Code Analysis Tools

Find your bugs before someone else does!

Thomas Hofer

2010-02-19

Goal

- Easy means of improving your code!
- Which programming languages do you use?

- Webpage...

Goal

- Easy means of improving your code!
- Which programming languages do you use?
- Webpage...

1. Introduction
   - Motivation
   - Code Analysis Tools
2. Recommendations
   - Our criteria
   - Selected tools
3. Further information

Computer Security

- CERN is a prime target
- Can have serious consequences (data loss, damaged image or reputation, loss of confidentiality, material damage...)

- “Computer Security is of highest priority”, CERN Director General, *Annual meeting, January 2010*

Computer Security

- CERN is a prime target
- Can have serious consequences (data loss, damaged image or reputation, loss of confidentiality, material damage...)

- “Computer Security is of highest priority”, CERN Director General, *Annual meeting, January 2010*

Computer Security

- CERN is a prime target
- Can have serious consequences (data loss, damaged image or reputation, loss of confidentiality, material damage...)

- “Computer Security is of highest priority”, CERN Director General, *Annual meeting, January 2010*

When does it apply?

- Creating / Managing
  - Documents
  - Webpages
  - Machines
- Providing services
- **Developing**
  - Software
  - Web applications

When does it apply?

- Creating / Managing
  - Documents
  - Webpages
  - Machines
- Providing services
- **Developing**
  - Software
  - Web applications

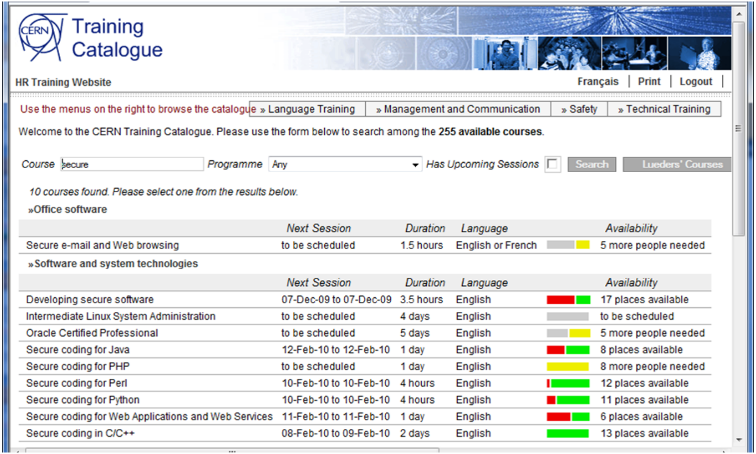
When does it apply?

- Creating / Managing
  - Documents
  - Webpages
  - Machines
- Providing services
- **Developing**
  - Software
  - Web applications

Developing secure software

- Training (before)
- Reviews (right after)
- Vulnerability scanning (*black box*) *(after)*

Developing secure software



Use the menus on the right to browse the catalogue

Welcome to the CERN Training Catalogue. Please use the form below to search among the **255 available courses**.

Course secure *Programme* Any *Has Upcoming Sessions* □

10 courses found. Please select one from the results below.

»Office software

	Next Session	*Duration*	*Language*	*Availability*
Secure e-mail and Web browsing	to be scheduled	1.5 hours	English or French	▨🟨 5 more people needed

»Software and system technologies

	Next Session	*Duration*	*Language*	*Availability*
Developing secure software	07-Dec-09 to 07-Dec-09	3.5 hours	English	🟥🟩 17 places available
Intermediate Linux System Administration	to be scheduled	4 days	English	▨ to be scheduled
Oracle Certified Professional	to be scheduled	5 days	English	▨🟨 5 more people needed
Secure coding for Java	12-Feb-10 to 12-Feb-10	1 day	English	🟥🟩 8 places available
Secure coding for PHP	to be scheduled	1 day	English	🟨 8 more people needed
Secure coding for Perl	10-Feb-10 to 10-Feb-10	4 hours	English	🟥🟩 12 places available
Secure coding for Python	10-Feb-10 to 10-Feb-10	4 hours	English	🟥🟩 11 places available
Secure coding for Web Applications and Web Services	11-Feb-10 to 11-Feb-10	1 day	English	🟥🟩 6 places available
Secure coding in C/C++	08-Feb-10 to 09-Feb-10	2 days	English	🟩 13 places available

Developing secure software

- Training (before)
- **Static Source Code Analysis** (during and after)
- Reviews (right after)
- Vulnerability scanning (*black box*) *(after)*

Developing secure software

- Training (before)
- **Static Source Code Analysis** (during and after)
- Reviews (right after)
- Vulnerability scanning (*black box*) *(after)*

Developing secure software

- What can YOU do about it...
- ... and still meet your deadlines!

- Static Analysis!
- The earlier a bug is caught, the cheaper it is to fix!

Developing secure software

- What can YOU do about it...
- ... and still meet your deadlines!

- Static Analysis!
- The earlier a bug is caught, the cheaper it is to fix!

