Identifying Open Source Insecurities Inside v1.0

Bill Jaeger, bjaeger@lenovo.com

October 26, 2018

About…

Bill Jaeger — Director, Security Architecture & DCG PSO

- Founding member Corporate & DCG Product Security Offices
- Work with global product teams, industry partners, and customers to drive product security enhancements – achieving a number of “firsts” for Lenovo
- 25+ years solving complex security, operational, and technical challenges for government and commercial enterprises

Lenovo — Data Center Group (DCG)

- Focused on Data Center Products: Servers, networking, storage, management, hyperconverged
- HQ’d in Morrisville, NC USA: ~5K staff across 50+ countries, ~1.4K in US
- Top 5 Global Server Manufacturer: Roots in Lenovo Server + IBM System x Divisions
- *Lots of firmware and software!*

Overview

- Open Source (In)Security: Is It Really a Problem?
- Software Composition Analysis Tools & Utilities
- Lenovo Data Center Group’s Approach
- What Can I Do?

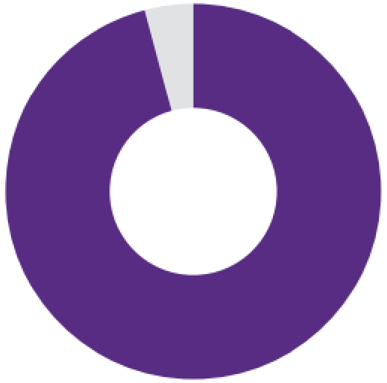
Open Source (In)Security

Is It Really a Problem?

Why Be Concerned?

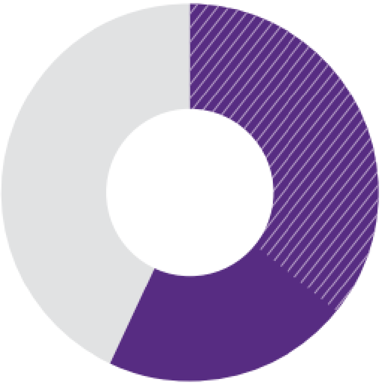
2018 Open Source Security and Risk Analysis, Synopsys Center for Open Source Research & Innovation
https://www.blackducksoftware.com/open-source-security-risk-analysis-2018

Prevalence



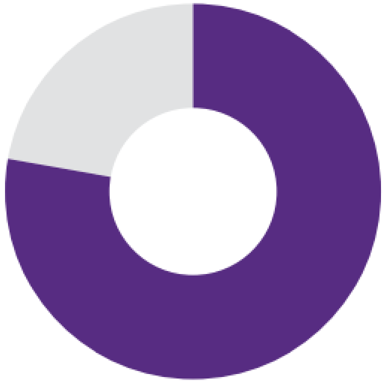
Black Duck On-Demand audits found open source components in **96%** of the applications scanned, with an average **257** components per application.

Proportion



The average percentage of codebase that was open source was **57%** vs. **36%** last year. Many applications now contain more open source than proprietary code.

Vulnerabilities



78% of the codebases examined contained at least one vulnerability, with an average **64** vulnerabilities per codebase.

*On average, vulnerabilities identified in the audits were disclosed nearly **6** years ago.*

Prevalence of [known vulnerable components] is very widespread…

Some of the largest breaches to date have relied on exploiting known vulnerabilities in components.

…perhaps this risk should be at the top of the list.

OWASP Top 10-2017
A9-Using Components with Known Vulnerabilities

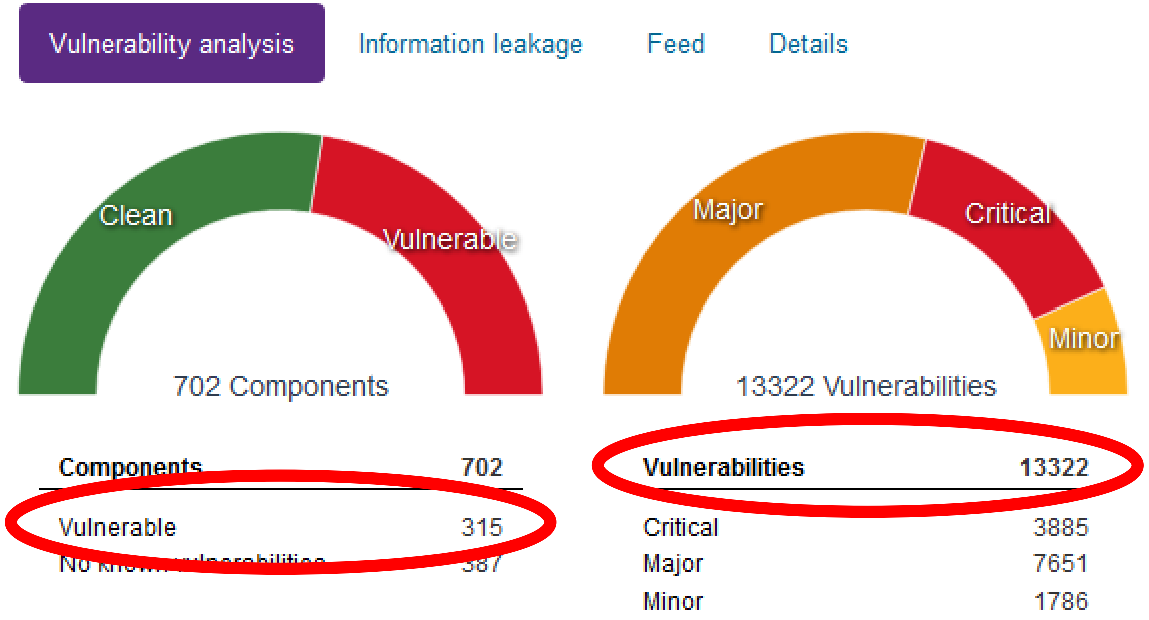
The Solution?

