needed_nodes - 1 + 3] = op_unit_num (vi) - start_bytes - units;
482 start_bytes += units;
483 snum012[needed_nodes - 1 + 3] = units;
484
485 if (needed_nodes > 2)
486 reiserfs_warning ("vs-8111: get_num_ver: split_item_position is out of boundary\n");
487 snum012[needed_nodes - 1] ++;
SIMPLE_BUFFER error: Accessing buffer "split_item_positions" of size "2" at position "2" with index variable "(needed_nodes -
/home/test/nightly-qa/test-packages/linux-2.6.4/fs/reiserfs/fix_node.c:488:get_num_ver:
488 split_item_positions[needed_nodes - 1] = i;
489 needed_nodes ++;
490 /* continue from the same item with start_bytes != -1 */
491 start_item = i;
492 i --;
493 total_node_size = 0;
494 }
495
496 // sum012[4] (if it is not -1) contains number of units of which
497 // are to be in S1new, snum012[3] - to be in S0. They are supposed
498 // to be S1bytes and S2bytes correspondingly, so recalculate
499 if (snum012[4] > 0) {
500 int split_item_num;
501 int bytes_to_r, bytes_to_l;
502 int bytes_to_S1new;
503
504 split_item_num = split_item_positions[1];
505 bytes_to_l = ((from == split_item_num && from_bytes != -1) ? from_bytes : 0);
506 bytes_to_r = ((end_item == split_item_num && end_bytes != -1) ? end_bytes : 0);
Find: xmms
Next
Previous
Highlight all
Match case
Done

