THE UNFORTUNATE REALITY OF INSECURE LIBRARIES

Jeff Williams, CEO

OWASP AppSec DC
April 4, 2012
Warning

- Nothing in the following presentation or associated paper should be read to imply that open source is any more or less insecure than commercial software.

- The authors are strong proponents of open source and have contributed many projects under open source licenses, including two of the libraries studied.

- Any attempt to claim otherwise after seeing this warning will be subject to public scorn and humiliation.

Stallman Clarification

This study is NOT ABOUT OPEN SOURCE. It’s about our infrastructure and old vulnerable software.
How Do I Prevent This?

The primary recommendations are to establish all of the following:

... 

2. A process for keeping abreast of and deploying all new software updates and patches in a timely manner to each deployed environment. This needs to include all code libraries as well, which are frequently overlooked.”
We asked 2,550 developers...

How do you know when a component is updated?

- 74% By searching the web
- 66% Keeping up with project sites
- 40% From colleagues
- 30% Word of mouth
- 20% No good way to find out
The Study

Over half the Global 500 Undercounted!

Dataset | Value
--- | ---
Libraries | 31
Library Versions | 1,261
Organizations | 61,807
Downloads | 113,939,358

The Libraries

Frameworks
- Spring MVC
- GWT
- Apache CXF
- Hibernate
- Java Servlet
- Apache Velocity
- Struts 1.x
- Apache Xerces
- Apache Axis
- Struts2
- Java Server Pages
- Tiles
- Wicket
- Lift
- Tapestry
- Java Server Faces
- JAX-WS
- Grails
- Stripes
- JBoss Seam

Security
- Log4j
- Spring Security
- ESAPI
- Apache Commons
- Validator
- Hibernate Validator
- Apache Santuario
- Jasypt
- Apache Shiro
- BouncyCastle
- AntiSamy
- HDIV
The Organizations

1403 Organizations Using ESAPI
Who Is Building Libraries?
Trust Your Business to Your Libraries?

- Our open source team approved them
- They’re compiled!
- We control our software?
- Open source? Many eyes?
- We pentest?
- We patch?
- Static analysis?

The *real* vulnerability is how the flaw got in!
Struts2 Remote Command Execution

CVE-2010-1870 exploit to run arbitrary OS command:

http://example.org/struts2app/myaction?foo=%28%23context[%22xwork.MethodAccessor.denyMethodExecution%22]%3D+ [...],%20%23_memberAccess[%22allowStaticMethodAccess%22]%3D+ [...],%20@java.lang.Runtime@getRuntime%28%29.exec%28%27mkdir%27/tmp/PWND%29[...]%27meh%27[...]=true

1,121,000 vulnerable downloads
10,700 organizations
Spring Expression Language Injection

**CVE-2011-2730** exploit to steal data out of user’s session:

http://example.org/springapp/search?query=${requestScope}

**Result:** Your search for:

```
"javax.servlet.forward.request_uri=/ELI
 njection/eval.htm,javax.servlet.forward.
 servlet_path=/eval.htm,user.roles=[ADMIN,
 USER,ANONYMOUS]
display name [WebApplicationContext
 for namespace 'cashflowServlet'];
startup by
[uid=root];org.springframework.web.se
 rvlet.view.InternalResourceView.DISPA
 CHED_PATH=/var/opt/test/eval.jsp,..."
```

returned zero results.

- 2,700,000 vulnerable downloads
- 43,700 organizations
What About Malicious Code?

Are you sure you know what’s in that jar file?

Dan Geer Quote

“Be very careful where the code you use comes from”

@OWASP AppSec DC 2012

Are you sure you know what’s in that jar file?

The amount of custom code in an application hasn’t changed very much in the past 10 years.
Dependency Management
The Ripple Effect

Hundreds are still tainted!
Trapped!
What's In Central?

