

*Connecting people and resources to
accelerate discovery by empowering the
science gateway community*



Cybersecurity for Gateways

SGCI Incubator Bootcamp

April 24-28, 2017

Randy Heiland



Award Numbers
ACI-1547611

Center for Trustworthy Scientific Cyberinfrastructure

The NSF Cybersecurity Center of Excellence



The mission of CTSC is to provide the NSF community with a coherent understanding of cybersecurity, its importance to computational science, and the resources to achieve and maintain an appropriate cybersecurity program.



NSF ACI-1547272

trustedci.org

Overview

- Cybersecurity is part of Sustainability
 - Things can (and do) go wrong
 - But lots of things go right!
- Assets and Risks
- Gateways use software. Software needs to be secure.
- Best Practices for Security
- You're not alone!

Cybersecurity is part of Sustainability

Sustainability requires a high degree of security and stability. This fails if, for example, your system is vulnerable, gets hacked and:

- locked up; held for ransom
- data is stolen/erased/tampered with
- defaced (public humiliation)
- ...

Things can (and do) go wrong

...attacks on unsecured instances of MongoDB ... The attacker erased the database and demanded a ransom be paid before restoring it.

https://www.mongodb.com/blog/post/how-to-avoid-a-malicious-attack-that-ransoms-your-data?utm_campaign=Int_EM_Monthly%20Newsletter_01_17_WW_Auto_enroll&utm_medium=email&utm_source=Eloqua

Library management refused to pay the \$35,000 demanded as ransom, and IT staff wiped affected servers and restored them from available backups.

<https://threatpost.com/st-louis-public-library-recovers-from-ransomware-attack/123297/>



One bug, one crash... of a \$7B rocket that took 10 years to build, due to a floating point overflow software bug. (June 1996)

<https://around.com/ariane.html>



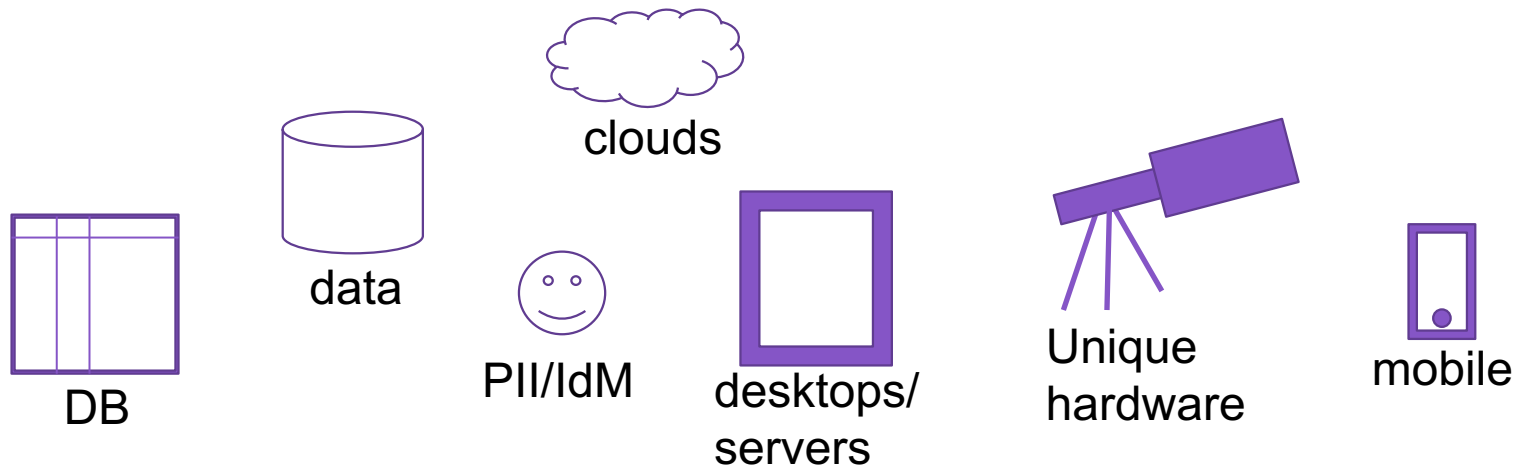
NSF TeraGrid compromised (2004)

https://usatoday30.usatoday.com/tech/news/computersecurity/2004-04-14-synchronized-hacking_x.htm

Assets and Risks

Class exercise (~15 mins)

- Sketch your gateway's & users' assets
- Sketch the flow of information (arrows)
- Try to prioritize the risks



What's the big deal about software?

(drum roll...) Gateways are made of software.

- Software is essential for the bulk of science
 - About half the papers in recent issues of Science were software-intensive projects.
- Software is not a one-time effort, it must be sustained
 - Development, production, and **maintenance** are people intensive.
 - Software life-times are long vs hardware.
 - Software has under-appreciated value.

<http://www.slideshare.net/danielskatz/metrics-citation-for-software-and-data>

Software Security

NSF “CI Framework for 21st century” (CIF21)

Software must be reliable, robust, and secure;
able to produce trustable and reproducible
scientific results; ...

<https://www.nsf.gov/pubs/2012/nsf12113/nsf12113.pdf>

Story time...

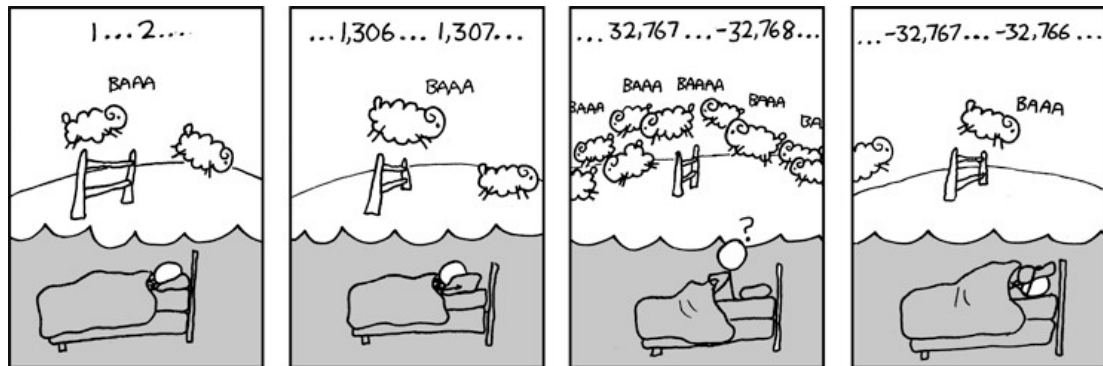
What are some of the worst software-related disasters in history?

Plenty to choose from, sadly...

- Therac-25 radiation therapy machine (1986)
[lack of documentation and sufficient testing]

<http://www.cs.umd.edu/class/spring2003/cmsc838p/Misc/therac.pdf>

- Ariane-5 (1996) [floating point overflow]



<https://xkcd.com/571/>

- Blackout for 50M people (2003) [race condition]

