Software Assurance Tools at Indiana University:

A Journey into the SWAMP

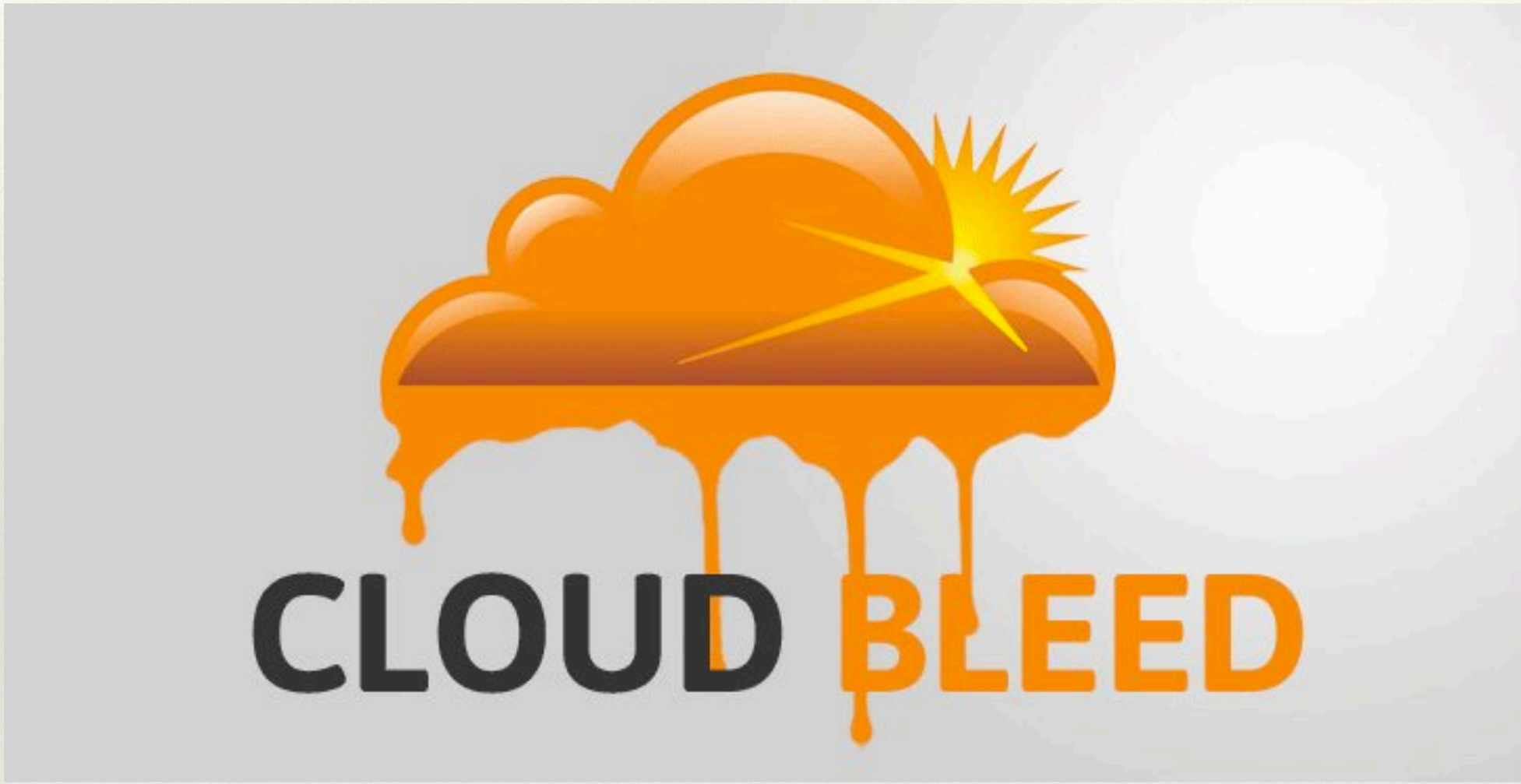
Rob Quick
Research Technologies Manager High Throughput Computing Operations Officer - OSG
Operations Officer - SWAMP

Buffer Overflow Example

- Some people believe that buffer overflows are ancient history, but…
  - Failure of an important library to validate the length field (as compared to the size of the actual message).
  - The heartbeat protocol is supposed to echo back the data sent in the request where the amount is given by the payload length.
  - Since the length field is not checked, `memcpy` can read up to 64KB of memory.