©2012 Ziggy and Friends, Inc./Dist. by Universal Uclick
...IT SHOULD BE OKAY NOW...I DOWNLOADED A PATCH!
COMPUTER REPAIR
Tom Wilson
4/17

https://www.gocomics.com/ziggy/2012/04/17

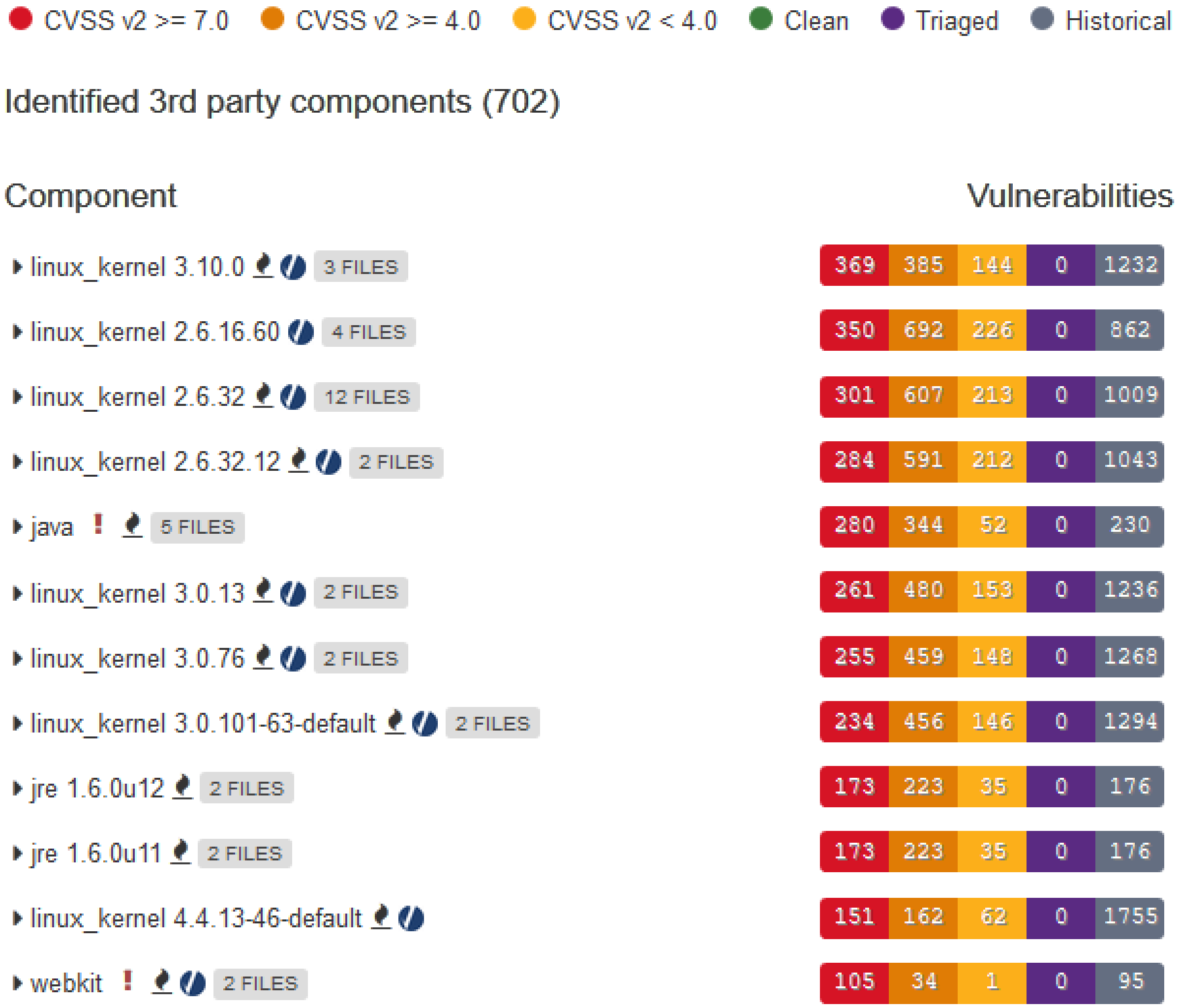
Ongoing Open Source Hygiene is Essential

Insecurities Accrue with Technical Debt



Components	702
Vulnerable	315
No known vulnerabilities	387

Vulnerabilities	13322
Critical	3885
Major	7651
Minor	1786



Identified 3rd party components (702)

