The TRUE STATE of OPEN SOURCE SECURITY

Based on the 2014 Sonatype Open Source Development Survey

DID YOU KNOW?

Applications are the **#1 attack vector** leading to breaches.[1]

90% of a typical application is assembled with open source components.[2]

6 in 10 don't track component vulnerabilities over time.

101010101

31% had or suspect a breach due to an open source (OSS) component.

77% have never banned an open source component.

PRACTICING INSECURE DEVELOPMENT

Last year, **44 million** vulnerable components were downloaded, such as...

CVE -2013-2251
Release Date: July 20, 2013
CVSS v2 Base Score: **9.3 HIGH**
Impact Subscore: **10.0**
Exploitability Subscore: **8.6**

STRUTS2

Since the alert was issued, **4,076 organizations** have downloaded it **179,050** times.

AS USAGE EXPLODES, PRACTICES LAG

63% have an incomplete view of license risk.

Only **21%** of organizations must prove they are using secure components.

75% don't have meaningful controls over what components are in their applications.

ARE POLICIES KEEPING US SAFER?

75% have a policy.

68% follow policies.

Top 3 challenges with policy:

No enforcement.
No security.
Not clear what's expected.

NEXT STEPS

1. **Understand your current component usage.**
Use a "bill of materials" to identify the suppliers in your "software supply chain." This report lists all components you use along with any known vulnerabilities.

2. **Design your Open Source Software (OSS) governance to be frictionless, scalable and automated.**
Policies must be agile enough to keep pace with modern development. Strive to automate policy enforcement and minimize drag on developers.

3. **Enable developer decision support.**
Provide information on component vulnerabilities (and licensing risk) within the IDE to make it easy for developers to pick the best components from the start.

4. **Continuously govern your risks throughout the software lifecycle.**
Since security isn't a point-in-time event, continuous monitoring should be used to alert you when you are about to use a vulnerable component and as new vulnerabilities are discovered in components you've already used.

For more information, visit www.sonatype.com/clm

Sonatype
www.sonatype.com
NEA
TSWA Application Security · Trusted Software Alliance
RUGGED
CONTRAST SECURITY

[1] 2014 Verizon Application Security Research
[2] Sonatype Inc. analysis of applications based on the Application Healthcheck
*2014 Sonatype Open Source Development Survey; 3,353 participants including developers, managers and architects involved in open source development