VERACODE

Open Source Components

Vulnerability Information Sources & Vulnerability Likelihood

Chris Wysopal, Veracode Co-founder and CTO

July 19, 2018

ca technologies

Not all public vulnerabilities are in NVD

- Public information about open source vulnerabilities is available directly from open source projects
- Security bulletins, release notes, commit comments, and source code comments contain vulnerability information
- This information is readily available to attackers and defenders
- CA Veracode SourceClear crawls this information daily. Security analysts performs quality review, and compile a database with SVEs (SourceClear Vulnerability Enumeration).

Percentage of vulnerabilities not in NVD – 31%

Language	CVE	Reserved CVE	SVE	% SVE Low	% SVE High
JS	604	47	490	42.94%	44.79%
PHP	522	14	128	19.28%	19.69%
DOTNET	58	0	1	1.69%	1.69%
JAVA	749	60	335	29.28%	30.90%
RUBY	284	43	268	45.04%	48.55%
PYTHON	389	59	228	33.73%	36.95%
GO	90	5	218	69.65%	70.78%
CPP	193	8	12	5.63%	5.85%
OBJECTIVEC	631	14	9	1.38%	1.41%
CSHARP	33	3	0	0.00%	0.00%
	3553	253	1689	**30.74%**	**32.22%**

% SVE Low assumes reserved CVEs overlap with SVEs

% SVE High assumes reserved CVEs do not overlap with SVEs

Component Vulnerability Exploitability

- A product is vulnerable when it contains a vulnerable component **and** the product uses the library in such a way that the vulnerable code can be exercised.
- Control flow analysis was used determine if vulnerable code is reachable from the product code.
- Analysis was not performed to determine if vulnerable code can be called directly by attacker or called when attacker has exploited another vulnerability.

For Java, Ruby and Python, less than 5% of products that contain a library with a vulnerability are vulnerable

	repos analyzed	% component vulnerabilities that make the products vulnerable
Ruby	510	**3.50%**
Java	5232	**4.15%**
Python	585	**0.69%**

Conclusions

- NVD should not be considered the authoritative source of vulnerability information. Open source projects have significant amount of additional public vulnerability information.
- Assumption should be a product is **not** likely vulnerable when it includes a component with a vulnerability.

Thank you

Chris Wysopal
cwysopal@veracode.com
@weldpond