**All Libraries**
- No Known Vulnerabilities: 63%
- Contains Known Vulnerabilities: 37%

**Downloaded**
- No Known Vulnerabilities: 74%
- Contains Known Vulnerabilities: 26%
It’s Not a Popularity Contest

38% Contains Known Vulnerabilities
62% No Known Vulnerabilities

28% Contains Known Vulnerabilities
72% No Known Vulnerabilities

Not Popular  Popular
License Doesn’t Matter (I Think)
Are We Downloading Old Versions?

7 years old!
Actual Vulnerable Downloads

17,666,703 vulnerable downloads of GWT

29,800,000 vulnerable downloads total
What’s More Secure: Vulns or No Vulns?

Extrapolate 31 libraries to 680,000 vulns in Central (Typical vuln rates are much higher)

“The best indicator of a library’s future security is a culture that places value on security and clear evidence of broad and rigorous security analysis”
Global 500

Total 2,800,000 downloads with known vulns!

Global 100 financials had 567,000 vulnerable downloads

How Many of These 31 Libraries Do They Use?

- Global 500: 19.2%
- All the Rest: 8.5%
<table>
<thead>
<tr>
<th>Directory</th>
<th>Library</th>
<th>Current</th>
<th>All Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>activation-1.1.jar</td>
<td>1.1</td>
<td>1.0.2, 1.1, 1.1-rev-1, 1.1.1</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>ant-1.7.0.jar</td>
<td>1.7.0</td>
<td>1.7.0, 1.7.1, 1.8.0, 1.8.1, 1.8.2, 1.8.3</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>ant-launcher-1.7.0.jar</td>
<td>1.7.0</td>
<td>1.7.0, 1.7.1, 1.8.0, 1.8.1, 1.8.2, 1.8.3</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>antisamy-1.4.3.jar</td>
<td>1.4.3</td>
<td>1.4.2, 1.4.3, 1.4.4, 1.4.5</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>antlr-2.7.7.jar</td>
<td>2.7.7</td>
<td>2.7.1, 2.7.2, 2.7.4, 2.7.5, 2.7.6rc1, 2.7.</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>antlr-3.4.jar</td>
<td>3.4</td>
<td>3.1, 3.1.1, 3.1.2, 3.1.2-1, 3.1.3, 3.2, 3.2:</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>antlr-runtime-3.4.jar</td>
<td>3.4</td>
<td>3.1, 3.1.1, 3.1.2, 3.1.2-1, 3.1.3, 3.2, 3.2:</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>aopalliance-1.0.jar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>asm-3.3.1.jar</td>
<td>3.3.1</td>
<td>2.2.1, 2.2.2, 2.2.3, 3.0, 3.0_RC1, 3.1, 3.2, 3.3, 3.3.1, 20041228.180559</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>aspect-spring-esapi-ldap-1.0.0.jar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>aspectjrt-1.6.12.jar</td>
<td>1.6.12</td>
<td>1.6.3, 1.6.4, 1.6.5, 1.6.6, 1.6.7, 1.6.8, 1.6.9, 1.6.10, 1.6.11, 1.6.12</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>aspectjweaver-1.6.12.jar</td>
<td>1.6.12</td>
<td>1.6.3, 1.6.4, 1.6.5, 1.6.6, 1.6.7, 1.6.8, 1.6.9, 1.6.10, 1.6.11, 1.6.12</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>avro-1.5.1.jar</td>
<td>1.5.1</td>
<td>1.4.1, 1.5.0, 1.5.1, 1.5.2, 1.5.3, 1.5.4, 1.6.0, 1.6.1, 1.6.2, 1.6.3</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axiom-api-1.2.12.jar</td>
<td>1.2.12</td>
<td>1.2.3, 1.2.4, 1.2.5, 1.2.6, 1.2.7, 1.2.8, 1.2.9, 1.2.10, 1.2.11, 1.2.12</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axiom-dom-1.2.11.jar</td>
<td>1.2.11</td>
<td>1.2.3, 1.2.4, 1.2.5, 1.2.6, 1.2.7, 1.2.8, 1.2.9, 1.2.10, 1.2.11, 1.2.12</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axiom-impl-1.2.12.jar</td>
<td>1.2.12</td>
<td>1.2.3, 1.2.4, 1.2.5, 1.2.6, 1.2.7, 1.2.8, 1.2.9, 1.2.10, 1.2.11, 1.2.12</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axis-1.3-atlassian-1.jar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axis-jaxrpc-1.3.jar</td>
<td>1.3</td>
<td>1.2-alpha-1, 1.2-beta-2, 1.2-beta-3, 1.2RC1, 1.2RC2, 1.2RC3, 1.2, 1.2.1, 1.3, 1.4</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axis-saaj-1.3.jar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axis-wsdl4j-1.5.1.jar</td>
<td>1.5.1</td>
<td>1.2-beta-2, 1.2-beta-3, 1.2-RC1, 1.2-RC2, 1.2-RC3, 1.2, 1.2.1, 1.3, 1.5.1</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axis2-adb-1.5.5.jar</td>
<td>1.5.5</td>
<td>1.4.1, 1.5, 1.5.1, 1.5.2, 1.5.3, 1.5.4, 1.5.5, 1.5.6, 1.6.0, 1.6.1</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axis2-kernel-1.5.5.jar</td>
<td>1.5.5</td>
<td>1.4.1, 1.5, 1.5.1, 1.5.2, 1.5.3, 1.5.4, 1.5.5, 1.5.6, 1.6.0, 1.6.1</td>
</tr>
<tr>
<td>/target/swp/WEB-INF/lib</td>
<td>axis2-transport-http-1.5.5.jar</td>
<td>1.5.5</td>
<td>1.5, 1.5.1, 1.5.2, 1.5.3, 1.5.4, 1.5.5, 1.5.6, 1.6.0, 1.6.1</td>
</tr>
</tbody>
</table>
Using Maven