<http://www.securityfocus.com/news/8016>



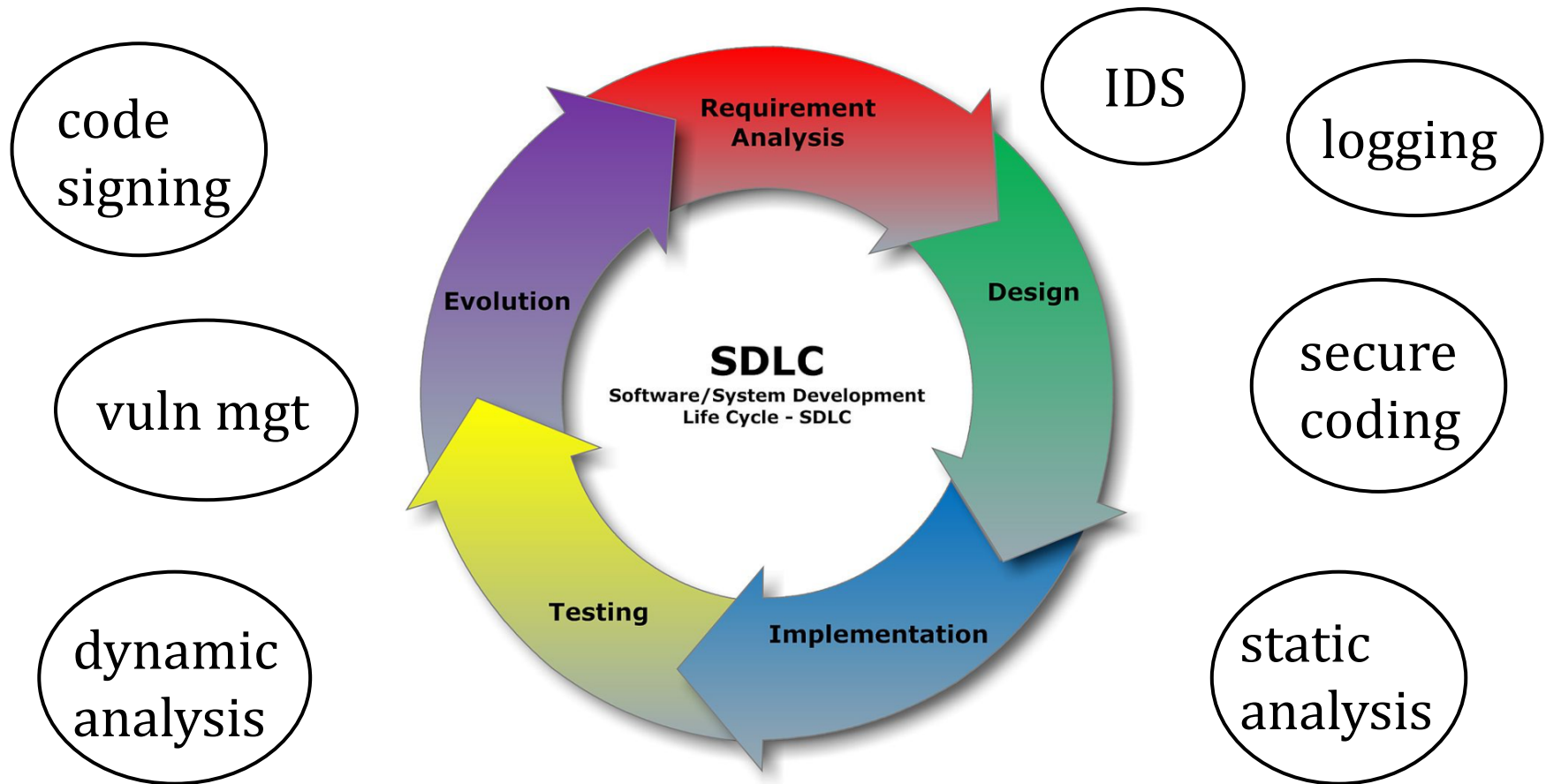
The diagram consists of a large purple rectangular frame. Inside this frame, there are two white rectangular boxes. The left box contains the text 'Software Engineering' and the right box contains the text 'Software Security'. A thin, light gray vertical line separates the two boxes, running from the top to the bottom of the frame.

**Software
Engineering**

**Software
Security**

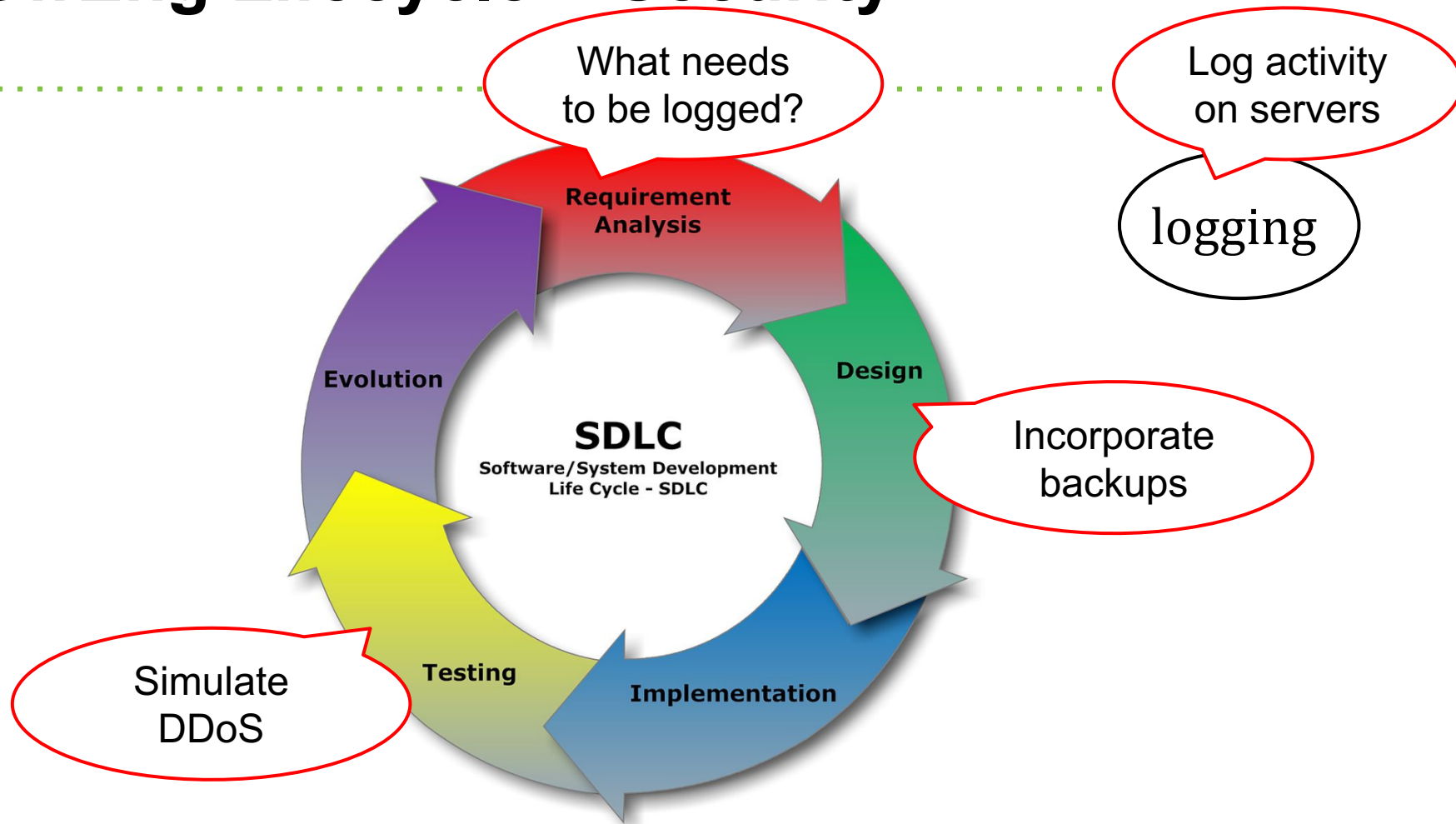
The line between them is fuzzy & almost invisible.