Component		Vulnerabilities				
linux_kernel 3.10.0	3 FILES	369	385	144	0	1232
linux_kernel 2.6.16.60	4 FILES	350	692	226	0	862
linux_kernel 2.6.32	12 FILES	301	607	213	0	1009
linux_kernel 2.6.32.12	2 FILES	284	591	212	0	1043
java !	5 FILES	280	344	52	0	230
linux_kernel 3.0.13	2 FILES	261	480	153	0	1236
linux_kernel 3.0.76	2 FILES	255	459	148	0	1268
linux_kernel 3.0.101-63-default	2 FILES	234	456	146	0	1294
jre 1.6.0u12	2 FILES	173	223	35	0	176
jre 1.6.0u11	2 FILES	173	223	35	0	176
linux_kernel 4.4.13-46-default		151	162	62	0	1755
webkit !	2 FILES	105	34	1	0	95

315 Vulnerable Components
13,322 Vulnerabilities

CVEs dating back 10+ years!

> Software doesn't age like wine.
>
> It ages like milk.

Chris Eng
VP of Research
Veracode

How Did We Get Here?

Development Process

- Developers adopt Open Source to speed development, solve a problem, or play with a shiny new technology
- Legal performs an initial license review and approves
- Developers develop, testers test
- Code works, is stable, and ships
- Repeat…

Reasons for Poor Code Hygiene

- Developers don't know that code has vulnerabilities
- Developers don't appreciate that vulnerabilities can be exploited
- The code works and is stable
- "We'll update if someone complains"
- "Open Source is defect-free"
- "It's been sooooo long since the last update that updating now is too difficult"

Software Composition Analysis Tools & Utilities

Divination Through Software Composition Analysis

(noun) The practice of seeking knowledge of the unknown by supernatural means

Benefits

- Provide visibility into otherwise opaque software
- Generate invaluable insights into software characteristics and development practices
- Component inventory generation as business enabler
- Proactive vulnerability identification and notification
- Far faster than manual review

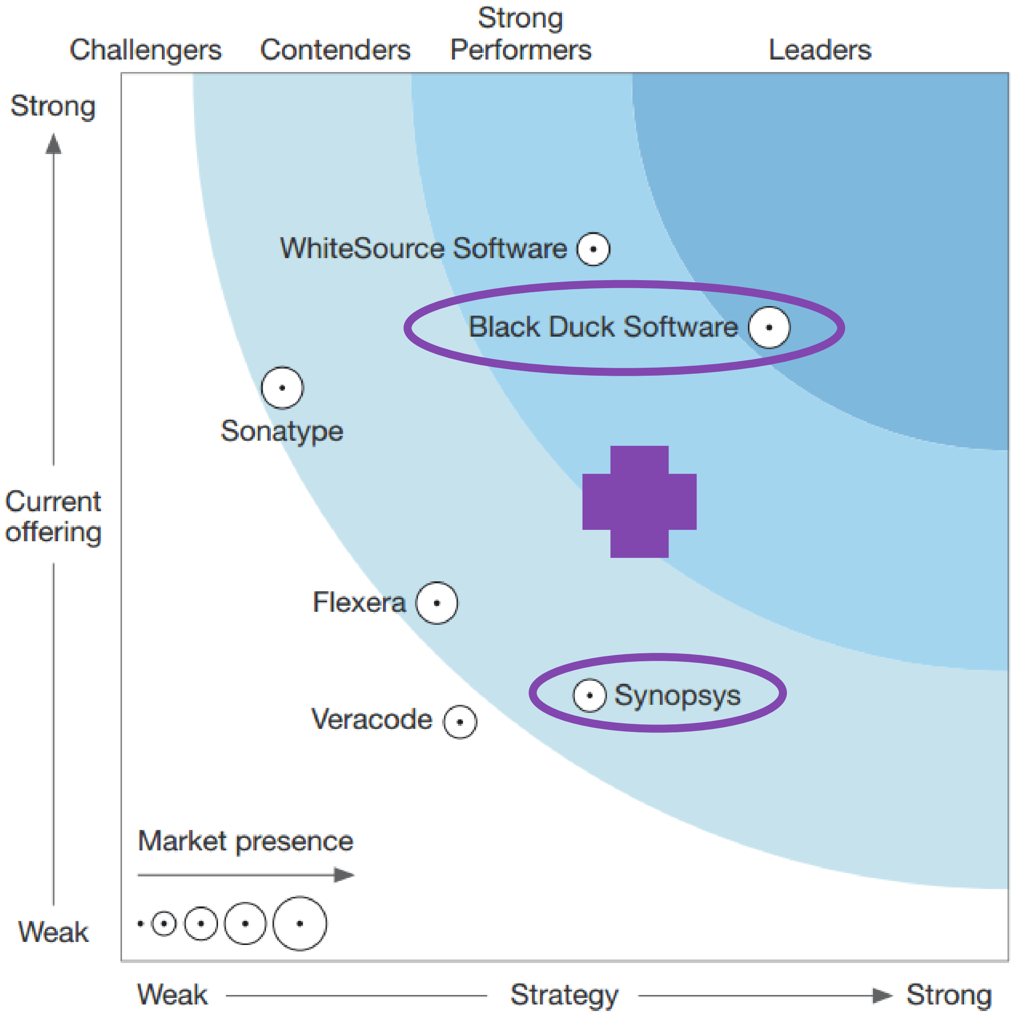
Challenges

- Imperfect identification of components, versions
- Decomposition limitations
- False positives
- Information overload
- Vulnerability overload
- Manual effort required to work through quirks, bridge gaps
- Slower than desired