B. Miller and E. Haymann

More Recently…



- February 17, 2017 a bug in Cloudflare's reverse proxies caused their edge servers to **run past the end of a buffer**. This led to **return of memory that contained private information** such as HTTP cookies, authentication tokens, HTTP POST bodies, and other sensitive data.

Top 10 Common Weakness Enumerations

- CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('**SQL Injection**')
- CWE-78 Improper Neutralization of Special Elements used in an OS Command ('**OS Command Injection**')
- CWE-120 Buffer Copy without Checking Size of Input ('**Classic Buffer Overflow**')
- CWE-79 Improper Neutralization of Input During Web Page Generation ('**Cross-site Scripting**')
- CWE-306 Missing Authentication for Critical Function
- CWE-862 Missing Authorization
- CWE-798 Use of Hard-coded Credentials
- CWE-311 Missing Encryption of Sensitive Data
- CWE-434 Unrestricted Upload of File with Dangerous Type
- CWE-807 Reliance on Untrusted Inputs in a Security Decision

http://cwe.mitre.org/top25/archive/2011/2011_cwe_sans_top25.html#Listing

Successful SQL Injection Attack

2. DB Queried

```
SELECT * FROM members
WHERE u='admin' AND p='' OR 'x'='x'
```

3. Returns all row of table members

JAVA

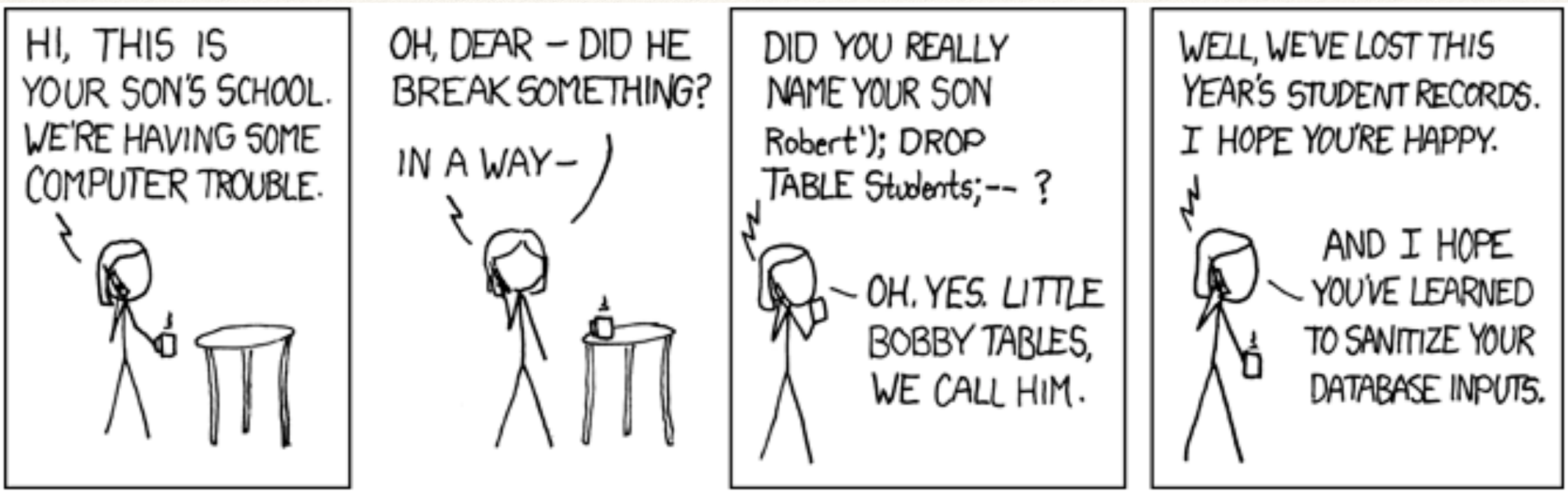
1. User sends malicious data

```
user="admin"; pwd="'OR 'x'='x"
```

```
boolean Login(String user, String pwd)  {
    boolean loggedIn = false;
    conn = pool.getConnection( );
    stmt = conn.createStatement();
    rs = stmt.executeQuery("SELECT * FROM members"
                           + "WHERE u='" + user
                           + "' AND p='" + pwd + "'");
    if (rs.next())
        loggedIn = true;
}
```