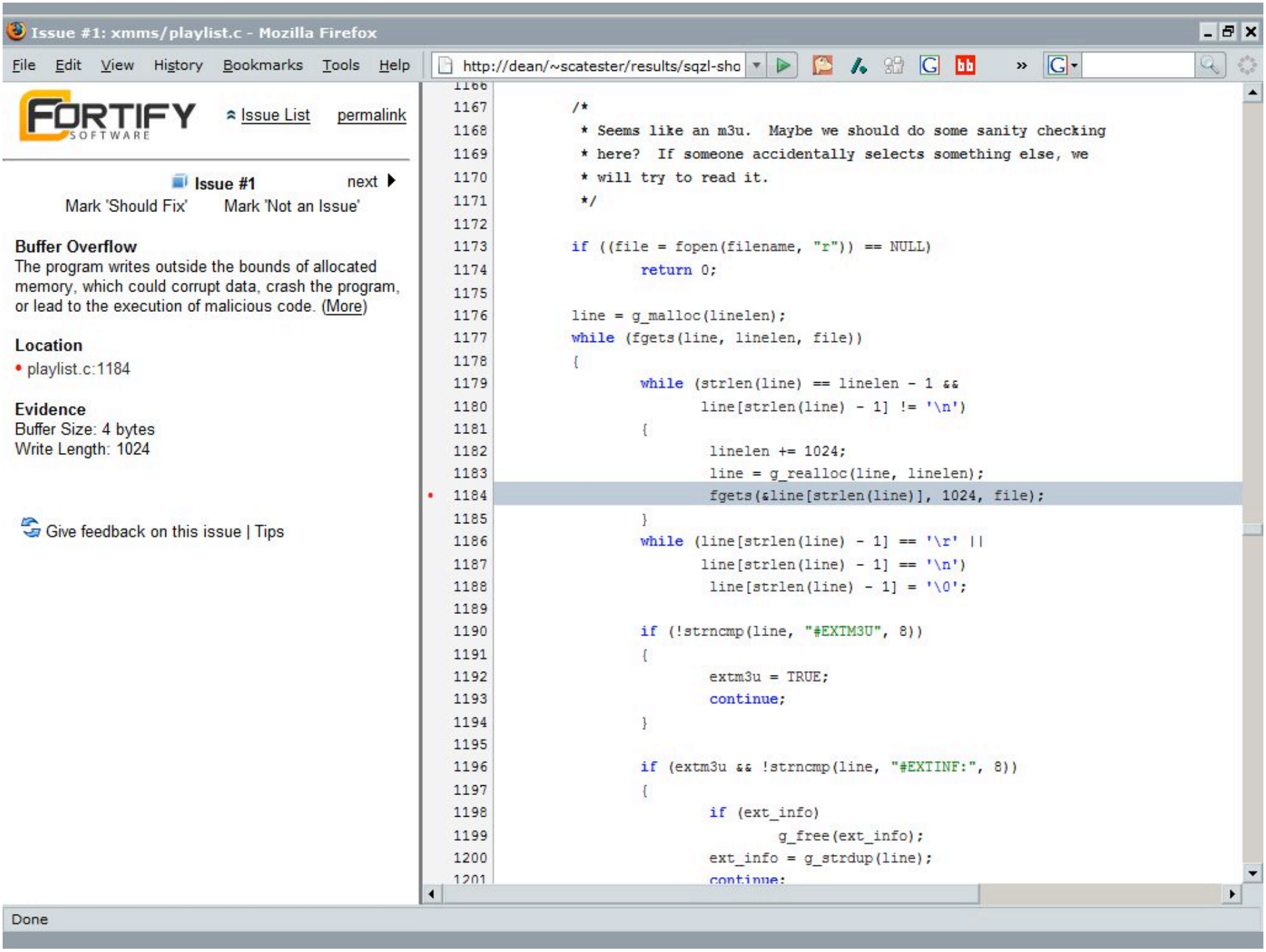


Show	Category	Issues
☑	Buffer Overflow	1
☑	Buffer Overflow: Off-by-One	1
☑	Memory Leak	84
☑	Missing Check against Null	315
☑	Null Dereference: Check before Dereference	8
☑	Poor Style: Redundant Initialization	13
☑	Type Mismatch: Signed to Unsigned	3
☑	Unreleased Resource	6
☑	Use After Free	10

Show	Category	Issues
☑	Buffer Overflow: Format String	9
☑	Double Free	14
☑	Memory Leak: (infeasible)	3
☑	Missing Check against Null: (infeasible)	2
☑	Null Dereference: Dereference before Check	2
☑	Poor Style: Value Never Read	8
☑	Unchecked Return Value	10
☑	Unreleased Resource: (infeasible)	1

<< < Page 1 of 3 > >>

Issue #	Primary Location	Category	Enclosing Function/Class
1	playlist.c:1184	Buffer Overflow	playlist_load_ins
2	**cddb.c:114**	**Buffer Overflow: Format String**	**cddb_generate_offset_string**
3	**id3_frame_text.c:48**	**Buffer Overflow: Off-by-One**	**id3_utf16_to_ascii**
4	**http.c:711**	**Double Free**	**http_buffer_loop**
5	**http.c:712**	**Double Free**	**http_buffer_loop**
6	**http.c:713**	**Double Free**	**http_buffer_loop**
7	**http.c:714**	**Double Free**	**http_buffer_loop**
8	**fileinfo.c:155**	**Double Free**	**add_tag**
9	**http.c:624**	**Double Free**	**http_buffer_loop**
10	**http.c:625**	**Double Free**	**http_buffer_loop**
11	**http.c:626**	**Double Free**	**http_buffer_loop**
12	**http.c:627**	**Double Free**	**http_buffer_loop**
13	**wmxmms.c:671**	**Double Free**	**init**
14	**input.c:549**	**Double Free**	**input_update_vis**
