THE STATE OF OPEN SOURCE VULNERABILITIES MANAGEMENT

David Habusha
VP of Product, WhiteSource
SO... WHAT’S A VULNERABILITY?

A weakness of an asset that can be exploited by one or more threats.
Security advisories

Other advisories examples: securityfocus, nodesecurity, rubysec and many others...
Open Source Vulnerability Databases

Security Advisories
- RAILS
- SensioLabs

CVE/NVD
- CVE
- NVD

Issue/Bug Trackers
- JIRA
- Bugzilla
1. Reported open source security vulnerabilities are on the rise.

2. The absence of standard practices and developer-focused tools lead to inefficient handling of open source vulnerabilities.

3. Prioritization is crucial to ensure companies address the most critical vulnerabilities on time.

4. Prioritization based on usage analysis can reduce security alerts by 70% to 85%.
OPEN SOURCE SECURITY VULNERABILITIES ARE ON THE RISE
The number of disclosed open source vulnerabilities Rose by over 50% in 2017
96.8% of developers rely on open source components.
7.5% of all open source projects are vulnerable, but when it comes to the most popular open source projects...
But, it's not all bad.
The rise in awareness also led to a sharp rise in suggested fixes...

97.4%
of all reported vulnerabilities have at least one suggested fix in the open source community.
Information about vulnerabilities is scattered across hundreds of resources, usually poorly indexed and therefore unsearchable.

Over 86% of reported open source vulnerabilities appear in the CVE database.
DEVELOPERS ARE NOT EFFICIENTLY MANAGING OPEN SOURCE VULNERABILITIES
Developers rated security vulnerabilities as the #1 challenge when using open source components.
Developers spend 15 hours each month dealing with open source vulnerabilities (e.g. reviewing, discussing, addressing, remediating, etc.).

The cost is even higher, considering that the more experienced developers are the ones remediating.
Out of the monthly 15 hours only 3.8 hours are invested in remediation.

The lack of set practices and tools can explain these inefficiencies.
PRIORITIZATION IS KEY TO OPEN SOURCE VULNERABILITY MANAGEMENT
Perfect security is impossible. 
Zero risk is impossible.
We must bring prioritization of application vulnerabilities to DevSecOps. In a futile attempt to remove all possible vulnerabilities from applications, we are slowing developers down and wasting their time chasing issues that aren’t real.

10 Things to Get Right for Successful DevSecOps
Neil MacDonald, Gartner
Survey results show that developers prioritize remediation of vulnerabilities based on available information, not necessarily on the impact of a vulnerability on the security of an application.
A new approach to prioritizing vulnerabilities - based their impact on an application’s security.

EFFECTIVE VULNERABILITY
If the proprietary code is making calls to the vulnerable functionality

INEFFECTIVE VULNERABILITY
If the proprietary code is NOT making calls to the vulnerable functionality
After testing 2,000 Java applications, WhiteSource found that 72% of all detected vulnerabilities were deemed ineffective.

Based on the data collected in our survey, this can be translated to saving 10.5 hours per month per each developer (70% of 15 monthly hours).
EFFECTIVE USAGE ANALYSIS
Effective Usage Analysis is the technology of prioritizing open source vulnerabilities based on the way they are used by the application.

Our beta testing on 25 commercial applications from 12 organizations showed that:

- 15.8% of analyzed projects were found to be effective.
- 90% of the vulnerabilities (effective and ineffective) were found in transitive dependencies.
- 86% of all vulnerability alerts were found to be ineffective.
- 64% of all analyzed projects were found to contain only ineffective vulnerabilities.
ABOUT WhiteSource

WhiteSource's vision is to empower businesses to develop better software, by securing and managing the open source components in their software.

- Founded in 2011
- 500+ Customers
- Empowering over 1.2M developers
- Supporting 23% of Fortune 100 companies
- Over 300% growth YOY