4. System grants access

```
Login() returns true
```



https://xkcd.com/327/

Successful OS Injection Attack

JAVA

1. User sends malicious data

```
hostname="x.com;rm -rf /*"
```

2. Application uses nslookup to get DNS records

```
String rDomainName(String hostname)  {
  …
  String cmd = "/usr/bin/nslookup " + hostname;
  Process p = Runtime.getRuntime().exec(cmd);
  …
```

3. System executes `nslookup x.com;rm -rf /*`

4. All files possible are deleted

B. Miller and E. Haymann

Reflected Cross Site Scripting (XSS)

JAVA

3. Generated HTML displayed by browser

```
<html>
…
You searched for:
<script>alert('Boo!')</script>
…
</html>
```

1. Browser sends request to web server

```
http://example.com?q=<script>alert('Boo!')</script>
```

2. Web server code handles request

```
…
String query = request.getParameter("q");
if (query != null) {
    out.writeln("You searched for:\n" + query);
}
…
```

B. Miller and E. Haymann

Thinking Like an Attacker

Exploit: A manipulation of a program's internal state in a way not anticipated by the programmer.

Start at the user's entry point: The attack surface.

- Network input buffer
- Field in a form
- Line in an input file
- Environment variable
- Program option
- Entry in a database
- …

B. Miller and E. Haymann

SwA Lessons Learned from Heartbleed

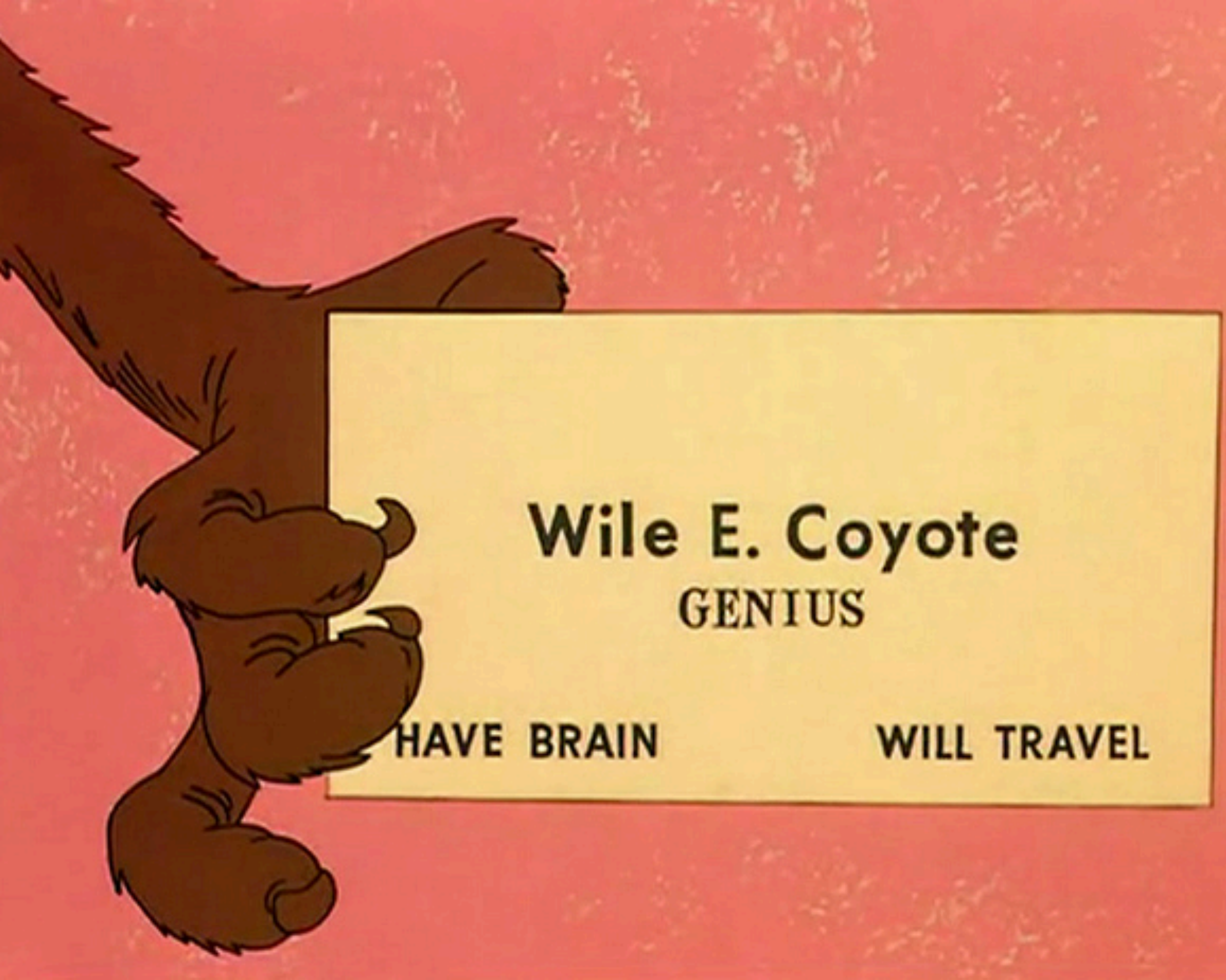
- "Why Do Software Assurance Tools Have Problems Finding Bugs Like Heartbleed?"
  - James A. Kupsch and Barton P. Miller https://continuousassurance.org/swamp/SWAMP-WP003-Heartbleed.pdf
- Software of sufficient complexity cannot be successfully analyzed.
- Software needs to be developed in such a way it can be assured.
- Early application of SwA framework helps achieve this.