15	**skin.c:334**	**Double Free**	**skin_create_transparent_mask**
16	**skin.c:365**	**Double Free**	**skin_create_transparent_mask**
17	**skin.c:366**	**Double Free**	**skin_create_transparent_mask**


Issue #1: xmms/playlist.c - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://dean/~scatester/results/sqzl-sho
FORTIFY SOFTWARE
Issue List
permalink
Issue #1
next
Mark 'Should Fix'
Mark 'Not an Issue'
Buffer Overflow
The program writes outside the bounds of allocated memory, which could corrupt data, crash the program, or lead to the execution of malicious code. (More)
Location
• playlist.c:1184
Evidence
Buffer Size: 4 bytes
Write Length: 1024
Give feedback on this issue | Tips
1167 /*
1168 * Seems like an m3u. Maybe we should do some sanity checking
1169 * here? If someone accidentally selects something else, we
1170 * will try to read it.
1171 */
1172
1173 if ((file = fopen(filename, "r")) == NULL)
1174 return 0;
1175
1176 line = g_malloc(linelen);
1177 while (fgets(line, linelen, file))
1178 {
1179 while (strlen(line) == linelen - 1 &&
1180 line[strlen(line) - 1] != '\n')
1181 {
1182 linelen += 1024;
1183 line = g_realloc(line, linelen);
1184 fgets(&line[strlen(line)], 1024, file);
1185 }
1186 while (line[strlen(line) - 1] == '\r' ||
1187 line[strlen(line) - 1] == '\n')
1188 line[strlen(line) - 1] = '\0';
1189
1190 if (!strncmp(line, "#EXTM3U", 8))
1191 {
1192 extm3u = TRUE;
1193 continue;
1194 }
1195
1196 if (extm3u && !strncmp(line, "#EXTINF:", 8))
1197 {
1198 if (ext_info)
1199 g_free(ext_info);
1200 ext_info = g_strdup(line);
1201 continue;
Done