SwEng Lifecycle + Security



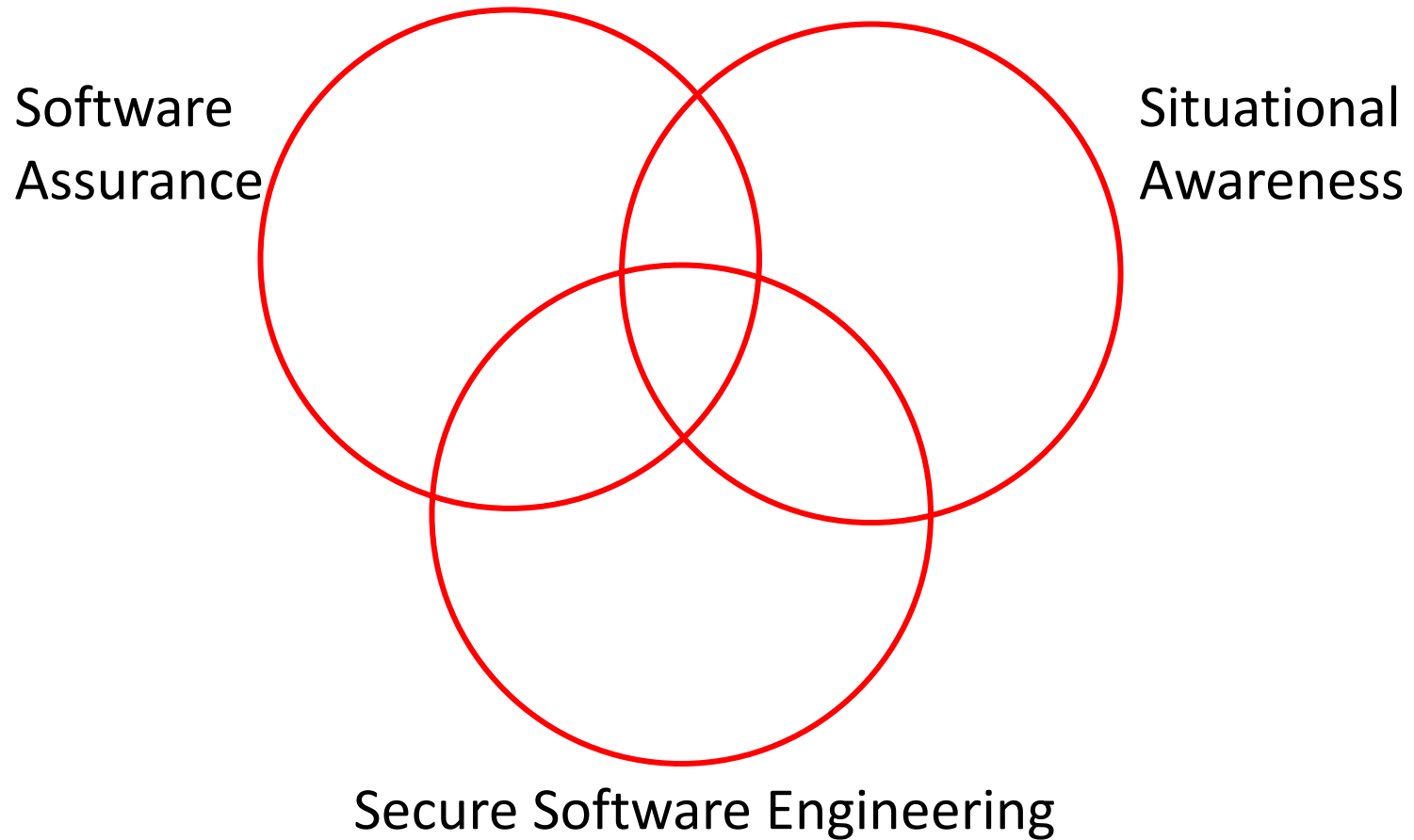
Be security conscious during each phase.

SwEng Lifecycle + Security



Be security conscious during each phase.

Software Security



Software Assurance (SwA)

#1) SwA is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software functions in the intended manner.

#2) The processes (e.g., secure coding, static analysis) that help improve this level of confidence.

→ secure coding instruction (<https://trustedci.org/trainingmaterials>)

Situational Awareness

Being aware of software vulnerabilities and how they might affect a user community. Offering advice on how to patch or update vulnerable software.

<https://trustedci.org/situational-awareness>

<https://blog.trustedci.org/2016/08/situational-awareness.html>

Secure SwEng: Topics

- Repositories/Hosting
- Testing
- Static Analysis
- Vulnerability Management
- Release & Delivery
- Coding/Project Tools
- Documentation

Repositories and Hosting Services

Regardless of the repo/hosting service you choose, be mindful of security considerations:

- physical security
- server logging
- encrypted access
- granularity of access control
- 2FA
- do not commit sensitive data to public repos
 - keep in mind that a currently-private repo may need to be shared more widely later: keep credentials separate from code, or you'll be sanitizing history.

Software Testing



Quality
Assurance
and Testing
Expert

■ Wished we had this

■ Yes, we had this

- why is it necessary?
 - test for “correctness”
 - help prevent bugs, vulnerabilities
 - improve usability
- why is it difficult?
- how well does it work?
- can it be made easier?

Dynamic Testing

- Regression
 - as software is modified, make sure no new (or old) bugs have been introduced
- Combinatorial
 - all combinations of input parameters
- Fuzz
 - with random/noisy inputs
- Security
 - for Confidentiality, Integrity, Availability (CIA)

Testing: think globally, act locally

Acting locally:

Use Assertions in code!

“primary purpose is to instrument code with test probes that will detect errors as close as possible to their place of occurrence.” Tony Hoare, 2002

Assertions

Assertions are always expected to be True:

```
assert(condition)
```

If they are false at runtime, they will throw an error.
(They can be disabled if desired).

C/C++:

```
assert(ptr);  
Assert(x > 0.0);
```

Java:

```
Assert.assertTrue((project1.getCreationTime() -  
                    project2.getCreationTime()) > 0);
```

Story time...

How far back does the idea of using assertions in computer programming go?

“... the programmer should make assertions about the various states that the machine can reach.”

Alan Turing, 1949

<https://pdfs.semanticscholar.org/dfd7/34b2de2cbcce6ac07e909011b0ed6ba32b01.pdf>

Secure SwEng: Topics

- Repositories/Hosting
- Testing
- **Static Analysis**
- Vulnerability Management
- Release & Delivery
- Coding/Project Tools
- Documentation

Static Analysis

Static analysis tools try to find bugs/vulnerabilities in source code. Bugs are then categorized by severity.

Q: why doesn't every software developer use static analysis tools?

A (typically): hassle (time, learning curve), false positives, doesn't catch complex vulnerabilities, ...

Coverity Scan (free for OSS)



Coverity Scan: scilab

Project Name scilab
Lines of code analyzed 685,109
On Coverity Scan since Sep 27, 2013
Last build analyzed about 10 hours ago

Language C/C++
Secondary Language Java
Repository URL [git://git.scilab.org/scilab/](https://git.scilab.org/scilab/)
Homepage URL <http://www.scilab.org/>