Add to your POM

```xml
<reporting>
  <plugins>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>versions-maven-plugin</artifactId>
      <version>1.3.1</version>
    </plugin>
  </plugins>
</reporting>
```

The following dependencies in Dependencies are using the newest version:

```xml
com.sun.jmx:jmxri .............................................. 1.2.1
commons-el:commons-el ............................................ 1.0
dbunit:dbunit ..................................................... 2.1
fitnesse:fitnesse .............................................. 20050731
jakarta:jakarta-oro ........................................... 2.0.8
javax.jms:jms ........................................................ 1.1
javax.sql.rowset ..................................................... 1.0.1
jdom:jdom ......................................................... 1.0
junit:junit ......................................................... 4.10
junit:junit-dep ..................................................... 4.10
oracle:ojdbc6-11g ............................................... 11.2.0.2.0
```

... 

The following dependencies in Dependencies have newer versions:

```xml
antlr:antlr ..................................................... 2.7.5 -> 20030911
cglib:cglib-nodep .................................................. 2.1 -> 2.2.2
com.hp.hpl.jena:jena ............................................. 2.3 -> 2.6.4
com.sun.xml.bind:jaxb-impl ..................................... 2.1.9 -> 2.2.4-1
commons-codec:commons-codec .................................... 1.3 -> 1.6
commons-collections:commons-collections ..................... 3.1 -> 3.2.1
commons-dbcp:commons-dbcp ...................................... 1.2.1 -> 1.4
jfree:jfreechart .................................................. 1.0.2 -> 1.0.12
log4j:log4j ......................................................... 1.2.14 -> 1.2.16
log4j:log4j ......................................................... 1.2.15 -> 1.2.16
org.apache.geronimo.specs:geronimo-servlet_2.4_spec .... 1.0.1 -> 1.1.1
```

...
Action Plan

Immediate: Inventory

• Scan for libraries
• Create tracking spreadsheet

Short Term: Analyze

• Purge unnecessary libraries
• Code review

Tactical: Control

• Centralize library control
• Consider Java sandbox

Monitor

• Manage your libraries
• Get security intelligence
Jeff Williams
Aspect Security CEO
jeff.williams@aspectsecurity.com
http://www.aspectsecurity.com