Analyzing source vs. analyzing executable

- **Why not analyze the exe?**
  - Everybody has it.
  - No need to guess at what the compiler will do.
  - No need for rules about how functions behave.
- **but …**
  - Decompilation is difficult in some cases.
  - Loss of context hurts.

Analyzing source vs. analyzing executable

Analyzing the binary:

- Lose ability to detect errors related to interface semantics
- SQL Injection:

```
ctx.getAuthUserName(&userName);
CString query;
query = "SELECT * FROM items WHERE owner = '"
      + userName + "' AND itemname = '"
      + request.Lookup("item") + "'";
dbms.ExecuteSQL(query);
```

Analyzing source vs. analyzing executable

Bytecode to the rescue? Nope.

- Cross-site scripting in JSP (Java server pages):

```
<fmt:message key="hello">
  <fmt:param value="${param.test}"/>
</fmt:message>
```

Analyzing source vs. analyzing executable

JSP translation into Java (100 lines of boilderplate omitted)

```
//^%$__TAG_CODEGEN : begin message custom tag block... //[ /WEB-INF/test_fmt.jsp; Line: 5]
    /*** declare AT_BEGIN TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 5]
    if (_fmt_message0 == null) _fmt_message0 = new org.apache.taglibs.standard.tag.el.fmt.MessageTag(); //[ /WEB-INF/test_fmt.jsp; Line: 5]
    _fmt_message0.setPageContext(pageContext); //[ /WEB-INF/test_fmt.jsp; Line: 5]
    _fmt_message0.setParent((javax.servlet.jsp.tagext.Tag)null); //[ /WEB-INF/test_fmt.jsp; Line: 5]
    _activeTag = _fmt_message0; //[ /WEB-INF/test_fmt.jsp; Line: 5]
    _fmt_message0.setKey(weblogic.utils.StringUtils.valueOf("hello")); //[ /WEB-INF/test_fmt.jsp; Line: 5]
    _int0 = _fmt_message0.doStartTag(); //[ /WEB-INF/test_fmt.jsp; Line: 5]
    /*** sync AT_BEGIN TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 5]
    if (_int0 != Tag.SKIP_BODY) { // begin !SKIP_BODY... //[ /WEB-INF/test_fmt.jsp; Line: 5]
      if (_int0 == BodyTag.EVAL_BODY_BUFFERED) { //[ /WEB-INF/test_fmt.jsp; Line: 5]
        out = pageContext.pushBody(); //[ /WEB-INF/test_fmt.jsp; Line: 5]
        _fmt_message0.setBodyContent((BodyContent)out); //[ /WEB-INF/test_fmt.jsp; Line: 5]
        _fmt_message0.doInitBody(); //[ /WEB-INF/test_fmt.jsp; Line: 5]
      } //[ /WEB-INF/test_fmt.jsp; Line: 5]
      do { //[ /WEB-INF/test_fmt.jsp; Line: 5]
        /*** sync AT_BEGIN Vars after doInitBody ***/ //[ /WEB-INF/test_fmt.jsp; Line: 5]
        /*** declare & sync NESTED TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 5]
        out.print("\r\n ");
        //^%$__TAG_CODEGEN : begin param custom tag block... //[ /WEB-INF/test_fmt.jsp; Line: 6]
        /*** declare AT_BEGIN TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 6]
        if (_fmt_param0 == null) _fmt_param0 = new org.apache.taglibs.standard.tag.el.fmt.ParamTag(); //[ /WEB-INF/test_fmt.jsp; Line: 6]
        _fmt_param0.setPageContext(pageContext); //[ /WEB-INF/test_fmt.jsp; Line: 6]
        _fmt_param0.setParent((javax.servlet.jsp.tagext.Tag)_fmt_message0); //[ /WEB-INF/test_fmt.jsp; Line: 6]
        _activeTag = _fmt_param0; //[ /WEB-INF/test_fmt.jsp; Line: 6]
        _fmt_param0.setValue(weblogic.utils.StringUtils.valueOf("${param.test}")); //[ /WEB-INF/test_fmt.jsp; Line: 6]
        _int1 = _fmt_param0.doStartTag(); //[ /WEB-INF/test_fmt.jsp; Line: 6]
        /*** sync AT_BEGIN TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 6]
        weblogic.servlet.jsp.StandardTagLib.fakeEmptyBodyTag(pageContext, _fmt_param0, _int1, true); //[ /WEB-INF/test_fmt.jsp; Line: 6]
        if (_fmt_param0.doEndTag() == Tag.SKIP_PAGE) { _activeTag = null; _releaseTags(_fmt_param0); return; } //[ /WEB-INF/test_fmt.jsp; Line: 6]
        _activeTag = _fmt_param0.getParent(); _fmt_param0.release(); //[ /WEB-INF/test_fmt.jsp; Line: 6]
        //end param custom tag.... //[ /WEB-INF/test_fmt.jsp; Line: 6]
        /*** sync AT_BEGIN TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 6]
        /*** declare & sync AT_END TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 6]
        out.print("\r\n ");
        //^%$__TAG_CODEGEN  //[ /WEB-INF/test_fmt.jsp; Line: 7]
      } while (_fmt_message0.doAfterBody() == IterationTag.EVAL_BODY_AGAIN); //[ /WEB-INF/test_fmt.jsp; Line: 7]
      if (_int0 == BodyTag.EVAL_BODY_BUFFERED) out = pageContext.popBody(); //[ /WEB-INF/test_fmt.jsp; Line: 7]
    } // end !SKIP_BODY //[ /WEB-INF/test_fmt.jsp; Line: 7]
    if (_fmt_message0.doEndTag() == Tag.SKIP_PAGE) { _activeTag = null; _releaseTags(_fmt_message0); return; } //[ /WEB-INF/test_fmt.jsp; Line: 7]
    _activeTag = _fmt_message0.getParent(); _fmt_message0.release(); //[ /WEB-INF/test_fmt.jsp; Line: 7]
    //end message custom tag.... //[ /WEB-INF/test_fmt.jsp; Line: 7]
    /*** sync AT_BEGIN TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 7]
    /*** declare & sync AT_END TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 7]
    _writeText(response, out, _wl_block2, _wl_block2Bytes);
    //^%$__TAG_CODEGEN : begin message custom tag block... //[ /WEB-INF/test_fmt.jsp; Line: 9]
    /*** declare AT_BEGIN TagExtra Vars here ***/ //[ /WEB-INF/test_fmt.jsp; Line: 9]
    if (_fmt_message0 == null) _fmt_message0 = new org.apache.taglibs.standard.tag.el.fmt.MessageTag();
```