Commercial Software Composition Analysis Tools

The Forrester Wave™: Software Composition Analysis, Q1 2017 – Top 6 Providers
https://www.blackducksoftware.com/sites/default/files/images/Downloads/Reports/USA/ForresterWave-Rpt.pdf

- Most SCA products analyze source code
- Some analyze easily decompiled binaries (e.g., Java, .NET)



- Synopsys' Black Duck Binary Analysis* analyzes binaries
- "Binary X-ray" capability provides insights into opaque code received from the supply chain

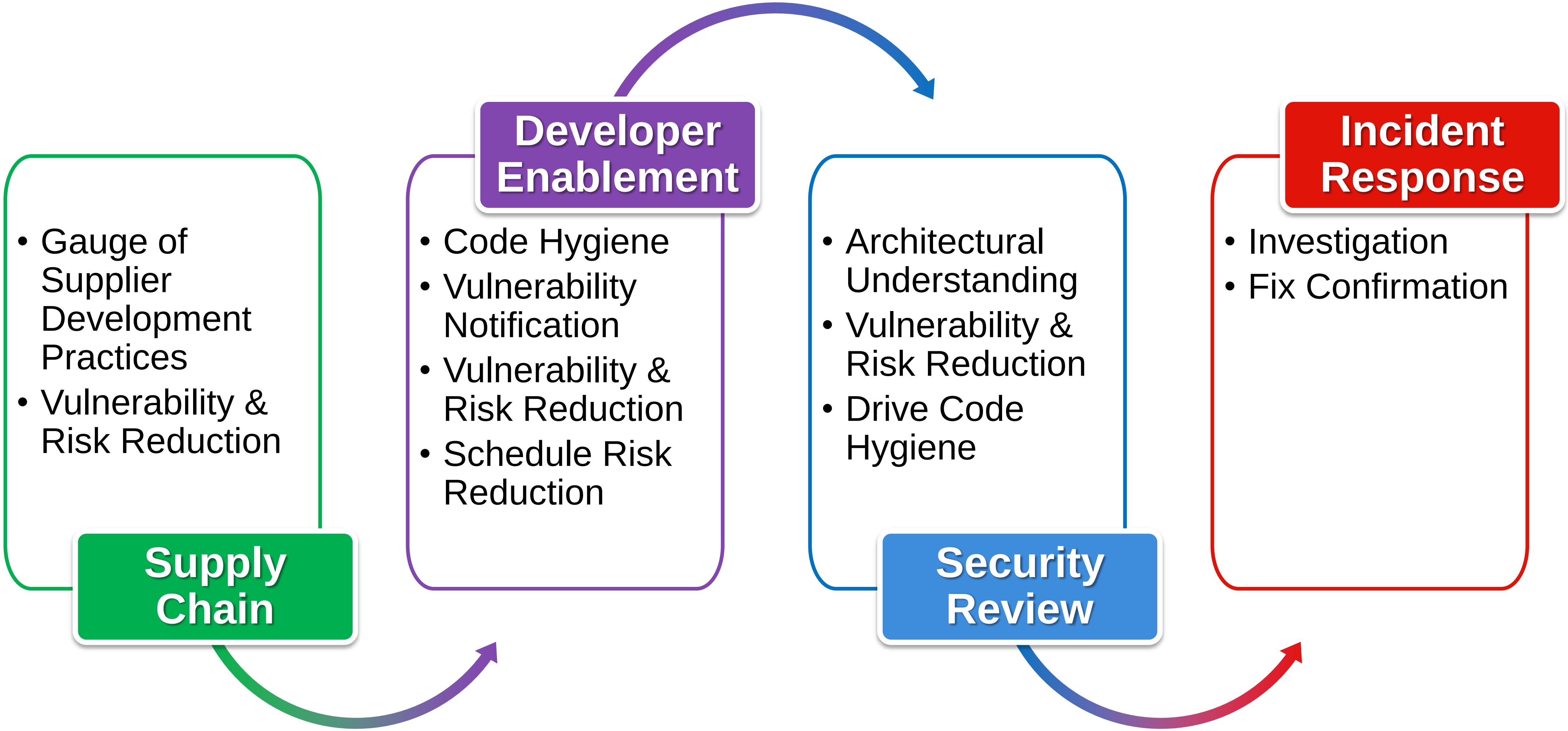
* = formerly Protecode SC

Free Software Composition Analysis Tools & Utilities

- **OWASP dependency-track**
  - SCA platform that identifies and helps reduce risk from the use of third-party and open source components
  - https://dependencytrack.org
- **OWASP DependencyCheck**
  - SCA utility that detects publicly disclosed vulnerabilities in application dependencies
  - https://github.com/jeremylong/DependencyCheck
- **retire.js**
  - Scan web and node applications for known vulnerable JavaScript libraries and/or node modules
  - http://retirejs.github.io/retire.js
- **7-Zip**
  - File archiver/extractor supporting many compressed file formats
  - https://www.7-zip.org
- **Binwalk**
  - Tool for extracting and analyzing firmware
  - https://github.com/ReFirmLabs/binwalk
- **Coverity Scan**
  - Browse open source project activity and static code analysis defects
  - https://scan.coverity.com/projects
- **CVEDetails**
  - Browse CVE details and statistics for vendors, products, and versions
  - https://www.cvedetails.com