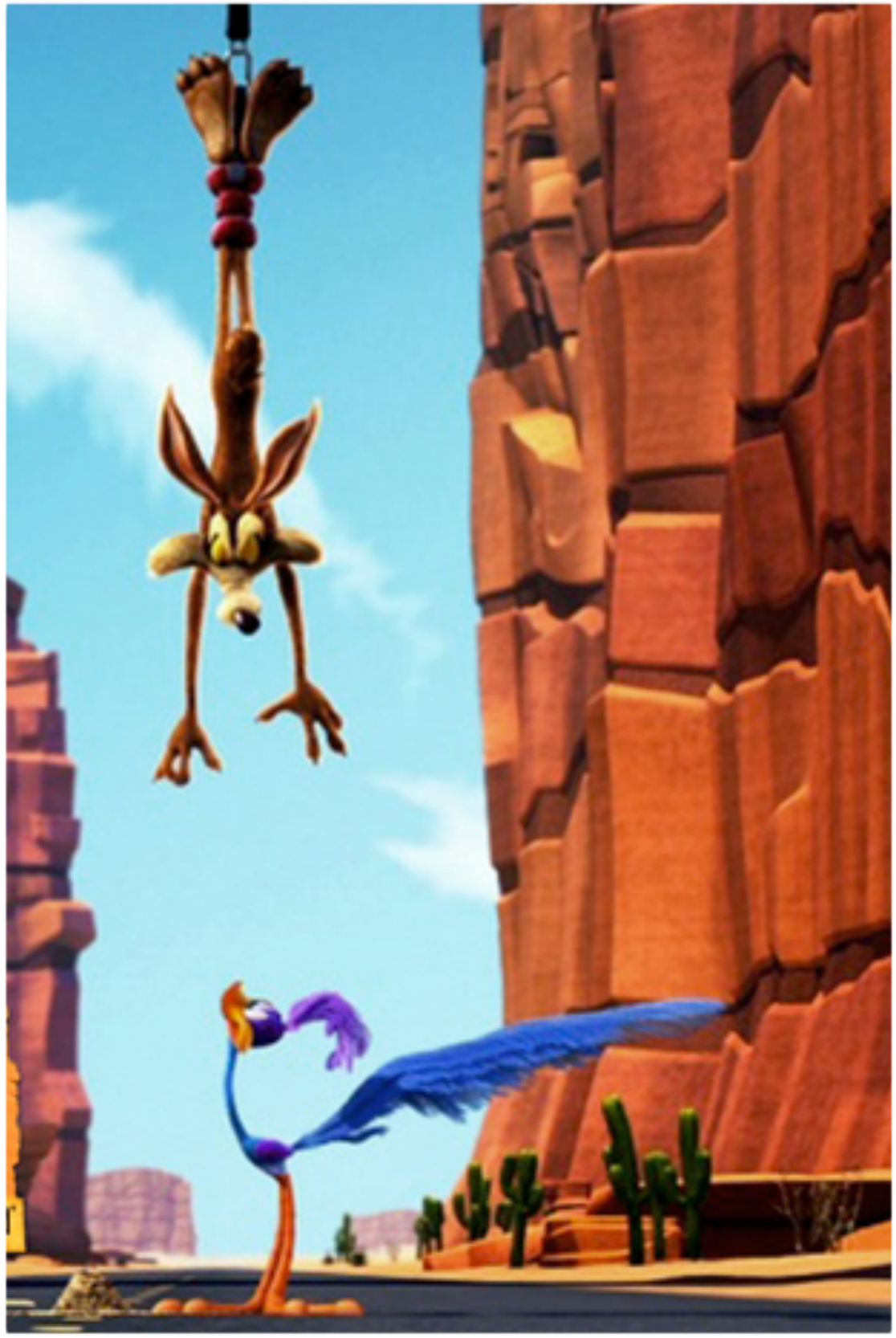
V. Welch

Software Assurance Motivation

- The world we live in today is **software-centric**, introducing **significant risks** to confidential data and physical resources
- Applications are leaving the protected enterprise network environment and moving onto the web
- Anything with an outward face to the Internet is a entry point for an attack
- Few developers are trained and equipped to build secure code
- Even those well equipped often utilize code developed by others

V. Welch


Wile E. Coyote
GENIUS
HAVE BRAIN
WILL TRAVEL




HELP!

The Tools

- Key assets in this battle are the software assessment tools that can scan the program for defects(weaknesses). However, using these tools comes with challenges:
- Each tool is good at finding some particular problem; no tool is good at everything (or even most things).
- Configuring, maintaining, and using these tools can be cumbersome, time consuming and tricky.

V. Welch

A Framework

- No single Software Assurance(SwA) tool is going to bridge the gap between software and assured software.
- A software assurance (SwA) framework allows construction and automation of SwA workflows.
- Our framework provides code analysis, result normalization and labeling, result merging and integration, visualization, result evaluation and annotation, and risk assessment.
- Aggregates, orchestrates and automates use of SwA tools rather than being a tool itself.
- Should support use cases of software developers, SwA Tool developers, SwA researchers, software users, and educators.

V. Welch

Welcome to the SWAMP

- A continuous assurance platform that enables significant improvements in the quality of SwA tools while broadening adoption of SwA methodologies
- Consists of:
  - 30(and growing) static analysis assessment tools
  - State-of-the-art assessment results viewer
  - “Plumbing” that simplifies access to SwA tools
  - Provides a hub for software assurance projects
  - Supports managed access to tools, packages and results
  - Maintains confidentiality of software and results at the discretion of the user

SWAMP
SOFTWARE ASSURANCE MARKETPLACE
Do It Early. Do It Often.

V. Welch

Vision of Continuous Assurance

- **Continuous integration (CI)** is the practice, in software engineering, of merging all developer working copies with a shared mainline several times a day.

- **Continuous Assurance (CoA)** takes the software engineering practice of Continuous Integration to a new level. CoA incorporates SwA tools into the frequent process of building and testing the software throughout its life cycle.