FORTIFY SOFTWARE

Selling Security Software

FORTIFY SOFTWARE

Why buy?

1) protect the business from “bad guys”

2) limit liability, comply with legislation

3) avoid damage to brand and reputation

FORTIFY SOFTWARE

Customers

The quick fix

Secure Development Lifecycle (SDL)

- Security Training
- Risk Analysis
- Abuse Cases
- **Source Code Analysis & Review**
- Security Testing

Programming is hard.

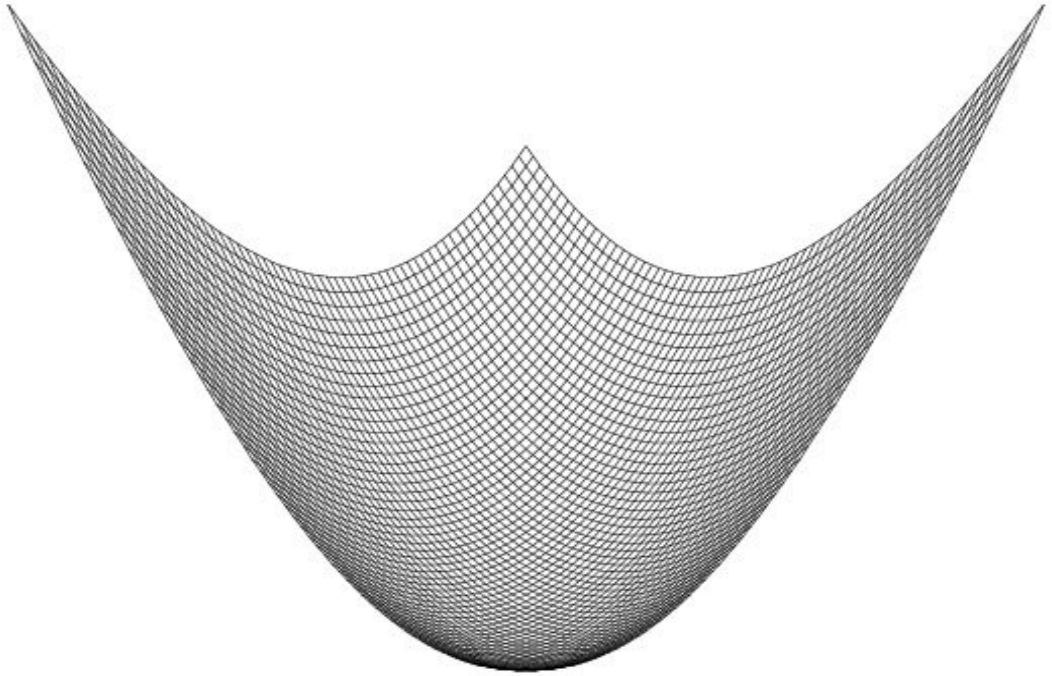
Donald Knuth

The security team

- Motivation: make code review more efficient
- Simple != unimportant

```
conn = DriverManager.getConnection
          (connStr, "scott", "tiger");
```

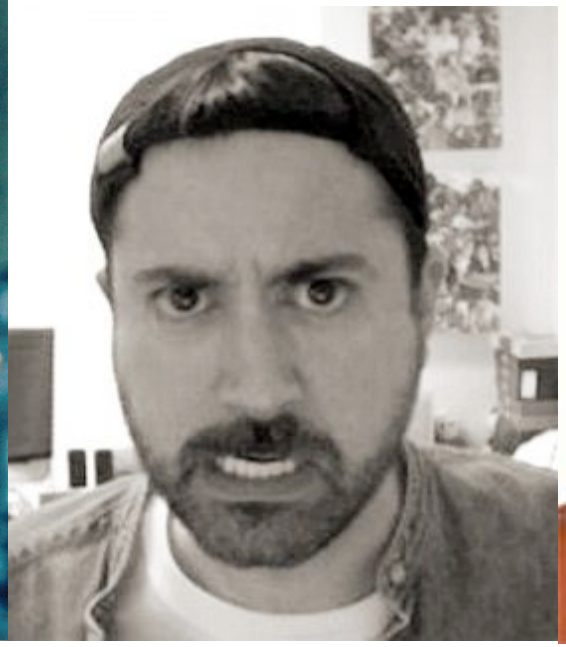
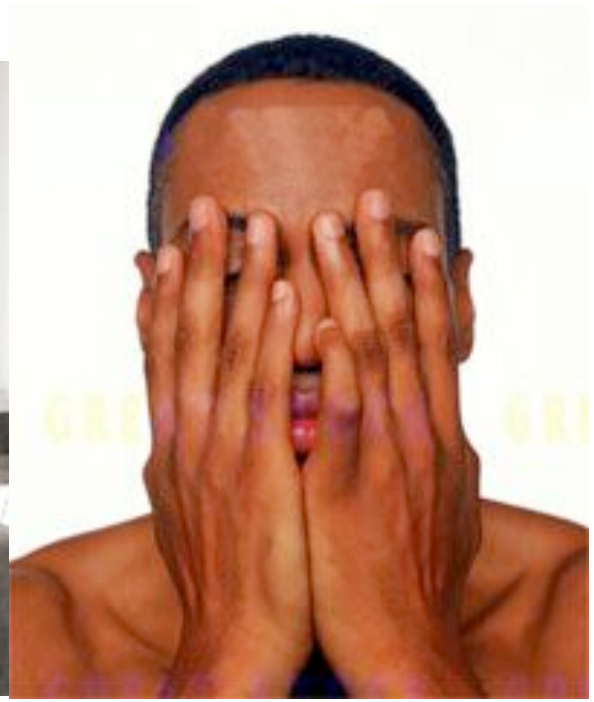
Developers are optimizers



MINIMUM OF 0.0 AT X = -0.0009 AND Y = -0.0009

FORTIFY SOFTWARE

Developers

Developers say the darndest things

“That’s not a vulnerability because …”

Developers say the darndest things

"That's not a vulnerability because ...

I trust the system administrators."

Developers say the darndest things

"That's not a vulnerability because ...

you have to authenticate before you can post to that URL."

Developers say the darndest things

"That's not a vulnerability because ...

no one would ever think to do that!"