Lenovo Data Center Group’s Approach

Software Composition Analysis Integral to Process



Supply Chain

- Gauge of Supplier Development Practices
- Vulnerability & Risk Reduction

Developer Enablement

- Code Hygiene
- Vulnerability Notification
- Vulnerability & Risk Reduction
- Schedule Risk Reduction

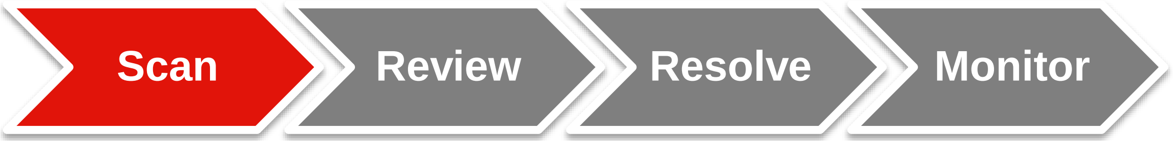
Security Review

- Architectural Understanding
- Vulnerability & Risk Reduction
- Drive Code Hygiene

Incident Response

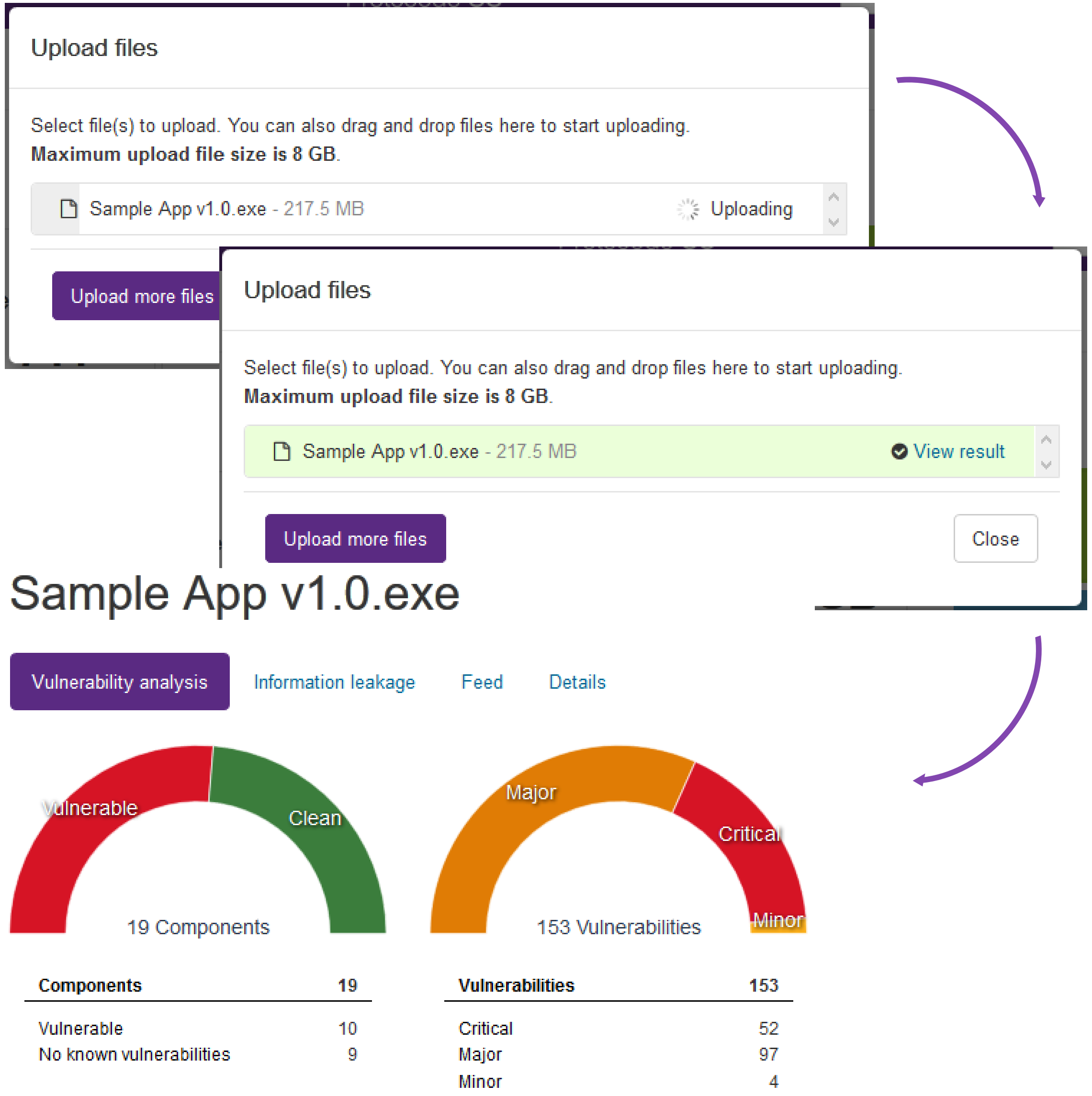
- Investigation
- Fix Confirmation

Software Composition Analysis in Action: Scan



Scan → Review → Resolve → Monitor

- **Upload Binary** to be analyzed
  - Via Web UI or REST API
- **Automatic Analysis** happens upon upload completion



Upload files

Select file(s) to upload. You can also drag and drop files here to start uploading.
Maximum upload file size is 8 GB.