Developing secure software

- What can YOU do about it...
- ... and still meet your deadlines!

- Static Analysis!
- The earlier a bug is caught, the cheaper it is to fix!

Code Analysis Tools

Static analyzers can:

- Read your source code but:

  ... do not execute or compile it
- Look for possible flaws and bugs
  - Security
  - Reliability
  - Functionality

What CAN they do?

Code Analysis Tools will

- Look for known vulnerabilities and common mistakes
- Report hits
- Possibly suggest fixes

- Help *finding* bugs...
- They find all sorts of bugs, not only security issues!

What CAN they do?

Code Analysis Tools will

- Look for known vulnerabilities and common mistakes
- Report hits
- Possibly suggest fixes

- Help *finding* bugs...
- They find all sorts of bugs, not only security issues!

What CAN they NOT do?

Code Analysis Tools will not

- Automagically fix bugs
- Find ALL bugs (*i.e.* false negatives)
- Find ONLY bugs (*i.e.* false positives)

Requirements

- Quick results
- Few false alarms
- Ease of use

- At least some results

Requirements

- Quick results
- Few false alarms
- Ease of use
- At least some results

Requirements

- Quick results
- Few false alarms
- Ease of use

- At least some results

Requirements

- Quick results
- Few false alarms
- Ease of use

- At least some results

Overview of selected tools

- C / C++
  - Flawfinder
  - RATS
  - Coverity
- Java
  - FindBugs
  - CodePro Analyser
- PHP
  - Pixy
  - RATS
- Perl
  - Perl::Critic
  - RATS
  - Lionel Cons' lint
- Python
  - RATS
  - pychecker
  - pylint

Flawfinder

- C / C++
- Freeware / Unix
- Commonly misused library calls

- Demo

  `http://cern.ch/security/codetools/c_tools.html#flawfinder`

Flawfinder

- C / C++
- Freeware / Unix
- Commonly misused library calls

- Demo

  http://cern.ch/security/codetools/c_tools.html#flawfinder

Flawfinder

- C / C++
- Freeware / Unix
- Commonly misused library calls

- Demo

  http://cern.ch/security/codetools/c_tools.html#flawfinder

Flawfinder

- C / C++
- Freeware / Unix
- Commonly misused library calls

- Demo

  http://cern.ch/security/codetools/c_tools.html#flawfinder

FindBugs

- Java
- Freeware / Eclipse plugin - Standalone application
- Many rules, configurable

http://cern.ch/security/codetools/java_tools.html#FindBugs

FindBugs

- Java
- Freeware / Eclipse plugin - Standalone application
- Many rules, configurable

http://cern.ch/security/codetools/java_tools.html#FindBugs

FindBugs

- Java
- Freeware / Eclipse plugin - Standalone application
- Many rules, configurable

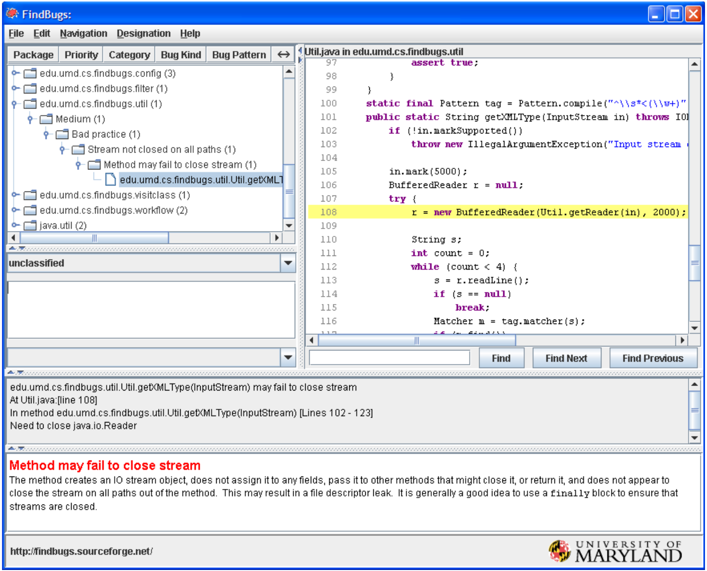
http://cern.ch/security/codetools/java_tools.html#FindBugs

FindBugs

- Java
- Freeware / Eclipse plugin - Standalone application
- Many rules, configurable

`http://cern.ch/security/codetools/java_tools.html#FindBugs`

FindBugs


FindBugs:
File Edit Navigation Designation Help
Package Priority Category Bug Kind Bug Pattern
Util.java in edu.umd.cs.findbugs.util
unclassified
Find
Find Next
Find Previous