Jul 26, 2016

Last Analyzed

685,109

Lines of Code Analyzed

0.35

Defect Density

Defect changes since previous build dated Jul 21, 2016

0

Newly detected

8

Eliminated

Defects by status for current build

3,684

Total defects

243

Outstanding

3,242

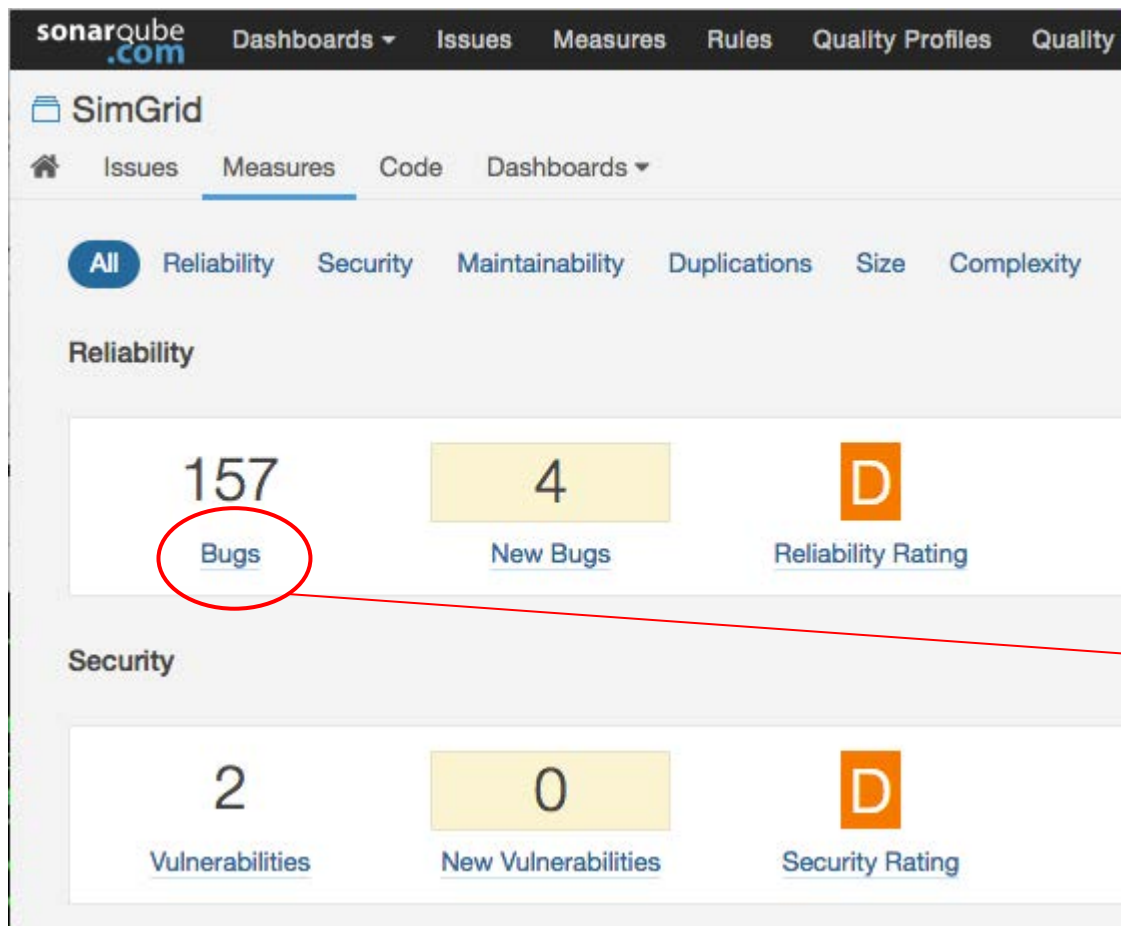
Fixed

number of defects per 1,000 lines of code

SonarQube (sonarqube.com)

"AS3 Core Lib" == GreenMail == Activiti AisLib application framework AngularJS Apache Abdera Apache Asyncweb Parent
Apache Commons BCEL Apache Commons BCEL Apache Commons BeanUtils Apache Commons Chain :: Parent
Apache Commons Collections Apache Commons Configuration Apache Commons DBCP Apache Commons Digester
Apache Commons IO Apache Commons Lang Apache Commons OGNL - Object Graph Navigation Library Apache Commons Pool
Apache Commons SCXML Apache Commons VFS Apache Directory LDAP API Apache Empire-db Apache Gora
Apache Hama parent POM Apache HBase Apache HTTP Server Apache Jackrabbit Apache Lucene
Apache Maven Wagon Apache MINA 3.0.0-M1-SNAPSHOT Apache MyFaces CODI Apache OpenNLP Reactor Apache PhotArk
Apache Pluto Apache Rampart Apache Shindig Project Apache Sirona Incubator Apache Tika Apache Tobago Apache Tomcat
Apache Vysper Parent Apache Wink Apache XBean ApacheDS Mavibot Parent Arakhné Foundation Classes
AssertJ fluent assertions Atlasboard Bash Bonita Camus Parent Cayenne ch-smpp ChakraCore chart.js checkstyle
Clang Closure Library CMake Codeception CodeNarc CodeStory - Fluent-http **CoreCLR** cougar-master-pom
CPython Decompiler DesertOctopus Dijit **disCoverJ** docker-maven-plugin Dojo **Doxia Aggregator** Doxygen Drupal
EasyHook ebms-sed **eclEmma** Elasticsearch: Parent **Fabric8 Maven :: Build** Fitness FlatPack **Flex** Git
Google Cloud Dataflow Java SDK - Parent **GraphT Dependency Injector** haproxy hRaven Project **Hudson** io.joyrn:joyrn ionic
ios-charts **IWS** J2ObjC Gradle Plugin Jackcess **JaCoCo** Jajuk Java Concurrency Stress Tests: Parent
Java Microbenchmark Harness Parent **JavaScript** **JDK 7** **JDK 9** Jenkins main module Jetspeed-2 Enterprise Portal
jGrades Application Jhipster Sample Application JmxTrans - parent project **jolokia-parent** **jOOQ Parent** jQuery JRDS **JUnit**
Kryo Parent lenskit libprelude libpreludedb **lightblue-applications** **lightblue-core** **lightblue-mongo** LogHub **Maven Release**
Maven SCM **Maven-Indexer** **MazarineBlue** **Microsoft Roslyn .NET Compiler Platform** Moneta (JSR 354 RI)
mybatis myob-sdk **MySQL** **NCLServices** **NCLUI** nginx Ninja Notepad++ **ObjectLab Kit** oCanvas opencover
OpenJDK OpenLayers OpenRPGUI OPS4J - Pax Construct **OPS4J Pax Exam (Build POM)** **OPS4J Pax Logging (Build POM)**
ovirt-root Paper.js parent **petclinic** PGJDBC-NG **PHP** **PHP** **PHPUnit** **Pippo Parent** **pltest-parent** **PMD**
POM Parent **PostgreSQL** PostgreSQL JDBC Driver aggregate prewikka project ProjectSend **protobuf** **psi-probe**
PyFFI **Python** react ReactiveUI **Restcomm Sip Servlets** **RestFB** **restfiddle** Restlet Framework **Retrofit (Parent)**
Samba **SeaClouds Platform** **sejda** **sevntu-checks** SimGrid **simple-spring-memcached-parent** sonar-persistit
SonarLint for Eclipse (parent) SonarLint for IntelliJ IDEA **SonarQube** **SonarQube CSS Plugin** **SonarQube Java**
SonarSource :: Language Recognizer **Stapler Parent** Struts 2 Symphony Java Client synthing-android **testing** **TYPO3 CMS**
utPLSQL Vert.x Core **vertx-web-parent** **waffle-parent** **WebGoat** Whirr **Wicket Parent** **XStream Parent** **Yildiz Module Graphic**
Yildiz Module Graphic Ogre YUI

Only the first 200 components are displayed



Not bugs; just weakness.
E.g. class too large;
downcasting.



sonarqube.com

Dashboards ▾ Issues Measures Rules Quality Profiles Quality Gates More ▾

SimGrid

Issues Measures Code Dashboards ▾

Issues Effort

☒ Type

Bug	0
Vulnerability	0
Code Smell	16

☒ Resolution

Unresolved	16	Fixed	344
False Positive	1	Won't fix	0
Removed	0		

☒ Severity

Blocker	16	Minor	722
Critical	210	Info	74
Major	4,998		

SimGrid examples/msg/dht-kademlia/node.c

Do not apply "<<" bitwise operator to a signed operand. ...

Code Smell Blocker Open Not assigned 30min effort

Do not apply "^" bitwise operator to a signed operand. ...

Code Smell Blocker Open Not assigned 30min effort

SimGrid examples/msg/dht-pastry/dht-pastry.c

Do not apply ">>" bitwise operator to a signed operand. ...

Code Smell Blocker Open Not assigned 30min effort

Do not apply "&" bitwise operator to a signed operand. ...

Code Smell Blocker Open Not assigned 30min effort

Static Analysis Plugins: e.g. IntelliJ IDEA + FindBugs

The screenshot displays the FindBugs-IDEA plugin interface within IntelliJ IDEA. The left pane shows a tree view of found bugs, categorized by type such as 'Multithreaded correctness', 'Dodgy', 'Correctness', and 'Performance'. The center pane shows a preview of the code file 'MyClass2.java', with a red box highlighting a specific bug in the 'run()' method. The right pane provides detailed information about the selected bug, including its class, priority, and problem classification.