Sample App v1.0.exe - 217.5 MB Uploading

Upload files

Select file(s) to upload. You can also drag and drop files here to start uploading.
Maximum upload file size is 8 GB.

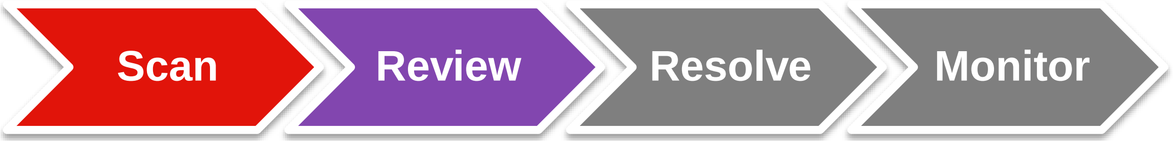
Sample App v1.0.exe - 217.5 MB ✓ View result

Sample App v1.0.exe

Components	19
Vulnerable	10
No known vulnerabilities	9

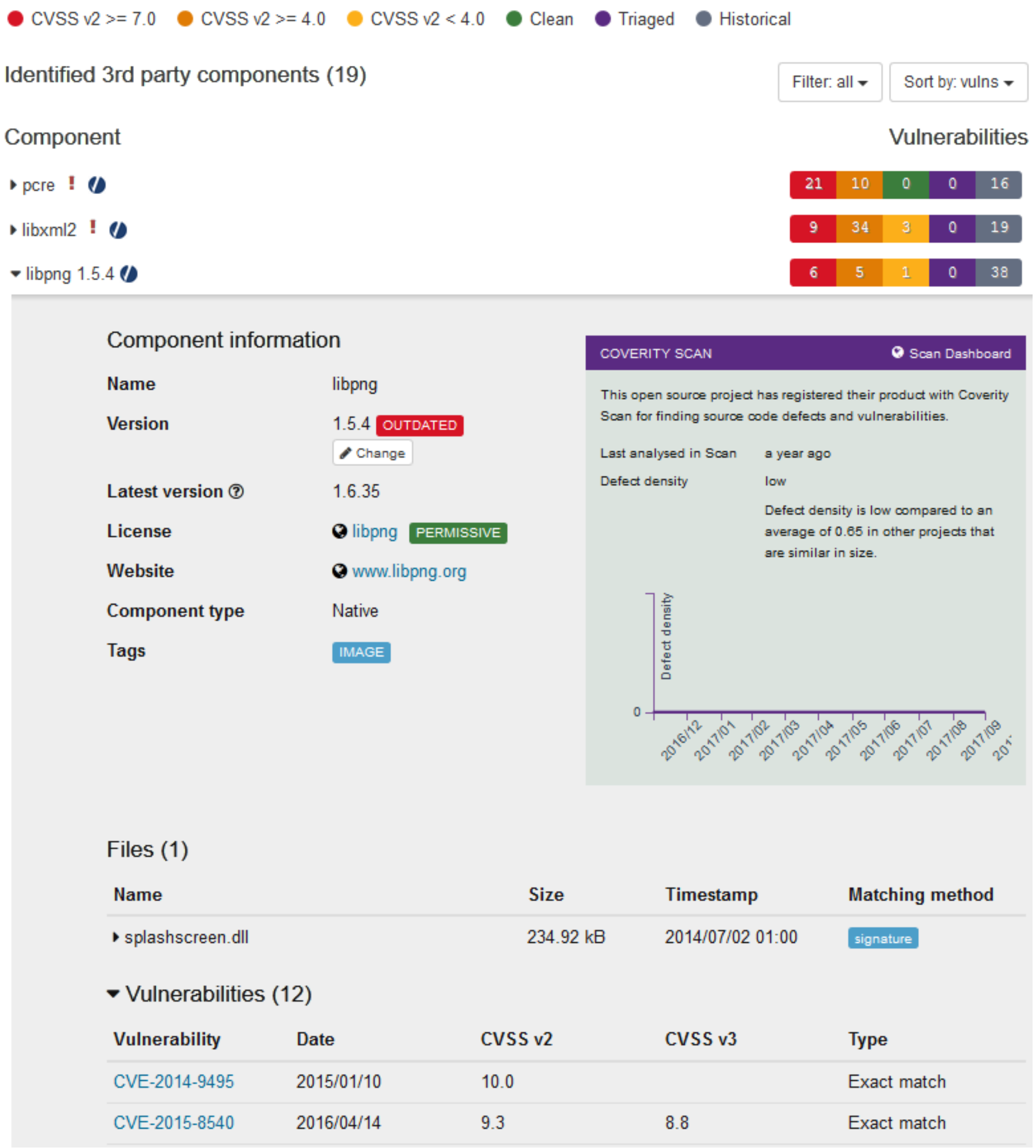
Vulnerabilities	153
Critical	52
Major	97
Minor	4