FORTIFY SOFTWARE

Developers say the darndest things

"That's not a vulnerability because ...

that method call can never fail!"

FORTIFY SOFTWARE

Developers say the darndest things

"That's not a vulnerability because ...

making it secure is someone else's job."

FORTIFY SOFTWARE

Developers say the darndest things

"That's not a vulnerability because ...

that code will never be run."

Developers say the darndest things

"That's not a vulnerability because ...

we already knew about it.

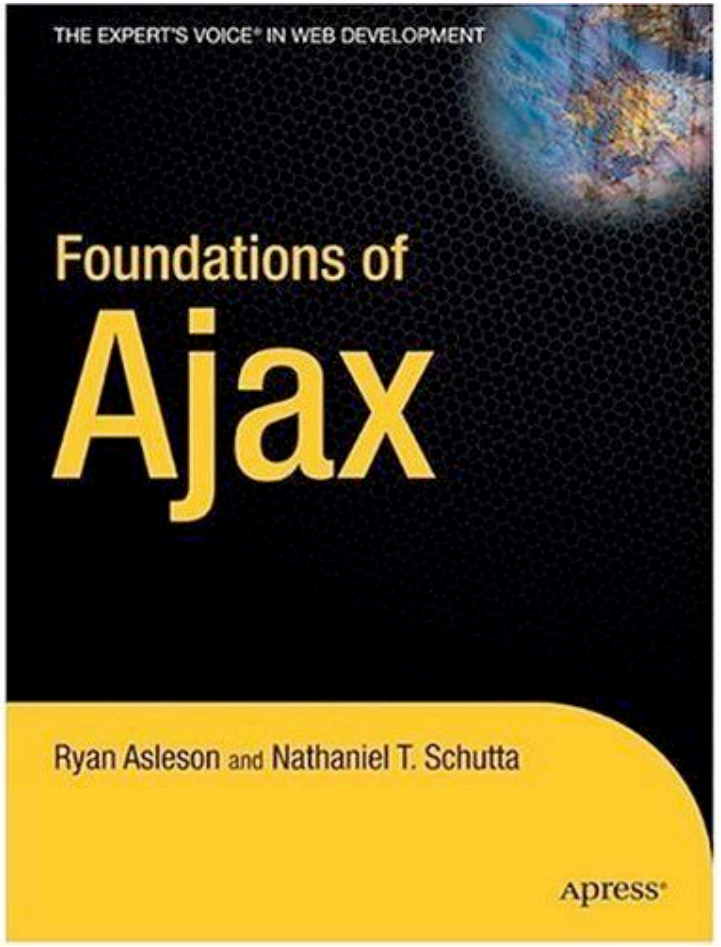
And we were going to fix it.

Someday."

Developers say the darndest things

"We never intended the code that's in there to actually be production-ready code"

Ryan Asleson



FORTIFY SOFTWARE

It needs to be easy

- **Predictable, comprehensible security for non-experts**
  - Minimal static analysis knowledge
  - Minimal security knowledge
- **Solutions**
  - Better languages
  - Better tools

Competition

Competitors

Network-devices

"Security in a Box"

Focusing on the attack over the root cause, "Now fixes applications" is now the latest craze.

- Focused on protecting the infrastructure not the software.
- Some solutions serve as a stop gap, but by no means replace the need to build security in.

Penetration Testing

"Badness-ometers"

A popular method for establishing awareness offered up as a sustainable solution.

- Great for demonstrating the problem.
- Testing without upstream activities to "test" is pointless and expensive.

An extra scoop of quality

"Security Light"

To broaden reach of niche products, static analysis vendors add security to a list of quality issues.

- Security issues are not "just another bug".

FORTIFY SOFTWARE

Security vs. Quality

- **Quality**
  - Bugs are cheap.
  - Be picky about what you report (low false positives).
- **Security**
  - Missing a bug is expensive.
  - Don’t throw results away (low false negatives).

Sound off: false positives vs. false negatives

- Instant gratification
- Tuning/triage
- Customization

FORTIFY SOFTWARE

Hard Problems, Real and Imagined

The Usual Suspects

- Pointer aliasing
- Loop invariants
- Precision vs. scalability tradeoffs
- Making use of idioms and programmer hints

A Few More Hard Ones

- Knowing what to check for
- Low false positives and low false negatives not enough
- Getting users to customize
- Usability, documentation, support
- How much should it cost?

end

brian@fortifysoftware.com