Found Bugs View

- new org.gnudot.findbugs.MyClass5\$Inne
- new org.gnudot.findbugs.MyClass5() inv
- org.gnudot.findbugs.MyClass5.testMe() i
- new org.gnudot.findbugs.OuterClass02C
- new org.gnudot.findbugs.OuterClass05C
- new org.gnudot.findbugs.OuterNOBug0
- Multithreaded correctness (1 items)
 - EmptySynchronized blocks (1 items)
 - Empty synchronized block (1 items)
 - Empty synchronized block in org.gnudot
- Dodgy (2 items)
 - Dead local store (2 items)
- Correctness (4 items)
 - Unwritten field (2 items)
 - Null pointer dereference (2 items)
 - Read of unwritten field (2 items)
 - Read of unwritten field map in org.gnud
 - Read of unwritten field map in org.gnud
- Performance (11 items)
 - Inner class could be made static (11 items)
 - Could be refactored into a named static inn
 - The class org.gnudot.findbugs.MyClass2
 - The class org.gnudot.findbugs.MyClass2
 - The class org.gnudot.findbugs.MyClass2**
 - The class org.gnudot.findbugs.MyClass5
 - The class org.gnudot.findbugs.MyClass5
 - The class org.gnudot.findbugs.MyClass5

Preview MyClass2.java:

```
37  
38  
39  
40 map.put("1", "value");  
41  
42 final int index = "abcd".indexOf('b');  
43  
44 System.err.println("test");  
45 System.exit(1);  
46  
47 SwingUtilities.invokeLater(new Runnable() {  
48     public void run() {  
49         //To change body of implemented methods use  
50         System.exit(1);  
51     }  
52 });  
53  
54 SwingUtilities.invokeLater(new Runnable() {  
55     public void run() {  
56         //To change body of implemented methods use  
57         System.exit(1);  
58     }  
59 });  
60  
61  
62 class InnerClass01 {  
63  
64     InnerClass01() {  
65         System.exit(1);  
66  
67         SwingUtilities.invokeLater(new Runnable() {
```

The class org.gnudot.findbugs.MyClass2\$1 could be refactored into a named _static_ inner class

Class:
[MyClass2\\$1](#) (org.gnudot.findbugs)

Priority:
Low Priority Performance


Problem classification:
Performance (Inner class could be made static)
SIC_INNER_SHOULD_BE_STATIC_ANON (Could be refactored into a named static inner class)
UnreadFields (NP|SIC|SS|ST|UrF|UuF|UwF)

Could be refactored into a named static inner class
This class is an inner class, but does not use its embedded reference to the object which created it. This reference makes the instances of the class larger, and may keep the reference to the creator object alive longer than necessary. If possible, the class should be made into a static inner class. Since anonymous inner classes cannot be marked as static, doing this will require refactoring the inner class so that it is a named inner class.

the Tools section)

Static analysis as a service: SWAMP



[Access SWAMP](#) [Products](#) [Solutions](#) [About](#) [Blog](#) [Support](#) 

Protect your bits.
The SWAMP is open.

[Register Today](#)

<https://continuousassurance.org/>

Example: Upload, Build, Analyze

The screenshot displays the SWAMP web application. The top navigation bar includes the SWAMP logo, links for 'About', 'Contact', and 'Help', a user greeting 'Welcome, heiland', and a 'Sign Out' button. The left sidebar contains a 'CONTINUOUS ASSURANCE' logo, navigation links for 'Home' and 'My Account', a section titled 'PACKAGES I OWN' with a button for 'airavata', and an 'Add New Package' button. The main content area features a 'Details' tab and the 'airavata Package Profile'. It lists package metadata: 'Package name' (airavata), 'Package type' (Java source), and 'Creation date' (2014-03-04). Below this, it states 'The following versions of this software package are available:' and presents a table with one entry. At the bottom of the main area are buttons for '+ Add Version', 'Edit Package', and 'Delete Package'. The footer contains copyright information for 2014 and the SWAMP logo.

SWAMP [About](#) [Contact](#) [Help](#) Welcome, heiland [Sign Out](#)

airavata Package Profile

Package name airavata
Package type Java source
Creation date 2014-03-04

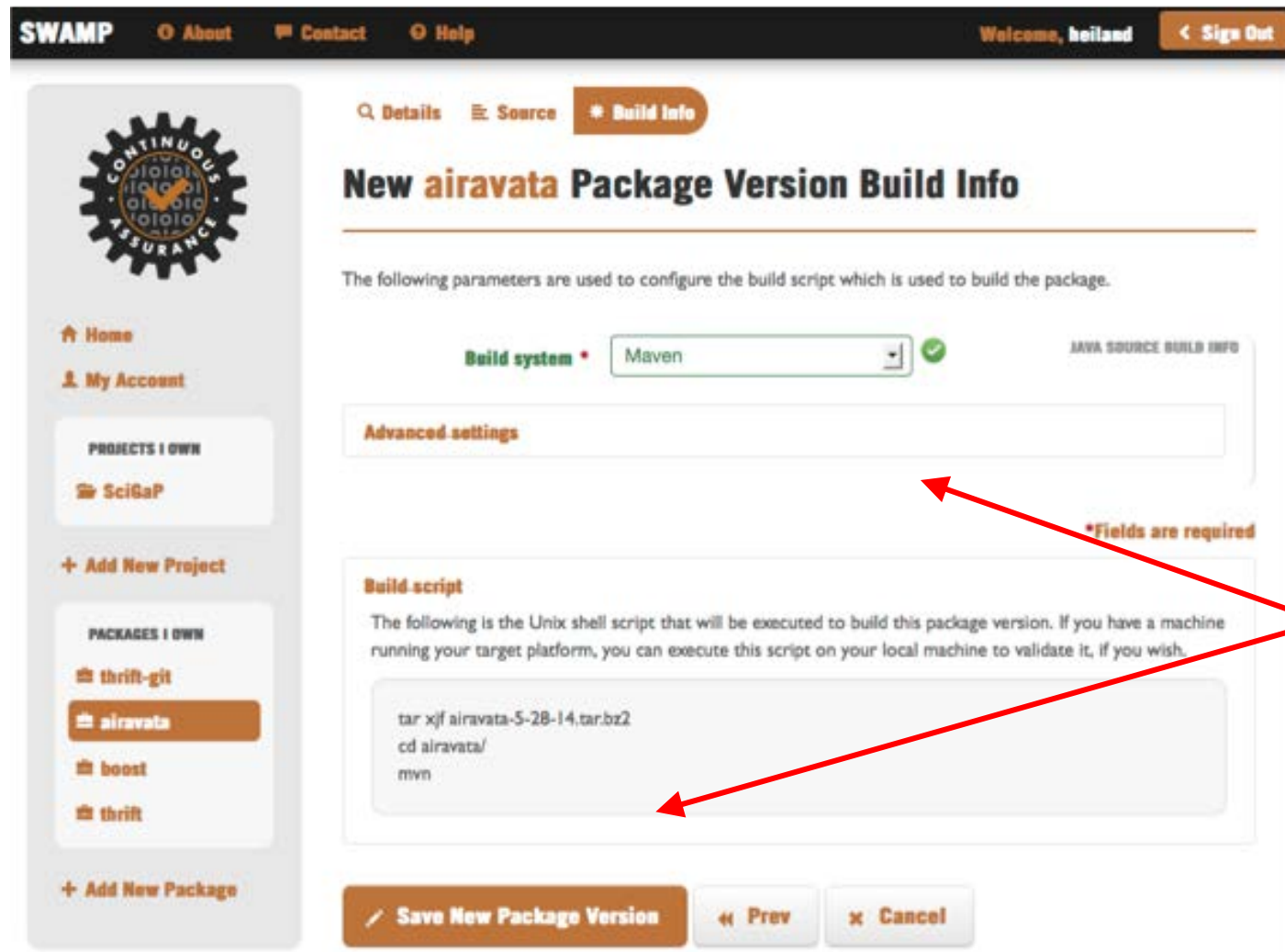
The following versions of this software package are available:

Version	Description	Date
3.4.14	The Apache airavata project, from github.	2014-03-04 12:51

[+ Add Version](#) [Edit Package](#) [Delete Package](#)

Copyright © 2014 Software Assurance Marketplace, Morgridge Institute for Research

Example: Upload, Build, Analyze




SWAMP [About](#) [Contact](#) [Help](#) Welcome, heiland [Sign Out](#)

[Details](#) [Source](#) *** Build Info**

New **airavata** Package Version Build Info

The following parameters are used to configure the build script which is used to build the package.