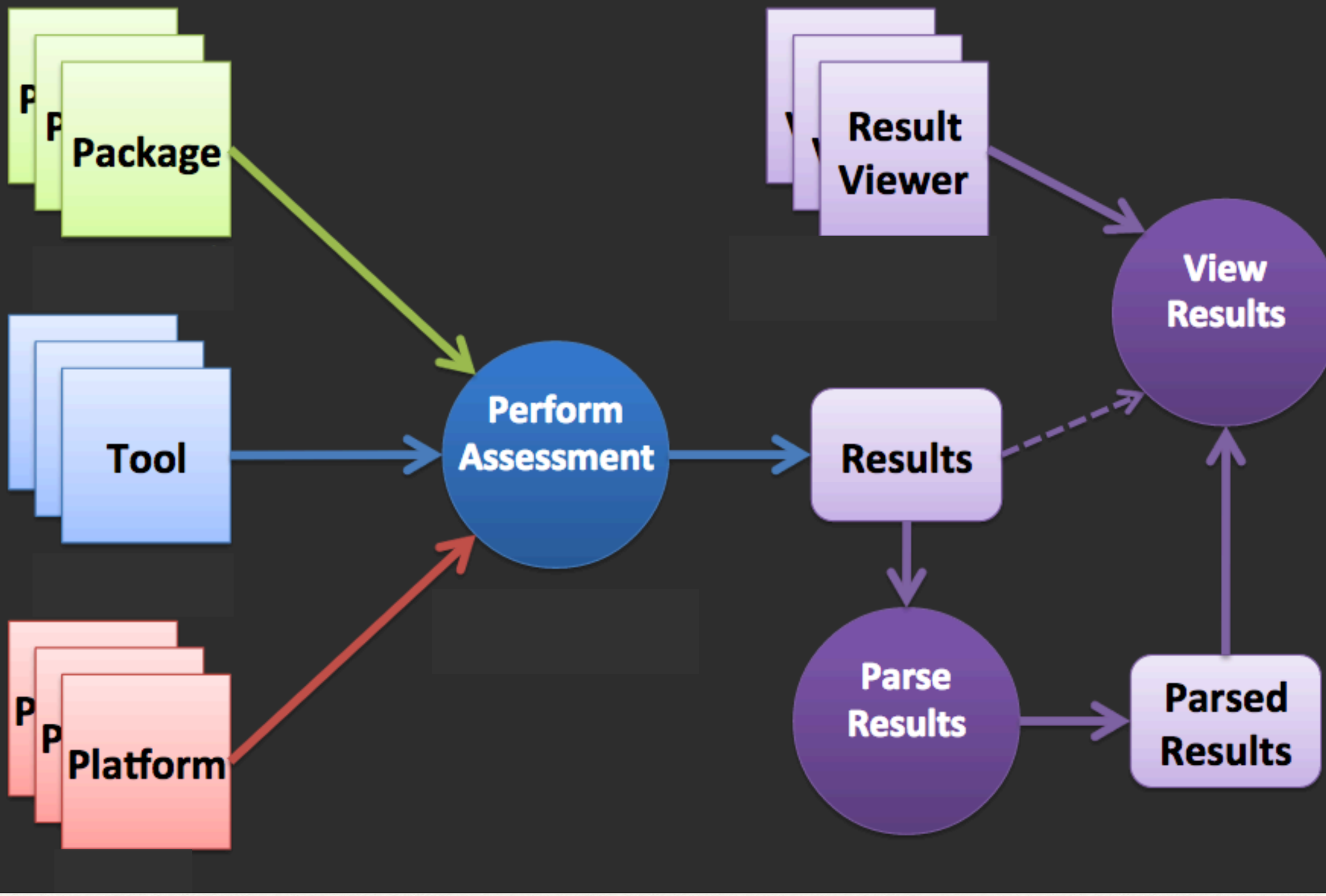
V. Welch


ACME
MEGA-MOTOR
HELMET
CEILING FAN
ACME
ACME
BONNIE BIKE

What is the SWAMP?

- The Software Assurance Marketplace (SWAMP) is a service that provides continuous software assurance capabilities to developers and researchers.
- This no-cost code analysis service is open to the public. Let the SWAMP help you to build better, safer, and more secure code today!

SWAMP
SOFTWARE ASSURANCE MARKETPLACE
Do It Early. Do It Often.