- edu.umd.cs.findbugs.config (3)
- edu.umd.cs.findbugs.filter (1)
- edu.umd.cs.findbugs.util (1)
  - Medium (1)
    - Bad practice (1)
      - Stream not closed on all paths (1)
        - Method may fail to close stream (1)
          - edu.umd.cs.findbugs.util.Util.getXMLT
- edu.umd.cs.findbugs.visitclass (1)
- edu.umd.cs.findbugs.workflow (2)
- java.util (2)

```
 97                assert true;
 98            }
 99        }
100        static final Pattern tag = Pattern.compile("^\\s*<(\\w+)"
101        public static String getXMLType(InputStream in) throws IO
102            if (!in.markSupported())
103                throw new IllegalArgumentException("Input stream
104
105            in.mark(5000);
106            BufferedReader r = null;
107            try {
108                r = new BufferedReader(Util.getReader(in), 2000);
109
110                String s;
111                int count = 0;
112                while (count < 4) {
113                    s = r.readLine();
114                    if (s == null)
115                        break;
116                    Matcher m = tag.matcher(s);
```

edu.umd.cs.findbugs.util.Util.getXMLType(InputStream) may fail to close stream
At Util.java:[line 108]
In method edu.umd.cs.findbugs.util.Util.getXMLType(InputStream) [Lines 102 - 123]
Need to close java.io.Reader

Method may fail to close stream

The method creates an IO stream object, does not assign it to any fields, pass it to other methods that might close it, or return it, and does not appear to close the stream on all paths out of the method. This may result in a file descriptor leak. It is generally a good idea to use a `finally` block to ensure that streams are closed.

http://findbugs.sourceforge.net/

UNIVERSITY OF MARYLAND

Perl::Critic

- perl
- Freeware / Unix - Perl Module
- Best Practices: style and security
- Demo

  `http://cern.ch/security/codetools/perl_tools.html#perlcritic`

Perl::Critic

- perl
- Freeware / Unix - Perl Module
- Best Practices: style and security
- Demo

  `http://cern.ch/security/codetools/perl_tools.html#perlcritic`

Perl::Critic

- perl
- Freeware / Unix - Perl Module
- Best Practices: style and security

- Demo

  http://cern.ch/security/codetools/perl_tools.html#perlcritic

Perl::Critic

- perl
- Freeware / Unix - Perl Module
- Best Practices: style and security

- Demo

  http://cern.ch/security/codetools/perl_tools.html#perlcritic

Pixy

- PHP
- Freeware
- XSS & SQLi

- Demo

  http://cern.ch/security/codetools/php_tools.html#Pixy

Pixy

- PHP
- Freeware
- XSS & SQLi

- Demo

  `http://cern.ch/security/codetools/php_tools.html#Pixy`

Pixy

- PHP
- Freeware
- XSS & SQLi

- Demo

  http://cern.ch/security/codetools/php_tools.html#Pixy

Pixy

- PHP
- Freeware
- XSS & SQLi

- Demo

  `http://cern.ch/security/codetools/php_tools.html#Pixy`

RATS

- C / C++ / Perl, and also (more limited) Python / PHP
- Freeware
- Commonly misused library calls

- Demo

  `http://cern.ch/security/codetools/c_tools.html#rats`

RATS

- C / C++ / Perl, and also (more limited) Python / PHP
- Freeware
- Commonly misused library calls
- Demo

  http://cern.ch/security/codetools/c_tools.html#rats

RATS

- C / C++ / Perl, and also (more limited) Python / PHP
- Freeware
- Commonly misused library calls

- Demo

  http://cern.ch/security/codetools/c_tools.html#rats

RATS

- C / C++ / Perl, and also (more limited) Python / PHP
- Freeware
- Commonly misused library calls

- Demo

  `http://cern.ch/security/codetools/c_tools.html#rats`

It's already happening!

- PH/SFT used Coverity on ROOT
- 100% path analysis
- optimistic approach
- *Very* satisfactory results

It's already happening!

- PH/SFT used Coverity on ROOT
- 100% path analysis
- optimistic approach
- *Very* satisfactory results

It's already happening!

- PH/SFT used Coverity on ROOT
- 100% path analysis
- optimistic approach
- *Very* satisfactory results

It's already happening!

- PH/SFT used Coverity on ROOT
- 100% path analysis
- optimistic approach
- *Very* satisfactory results

What else?

- "Good, now that I ran the tool, I'm safe..."

- Tools are NOT enough!
- Even the best tool will miss most non-trivial errors!
- Sensitive projects should be reviewed "by hand".

A Fool with a Tool is still a Fool!

- “A fool with a tool is still a fool!”, D. Wheeler
- The code below was found in RealPlayer in 2005. (CVE-2005-0455)

```
char tmp[256]; /* Flawfinder: ignore */
strcpy(tmp, pScreenSize); /* Flawfinder: ignore */
```

A Fool with a Tool is still a Fool!

- “A fool with a tool is still a fool!”, D. Wheeler
- The code below was found in RealPlayer in 2005. (CVE-2005-0455)

```
char tmp[256]; /* Flawfinder: ignore */
strcpy(tmp, pScreenSize); /* Flawfinder: ignore */
```

Website

`http://cern.ch/security/codetools/`

- Tools presentation
- Installation, configuration and usage guidelines
- Explanation of some common vulnerabilities
- Recommendations for creating secure software

Questions

?