Build system *  JAVA SOURCE BUILD INFO

Advanced settings

***Fields are required**

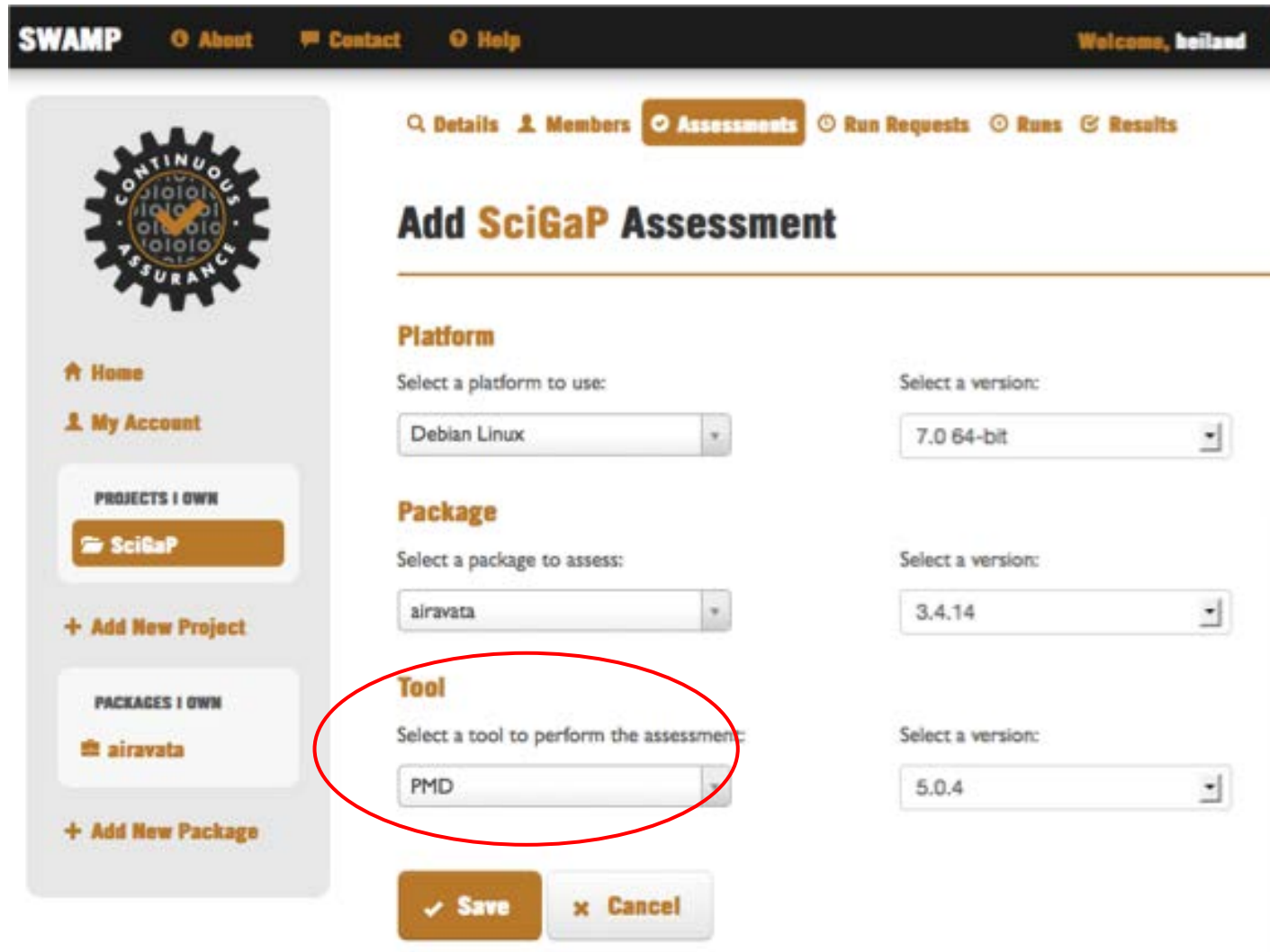
Build-script

The following is the Unix shell script that will be executed to build this package version. If you have a machine running your target platform, you can execute this script on your local machine to validate it, if you wish.

```
tar xjf airavata-5-28-14.tar.bz2
cd airavata/
mvn
```

[Save New Package Version](#) [Prev](#) [Cancel](#)

Example: Upload, Build, Analyze



SWAMP [About](#) [Contact](#) [Help](#) Welcome, heiland

[Details](#) [Members](#) **Assessments** [Run Requests](#) [Runs](#) [Results](#)

Add SciGaP Assessment

Platform
Select a platform to use:
Debian Linux
Select a version:
7.0 64-bit

Package
Select a package to assess:
airavata
Select a version:
3.4.14

Tool
Select a tool to perform the assessment:
PMD
Select a version:
5.0.4

Left Sidebar:
CONTINUOUS ASSURANCE
Home
My Account
PROJECTS I OWN
SciGaP
+ Add New Project
PACKAGES I OWN
airavata
+ Add New Package

Example: Upload, Build, Analyze

Package	Tool	Platform
<input checked="" type="checkbox"/> airavata 3.4.14	PMD 5.0.4	Debian Linux 7.0 64-bit

☐ Schedule Run Requests

Date / Time	Package	Tool	Platform	Status
2014-04-03 10:19	airavata 3.4.14	PMD 5.0.4	Debian Linux 7.0 64-bit	Performing assessment

Date / Time	Package	Tool	Platform	Status
2014-04-03 10:43	airavata 3.4.14	PMD 5.0.4	Debian Linux 7.0 64-bit	Finished

Examples of potential vulnerabilities

cwe.mitre.org - Common Weakness Enumeration:
a dictionary of software weakness types.

- CWE-547: Use of Hard-coded, Security-relevant Constants
- CWE-252: Unchecked Return Value
- CWE-571: Expression is Always True
- CWE-584: Return Inside Finally Block
- CWE-563: Assignment to Variable without Use ('Unused Variable')
- CWE-478: Missing Default Case in Switch Statement
- CWE-495: Private Array-Typed Field Returned From A Public Method

Secure SwEng: Topics

- Repositories/Hosting
- Testing
- Static Analysis
- **Vulnerability Management**
- Release & Delivery
- Coding/Project Tools
- Documentation

Vulnerability: Injection flaw (e.g. SQL, LDAP)

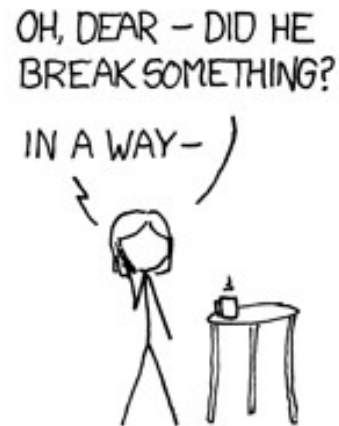
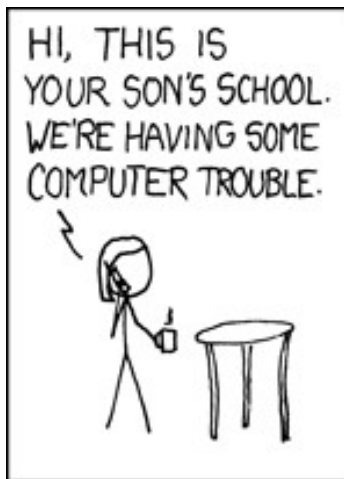
A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application.

A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), ...

Fix: Sanitize user input

(#1 in the OWASP Top 10 handout)

https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet



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

Vulnerable library in your software stack¹

- e.g., OpenSSL Heartbleed (2014)
 - <http://heartbleed.com/>
- Fix: update your version to the patched version
(and deal with repercussions)



[1] Currently #9 in the OWASP Top 10 handout

Vulnerability Management

- Preventing them in the first place (previous slides on developing secure software)
- Detecting them (if they do occur)
- Notifying* appropriate people
- Fixing/Patching
- Testing
- Communicating* fix

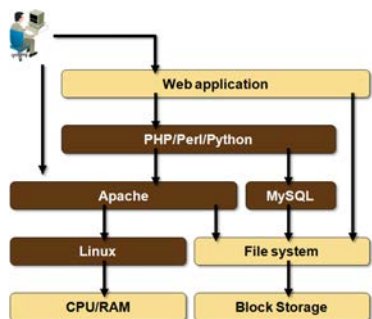
patch: a software update that can be applied to an existing code base in order to eliminate one or more vulnerabilities.

* responsibly, hopefully

Vulnerability Management (cont'd)

It can be complicated:

- software dependencies
- complex configuration
- mission-critical uptime
- difficult to reach resources



LIGO

Story time...