Package
Result Viewer
View Results
Tool
Perform Assessment
Results
Platform
Parse Results
Parsed Results

The SWAMP

- While proven useful for SwA there was a trust gap between developers and the Morgridge Institute for Research run, Department of Homeland Security funded facility
- The SWAMP-in-a-box (SiB) framework allows local administrators to host a Software Assurance framework
  - Though some commercial tools and power are sacrificed by using local instances
- Both SWAMP and SiB provide strong sandboxing
  - VM created for the assessment. Upon completion the VM is destroyed leaving only the assessment report

Languages Supported

SWAMP

- C/C++
- Java source
- Java bytecode
- Python
- Ruby
- PHP
- Javascript
- HTML
- CSS
- XML

SiB

- C/C++
- Java source
- Java bytecode
- Python
- Ruby
- **Coming Soon:**
  - PHP
  - Javascript

Tools Supported

SWAMP

- **Open tools**
  - Android lint
  - Bandit
  - Brakeman
  - checkstyle
  - Clang Static Analyzer
  - cppcheck
  - CSS Lint
  - Dawn
  - error-prone
  - ESLint
  - Findbugs
  - Flake8
  - Flow
  - GCC
  - HTML Tidy
  - JSHint
  - OWASP Dependency Check
  - PHPMD
  - PHP_CodeSniffer
  - PMD
  - Pylint
  - Reek
  - Ruby-lint
  - Retire.js
  - RevealDroid
  - RuboCop
  - ruby-lint
  - XML Lint
- **Commercial tools**
  - GrammaTech CodeSonar
  - Parasoft C/C++test
  - Parasoft Jtest

SiB

- Bandit
- Brakeman
- checkstyle
- Clang Static Analyzer
- cppcheck
- Dawn
- error-prone
- Findbugs
- Flake8
- GCC
- OWASP Dependency Check
- PMD
- Pylint

Platforms Supported

SWAMP

- Android
- CentOS Linux 5 32-bit and 64-bit
- CentOS Linux 6 32-bit and 64-bit
- Debian Linux
- Fedora Linux
- Red Hat Enterprise Linux 6 32-bit and 64-bit
- Scientific Linux 5 32-bit and 64-bit
- Scientific Linux 6 32-bit and 64-bit
- Ubuntu Linux
- **Upcoming:**
  - Mac OS X
  - Microsoft Windows

SiB

- Ubuntu Linux
- **Upcoming:**
  - Mac OS X
  - Microsoft Windows

Weakness Details

High Severity Weaknesses (2)

Memory leak detected by **Clang**

CWE-401 Improper Release of Memory Before Removing Last Reference ('Memory Leak')

Instances (2)

Path: glite-ce-cream-utils-1.2.0-4.sl6.zip/src (2)

Location: glite-ce-cream-utils-1.2.0-4.sl6.zip/src/glite-cream-createsandboxdir.cpp:126-126
ID: 21504
Status: Unresolved
Details: Potential leak of memory pointed to by 'directoryToCreate'
Code Snippet:

```
123     errOccurred = mkdir(directoryToCreate, 0700);
124     checkErrno(errOccurred);
125
126     string jobDir = directoryToCreate;
127
128     errOccurred = mkdir((jobDir + "/ISB").c_str(), 0700);
129     checkErrno(errOccurred);
```

Location: glite-ce-cream-utils-1.2.0-4.sl6.zip/src/glite-cream-createsandboxdir.cpp:115-115
ID: 21505
Status: Unresolved
Details: Potential leak of memory pointed to by 'first2charJobId'
Code Snippet:

```
112         strncpy(first2charJobId, jobId+5,2);
113
114         strcat(directoryToCreate, first2charJobId);
115         strcat(directoryToCreate, "/");
116
117         errOccurred = mkdir(directoryToCreate, 0700);
118         checkErrno(errOccurred);
```

401 Improper Release of Memory Before Removing Last Reference ('Memory Leak')

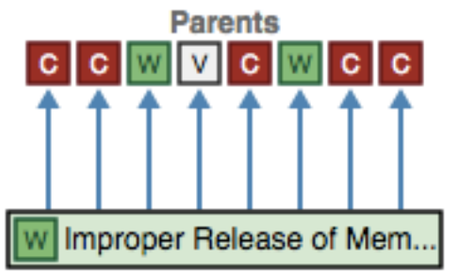
CWE Page
Permalink

Description

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Relationships



Time of Introduction

- Architecture and Design
- Implementation

Languages

- C
- C++

Likelihood of Exploit

- Medium

http://siab.grid.iu.edu/

SWAMP

SOFTWARE ASSURANCE MARKETPLACE

Do It Early. Do It Often.