Software Composition Analysis in Action: Review



- **Sanity Check Results**
  - Verify valid processing; manually pre-process and re-scan, if needed
  - Assess for imperfect identification, tuning scan results for accuracy as applicable

- **Analyze Results**
  - Start with focus on vulnerable components, not individual vulnerabilities
  - Dive into individual vulnerabilities only to the extent necessary
  - Manual reconciliation may be needed for patched code, statically compiled code, or where version information is missing



● CVSS v2 >= 7.0 ● CVSS v2 >= 4.0 ● CVSS v2 < 4.0 ● Clean ● Triaged ● Historical

Identified 3rd party components (19)

Component	Vulnerabilities				
pcre !	21	10	0	0	16
libxml2 !	9	34	3	0	19
libpng 1.5.4	6	5	1	0	38

Component information

Name	libpng
Version	1.5.4 OUTDATED Change
Latest version	1.6.35
License	libpng PERMISSIVE
Website	www.libpng.org
Component type	Native
Tags	IMAGE

COVERITY SCAN — Scan Dashboard

This open source project has registered their product with Coverity Scan for finding source code defects and vulnerabilities.

Last analysed in Scan	a year ago
Defect density	low
	Defect density is low compared to an average of 0.65 in other projects that are similar in size.

Files (1)

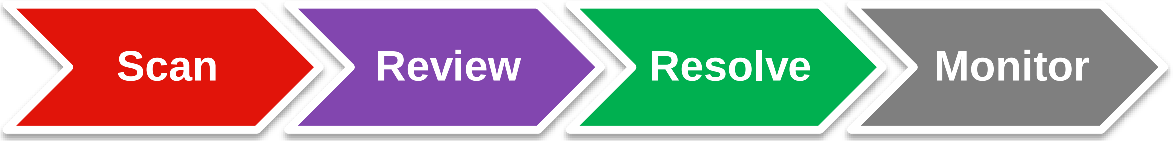
Name	Size	Timestamp	Matching method
splashscreen.dll	234.92 kB	2014/07/02 01:00	signature

Vulnerabilities (12)

Vulnerability	Date	CVSS v2	CVSS v3	Type
CVE-2014-9495	2015/01/10	10.0		Exact match
CVE-2015-8540	2016/04/14	9.3	8.8	Exact match

Lenovo 2018 Lenovo. All rights reserved.

Software Composition Analysis in Action: Resolve


Scan

Review

Resolve

Monitor

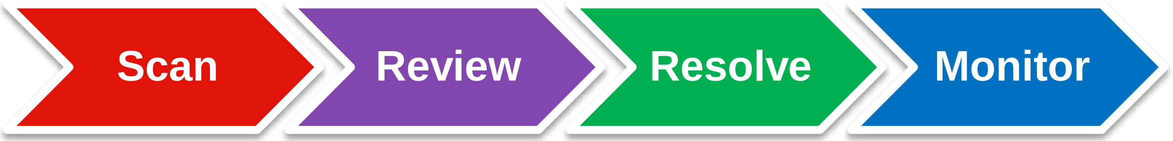
- **Remove** components if not used
  - Eliminate legacy baggage to reduce attack surface
  - Why maintain what isn’t needed?