Yes, things can (and do) go wrong, but a LOT of things go right...

e.g., Public Key Infrastructure (PKI)

- Makes e-commerce possible
- Involves asymmetric cryptography
<https://blog.vrypan.net/2013/08/28/public-key-cryptography-for-non-geeks/>
- Uses number theory; difficulty of factoring very large semi-prime numbers.
- “New directions in cryptography”, 1976
<http://dl.acm.org/citation.cfm?id=2269104>
- Turing Award (“Nobel Prize of Computing”) in 2015
<https://awards.acm.org/about/2015-turing>

(Extra point: Who recently won the 2016 Turing Award?)

Secure SwEng: Topics

- Repositories/Hosting
- Testing
- Static Analysis
- Vulnerability Management
- **Release & Delivery**
- Coding/Project Tools
- Documentation

Release & Delivery

How can one help ensure the authenticity and integrity of software (and data)?

- cryptographic checksums, hashes
- SHA- $\{1,2,3\}$ (Secure Hash Alg) ...
- digital signatures (e.g., GPG)

- 1) Download a file
- 2) Compute a hash on it
- 3) Compare to published hash

Your gateway may primarily be a “service”, without much software to download. However, you may provide clients or SDKs, and those should be signed.

Download verification:

Role	Files
Cryptographic Hashes	cmake-3.8.0-SHA-256.txt cmake-3.8.0-SHA-256.txt.asc

b04ea40152633fe351fc60f82b023700dfd84d06b63e3fda87c95b9d01af0cbb cmake-3.8.0.zip

Python-3.6.1.tgz.asc

-----BEGIN PGP SIGNATURE-----

iQIzBAABCAAdFiEEDZbftUEQ5cQ/v7F/LTR+pqplQh0FAIjQ2rQACgkQLTR+pqpl
Qh2Mxw/+NoRiLklaliERGead3xJKLa//WjCnIBoH9dl0SaZwOUkotzkiYOB7+E1C
Ms2Y2h/Ey15JzW4kTfskYanVATKaeVBGwjQQ1GxT0h9EGHQMqZfcxw40vSLOLkn
B1U3G3NkuKdurxgzG4HSZJFu4EIRxYH8DVgovgshWQJXakaSxt0tQedHDgN857X7
JK7O4SFD/pLpX+eV0aMWRxo3Y+QTy/DE4UYiNdqJH/4itawni7ezuB8mcimyp9M8
Eiw+cVCszpjnOidAdwbsihLayvr3KzaqqqE6OVKSLnGSRatt7IjXNWI/0IVJT3HI
dHQuMQqabM4MaDRI5eHkxG5oBGQa/QzoBbSiRGQTnXfOSf5ilwBC2CHZR/zabfP1
tQAHBKfq9Y3feGhQih4Q/diQbyjCEOiSPXorqEDB+GVg2ZcNZdLGmrUSkloPmzEm
wnOh9x935tmSD98VxLM8x3DBCXX8T8nz8052qZdcJNCdP7/ETViaKOUfKZJcFQCQ
3VJH4jEp9GyJq86PFzHX5+72RC87UTZK71xlq03g7HfVNE9bbWBK+2fWXvp/HQE
ntmVS69qBW5sLHfO6gluCvNXaVzEwDJWnBRfB7t5xWDEzhr4c2vX31j2v6/EuUh9
tcQPOp/A0GReyAMMZRCf4SeqLexdrqHKfhloq5wulkLi/F9TVds=
=nTdl


-----END PGP SIGNATURE-----

Secure SwEng: Topics

- Repositories/Hosting
- Testing
- Static Analysis
- Vulnerability Management
- Release & Delivery
- Coding/Project Tools
- Documentation

Coding/Project Tools: Issue Tracking

e.g. JIRA (<https://www.atlassian.com/software/jira>)

**Airavata**
Key: AIRAVATA

[Summary](#)
Issues
[Road Map](#)
[Change Log](#)
[Reports](#)
[Versions](#)
[Components](#)
[Source](#)
[Reviews](#)

Solved: By Component

Component	Issues
Airavata API	5
Airavata Client	3
Airavata Job Monitor	1
 Airavata Orchestrator	3
 Airavata System	14

Status Summary

Status	Issues	Percentage
Open	179	<div><div></div></div> 9%
In Progress	7	<div><div></div></div>
Reopened	14	<div><div></div></div> 1%
Resolved	612	<div><div></div></div> 31%
Closed	1185	<div><div></div></div> 59%

Courtesy of Apache
Airavata project.

Continuous Integration (cloud-based)

Some popular CI tools include:



Travis CI



Bamboo



circleci



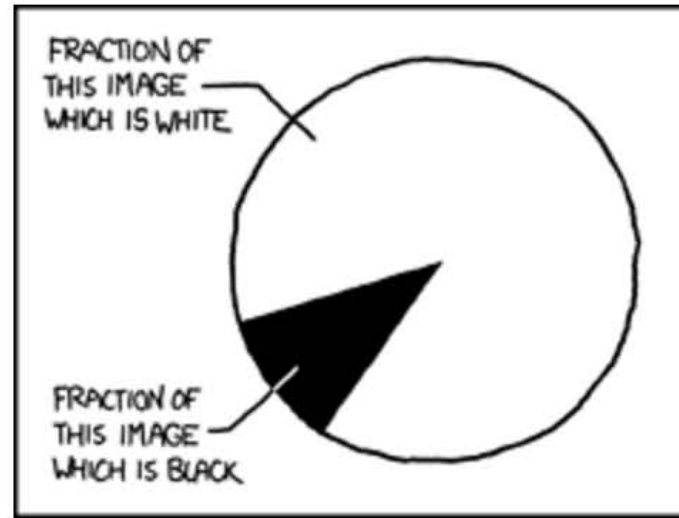
Jenkins

Meyer, M. 2014. "Continuous Integration and Its Tools." *IEEE Software* 31 (3): 14–16.

Secure SwEng: Topics

- Repositories/Hosting
- Testing
- Static Analysis
- Vulnerability Management
- Release & Delivery
- Coding/Project Tools
- Documentation

Some things are self-explanatory.