The Software Assurance Marketplace (SWAMP) is a service that provides continuous software assurance capabilities to developers and researchers.

This no-cost code analysis service is open to the public. Let the SWAMP help you to build better, safer, and more secure code today!

Sign Up!

Get results in just three steps:

Rather than spending time installing, licensing and configuring software assessment tools on your own machine, let the SWAMP do the work for you.

1) Upload your package

First, upload your code. Rest assured that it will remain private and secure.

2) Run your assessment

Next, create and run an assessment by choosing a package, tool, and platform.

3) View your results

Last, view your results using a native viewer or Code Dx™ for full featured analysis.

Test Instance

- Questions:
  - **Is this useful** to IU based coders?
    - Can they be convinced to **start SwA at the onset** of a project?
    - What are the **barriers to adopting SwA** policy?
- How does it perform?
  - If this becomes a production facility what resources will need to be allocated to it?
- How much effort does it take to **support** a SiB instance?
- What are the bugs in the framework?
- How does SiB compare to SWAMP?

WILE E. COYOTE

You Know You Just Can't Win When You're Stuck
Between A Rock And Another Rock

- SWAMP Website: https://continuousassurance.org/
- SiB Info: https://continuousassurance.org/swamp-in-a-box/
- Von's full slide set: http://www.vonwelch.com/pres/SWAMP-Regenstrief-Sep-2014.pdf
- Bart's and Elisa's full slide set: https://static1.squarespace.com/static/5047a5a6e4b0dcecada15549/t/54071f4ce4b00e19c7ef11c9/1409752908265/Miller-Heymann-NSF-2014.pdf
- SWAMP-in-a-Box git repo https://github.com/mirswamp/deployment

Now, go use SWAMP or SiB and give us your feedback!!!

https://github.com/mirswamp/deployment

SwA WG in WISE?

- Gauging interest from those attending today.
- Proposal of working session on SwA
  - Bring code and we'll do live assessments
  - Talk about what to do with the results

Additional Mitigation Slides from B. Miller and E. Haymann

Mitigated SQL Injection Attack

```
SELECT * FROM members WHERE u = ?1 AND p = ?2
        ?1 = "admin"    ?2 = "' OR 'x'='x"
```

2. DB Queried

3. Returns null set

JAVA

1. User sends malicious data

```
user="admin"; pwd="' OR 'x'='x"
```

```
boolean Login(String user, String pwd)  {
    boolean loggedIn = false;
    conn = pool.getConnection( );
    PreparedStatement pstmt = conn.prepareStatement(
        "SELECT * FROM members WHERE u = ? AND p = ?");
    pstmt.setString( 1, user);
    pstmt.setString( 2, pwd);
    ResultSet results = pstmt.executeQuery( );
    if (rs.next())
        loggedIn = true;
}
```

4. System does not grant access

```
Login() returns false
```

Mitigated OS Injection Attack

JAVA

1. User sends malicious data

```
hostname="x.com;rm -rf /*"
```

2. Application uses nslookup only if input validates

```
String rDomainName(String hostname)  {
  …
  if (hostname.matches("[A-Za-z][A-Za-z0-9.-]*")) {
    String cmd = "/usr/bin/nslookup " + hostname);
    Process p = Runtime.getRuntime().exec(cmd);
  } else {
    System.out.println("Invalid host name");
    …
```

3. System returns error `"Invalid host name"`

XSS Mitigation

JAVA

3. Generated HTML displayed by browser

```
<html>
...
Invalid query
...
</html>
```

1. Browser sends request to web server

```
http://example.com?q=<script>alert('Boo!')</script>
```

2. Web server code correctly handles request

```
...
String query = request.getParameter("q");
if (query != null) {
    if (query.matches("^\\w*$"))  {
        out.writeln("You searched for:\n" + query);
    }  else  {
        out.writeln("Invalid query");
    }
}
...
```