- **Upgrade** components
  - The latest LTS release is preferred

- **Patch** components where upgrade is not feasible or available

- **Mitigate** where upgrade or patch is not available or feasible

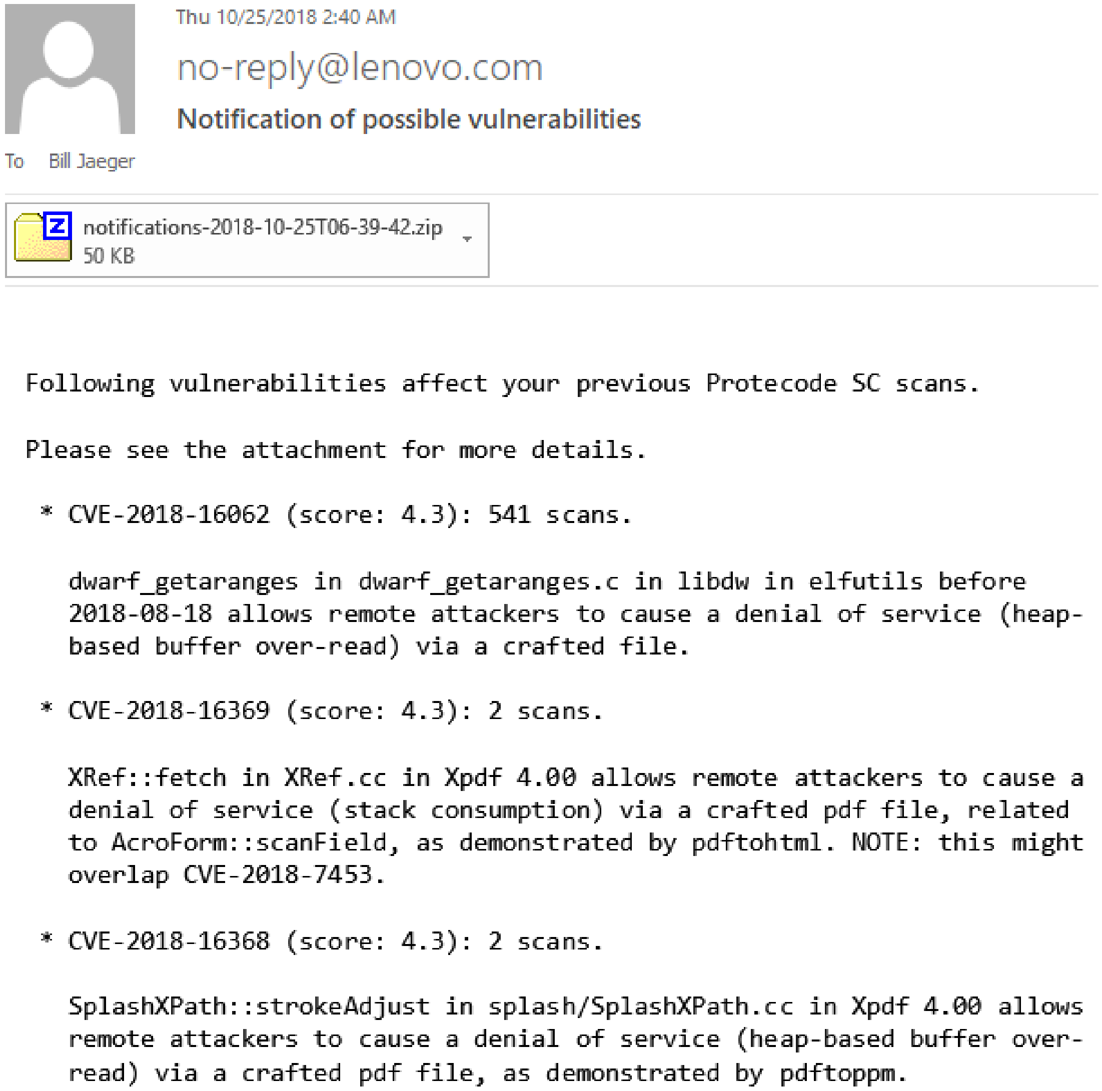
Software Composition Analysis in Action: Monitor



- **New Vulnerabilities** discovered daily
  - E-mail alerts
  - REST API for notifications
  - Manual dashboard review
  - Periodic scanning

- **Use Public Resources**
  - Component mailing lists and forums
  - CVEDetails (https://www.cvedetails.com)

- **Leverage Threat Intelligence** service or other commercial providers your organization may subscribe to


Thu 10/25/2018 2:40 AM

no-reply@lenovo.com

Notification of possible vulnerabilities

To Bill Jaeger

notifications-2018-10-25T06-39-42.zip
50 KB

Following vulnerabilities affect your previous Protecode SC scans.

Please see the attachment for more details.

* CVE-2018-16062 (score: 4.3): 541 scans.

  dwarf_getaranges in dwarf_getaranges.c in libdw in elfutils before 2018-08-18 allows remote attackers to cause a denial of service (heap-based buffer over-read) via a crafted file.

* CVE-2018-16369 (score: 4.3): 2 scans.

  XRef::fetch in XRef.cc in Xpdf 4.00 allows remote attackers to cause a denial of service (stack consumption) via a crafted pdf file, related to AcroForm::scanField, as demonstrated by pdftohtml. NOTE: this might overlap CVE-2018-7453.

* CVE-2018-16368 (score: 4.3): 2 scans.

  SplashXPath::strokeAdjust in splash/SplashXPath.cc in Xpdf 4.00 allows remote attackers to cause a denial of service (heap-based buffer over-read) via a crafted pdf file, as demonstrated by pdftoppm.

Tips & Tricks

Good Hygiene Indicators

- No / few vulnerabilities
- Vulnerabilities published post-release
- No / few duplicate components

Install and Re-Package What Won’t Scan

- Some installers are encrypted
- Deep-nesting of archives sometimes problematic

Encourage Development to Know Their Code

- Account for what’s been patched
- Articulate patch and mitigation strategies

Re-package Live Systems

- Some installers include the kitchen sink, which can distract from analyzing what is installed
- Some installers are stubs that download installed code

What Can I Do?

What Can I Do to Reduce Open Source Insecurities?

As a Buyer

- Require secure software from suppliers
- Proactively monitor for vulnerabilities
- Hold suppliers accountable for vulnerability fixes

As a Developer

- Use only active / supported projects
- Opt for long-term support releases
- Review project security track record
- Proactively resolve vulnerabilities
- Hygiene – embrace continuous updates

As a Security Team

- Drive adoption of Software Composition Analysis tools and techniques
- Proactively monitor for vulnerabilities
- Provide governance and guidance to Buyers and Developers

thanks.

Different is better

Lenovo