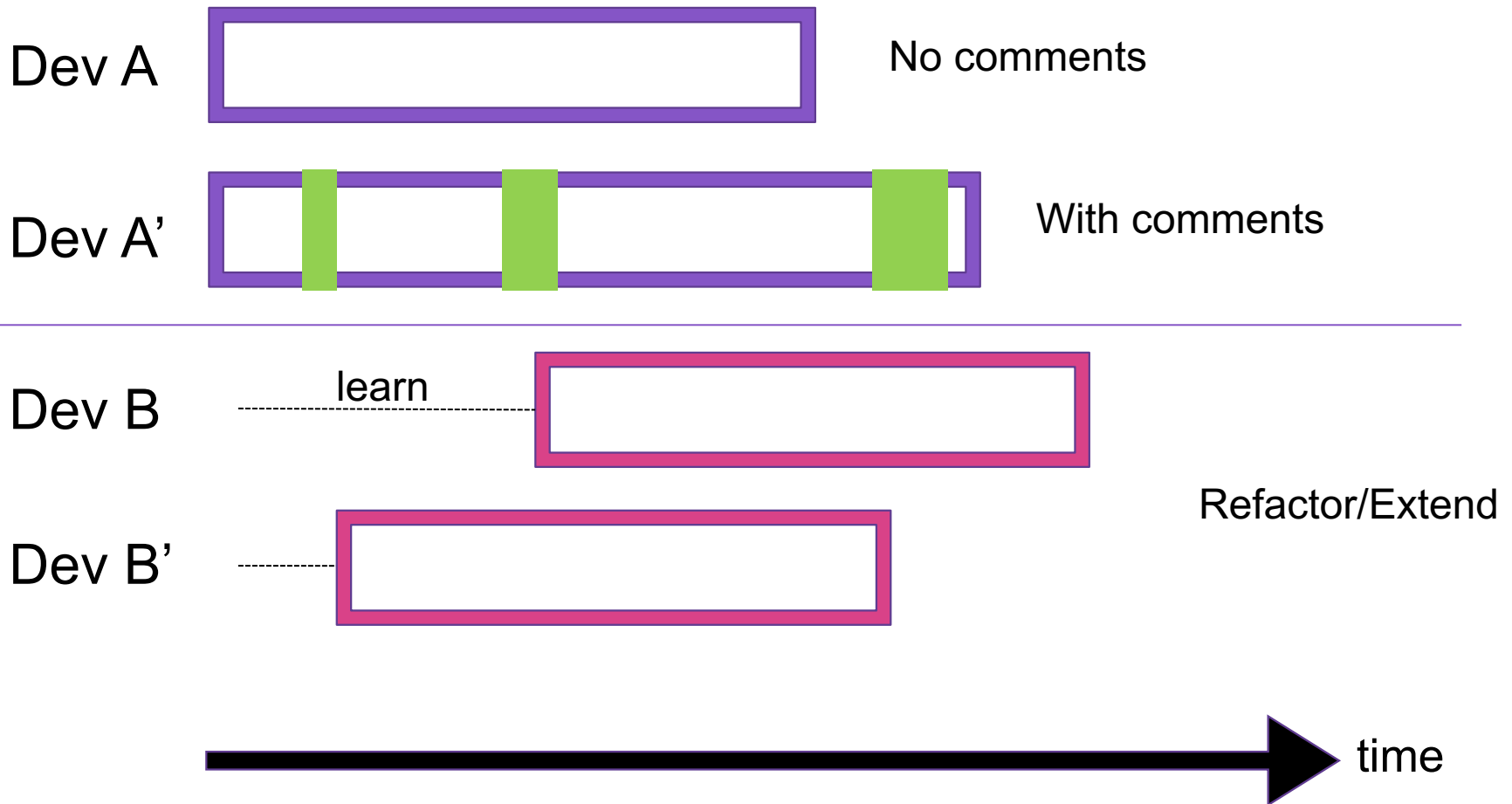
from <https://xkcd.com/688/>

Unfortunately, **Software** is not!

Document your code!

For dev/maintenance, usage, and operation.

Hypothetical Gantt chart for code+docs



Remember the Therac-25 disaster?

around 1978.

Apparently, very little software documentation was produced during development. In a 1986 internal FDA memo, a reviewer lamented, “Unfortunately, the AECL response also seems to point out an **apparent lack of documentation** on software specifications and a software test plan.”

The manufacturer said that the hardware and software were “tested and exercised separately or together over many years.” In his deposition for one of the lawsuits, the quality assurance manager explained that testing was done in two parts. A “small amount” of software testing was done on a simulator, but most testing was done as a system. It appears that unit and **software testing was minimal**, with most effort directed at the integrated system test. At a Therac-25 user group meeting, the same quality assurance man-

Documentation

Document design & purpose, not mechanics.

- a) Document interfaces and reasons, not implementations.
- b) Refactor code in preference to explaining how it works.
- c) Embed the documentation for a piece of software in that software.

Wilson, Greg, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, et al. 2014. "Best Practices for Scientific Computing." *PLoS Biology* 12 (1): e1001745.
[dx.doi.org/10.1371/journal.pbio.1001745](https://doi.org/10.1371/journal.pbio.1001745)

Automatic documentation

Tools exist that generate useful docs for your code if you include that documentation in your code and follow the tools' syntactic rules.

- motivation for embedding your documentation
- generates easy-to-navigate HTML/Latex/etc docs

Javadoc: Generates HTML pages of API documentation from Java source files

/**

- * Returns an Image object that can then be painted on the screen.
- * The url argument must specify an absolute {@link URL}. The name argument is a specifier that is relative to the url argument.

* <p>

- * This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

*

* @param url an absolute URL giving the base location of the image.

* @param name the location of the image, relative to the url argument

* @return the image at the specified URL

* @see Image

*/

```
public Image getImage(URL url, String name) {  
    try {  
        return getImage(new URL(url, name));  
    } catch (MalformedURLException e) {  
        return null;  
    }  
}
```

getImage

```
public Image getImage(URL url,  
    String name)
```

Returns an `Image` object that can then be painted on the screen. The `url` argument must specify an absolute URL. The `name` argument is a specifier that is relative to the `url` argument.

This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

Parameters:

`url` - an absolute URL giving the base location of the image.

`name` - the location of the image, relative to the `url` argument.

Returns:

the image at the specified URL.

See Also:

`Image`

java.awt

```
java.lang.Object
    java.awt.Toolkit
```

This class is the abstract superclass of all actual implementations of the Abstract Window Toolkit. Subclasses of the Toolkit class are used to bind the various components to particular native toolkit implementations.

Many GUI events may be delivered to user asynchronously, if the opposite is not specified explicitly. As well as many GUI operations may be performed asynchronously. This fact means that if the state of a component is set, and then the state immediately queried, the returned value may not yet reflect the requested change. This behavior includes, but is not limited to:

- Scrolling to a specified position.
For example, calling `ScrollPane.setScrollPosition` and then `getScrollPosition` may return an incorrect value if the original request has not yet been processed.
- Moving the focus from one component to another.
For more information, see [Timing Focus Transfers](#), a section in [The Swing Tutorial](#).
- Making a top-level container visible.
Calling `setVisible(true)` on a `Window`, `Frame` or `Dialog` may occur asynchronously.
- Setting the size or location of a top-level container.
Calls to `setSize`, `setBounds` or `setLocation` on a `Window`, `Frame` or `Dialog` are forwarded to the underlying window management system and may be ignored or modified. See `Window` for more information.

Doxygen

“Doxygen is the de facto standard tool for generating documentation from annotated C++ sources, but it also supports other popular programming languages such as C, Objective-C, C#, PHP, Java, Python, IDL, Fortran, VHDL, Tcl, ...”

C/C++: Docs annotation is inserted into headers (.h):

```
// .NAME classname - brief description
// .SECTION Description
// more detailed description
...
// Description:
// Assign a data object as input. Note that this method ...
void SetInputData(int index, vtkDataObject* obj);
```

Public Types

typedef **vtkAlgorithm** Superclass

► Public Types inherited from **vtkAlgorithm**

► Public Types inherited from **vtkObject**

Public Member Functions

virtual int **IsA** (const char *type)

vtkUndirectedGraphAlgorithm * **NewInstance** () const

void **PrintSelf** (ostream &os, **vtkIndent** indent)

virtual int **ProcessRequest** (**vtkInformation** *, **vtkInformationVect**

vtkUndirectedGraph * **GetOutput** ()

vtkUndirectedGraph * **GetOutput** (int index)

void **SetInputData** (**vtkDataObject** *obj)

void **SetInputData** (int index, **vtkDataObject** *obj)

► Public Member Functions inherited from **vtkAlgorithm**

► Public Member Functions inherited from **vtkObject**

► Public Member Functions inherited from **vtkObjectBase**

Static Public Member Functions

static **vtkUndirectedGraphAlgorithm** * **New** ()

static int **IsTypeOf** (const char *type)

static **vtkUndirectedGraphAlgorithm** * **SafeDownCast** (**vtkObjectBase** *o)

► Static Public Member Functions inherited from **vtkAlgorithm**

Operational Security

Now that your gateway **software** is 100% secure¹, what about securely **operating** it?

- Network communication protocols
- Monitor traffic
- Log activity on servers
- Identity Management
- Assign responsibility for security to someone²



² Consider university personnel

Monitoring (IDS¹) and Logging



- Detect the unexpected/unwanted
- Network traffic (e.g., Bro IDS)
- Log events for later analysis

[1] IDS: Intrusion Detection Systems

Gateway Security Best Practices

Ask Questions!

Software:

- Use software repos & hosting (e.g. GitHub); use 2FA
- Use Continuous Integration on your repo
- Static analysis on code – and automate it
- Test, test, test – and automate it
- Comment/document – and automate it
- Securely hash/sign (e.g. SHA-2) your code releases

Operation:

- Use https
- Monitor traffic on your gateway (e.g. Snort or Bro)
- Log activity on servers; Perform vulnerability scans
- Consider outsourcing Identity Management¹ (e.g. CILogon)

[1] <https://blog.trustedci.org/2014/04/idm.html>

You're not alone



We're here to help

- This cohort – an ongoing support group
- SGCI and CTSC - join in!
 - join mailing lists & webinars
 - follow us on social media
- Ask questions!
- Be an **active** community member!

Additional reading:

<https://trustedci.org/trainingmaterials/>

Thanks!

Discussion time?