HOW TO MANAGE THE

tangled web

of

dependencies

31st annual FIRST conference
Hello!

Lisa Bradley, PhD
NVIDIA Senior Manager - PSIRT
LBradley @ nvidia.com

Jessica Butler
NVIDIA Security Tools Development Lead
Jessicab @ nvidia.com
April 7, 2014
CVE-2014-0160
The day

The world realized they had Open Source in their software

Companies realized they had no clue what Open Source they were using

Product teams realized they had to update their Open Source
FIGURING OUT WHAT YOU USE

OSS APPROVAL TICKET

LICENSE SCANNING

PSIRT SPREADSHEETS
Let’s get the right stuff in the first place!!

Where to begin?
Selecting the right open source

Having the right security practices for internal components

Choosing the right vendors
What else do you need?

- Project Development Model
- Quality assurance
- Documentation
- Use open standards
- Security evaluations
- Reviewers
- Cryptographic signatures
- Way to report security issues
- Maintainability
- Stability
- Active support
- Community support
- Reputation
What are we using already?

Do we actually know?
We thought we could use...

Content from our OSS approvals
Certificate of Originality
OSS licensing scanning tool results

The reality...

Emails and requests in bugs!
Non-consistent formats!
No mapping to product lines!
False positives!
Save me from the spreadsheet nightmare!
open source scanning
VULNERABILITY DETECTION
MANUAL PROCESS

1. REQUEST RECEIVED
2. GATHER SOURCE LOCATION
3. RUN SCANNING TOOL
4. MANUALLY GROUP DATA
5. SHIP REPORT TO REQUESTER
whatever you're thinking bigger
DevOps teams are 90% more likely to comply with open source governance when policies are automated.
~Sonatype’s 2018 State of the Software Supply Chain

Did I forget to say I also want...

Everything needs to be automated!!

All parts plug and play :)

15
Big thing

Break it down

1
2
3

Schedule time

Track progress
Step 1: Defining the PORTFOLIO

PRODUCTS
- Top level
- Shippable or deployable
- Executive ownership
- Versioning and EOL

COMPONENTS
- Logical segregation of product
- 1:n source code projects
- n:n products
- Build level ownership
- 1:n developer teams

DEPENDENCIES
- Internal components
- External open source software
- External third-party software
- Nestable

OPEN SOURCE SOFTWARE
- Versioning detection
- Vulnerability mapping
- Fix recommendations
- Fix verification
Step 2: **REGISTRATION**

Automating component mapping via build tool synchronization

1. PRODUCT INFO
2. 
3. 
4. 
5. 
6.
Step 2: Registration
Automating component mapping via build tool synchronization

ADD/IMPORT CONTACTS

PRODUCT INFO
Step 2: REGISTRATION

Automating component mapping via build tool synchronization

1. PRODUCT INFO
2. ADD/IMPORT CONTACTS
3. SYNC BUILD TOOLS
Step 2: **REGISTRATION**

Automating component mapping via build tool synchronization

**Components and Source Code Repositories**

Great! You're almost done, just verify the results and add anything that we weren't able to pull.

```
shell
#659e8c-3ac5-11db-a350-4f1e1a142bafe0
Static Analysis Info
grid-devops-jenkins-DGX-Cloud-WebService-Deploy
271ac49b-6b84-4377-a1c6-8b0b3d1d06e
Static Analysis Info
```

4. **VERIFY COMPONENTS**

5. **DEPENDENCIES**

6. **CHECKLIST**
Step 2: REGISTRATION

Automating component mapping via build tool synchronization

1. PRODUCT INFO
2. ADD/IMPORT CONTACTS
3. SYNC BUILD TOOLS
4. VERIFY COMPONENTS
5. LINK SERVICES
6. Complete
Step 2: REGISTRATION
Automating component mapping via build tool synchronization

1. PRODUCT INFO
2. ADD/IMPORT CONTACTS
3. SYNC BUILD TOOLS
4. VERIFY COMPONENTS
5. LINK SERVICES
6. REPORT ENABLED
Step 3: DEPENDENCY DEVOPS

Open Source screening
Step 3:

DEPENDENCY DEVOPS

Shift Left

- CODE
- BUILD
- PKG
- RELEASE
- CONFIG
- MONITOR

Open Source screening
Commit triggered scanning
Step 3: DEPENDENCY DEVOPS Shift Left

- CODE: Open Source screening
- BUILD: Commit triggered scanning
- PKG: Build defined components
- RELEASE
- CONFIG
- MONITOR
Step 3: DEPENDENCY DEVOPS Shift Left

- CODE: Open Source screening
- BUILD: Commit triggered scanning
- PKG: Build defined components
- RELEASE: Release tracking
- CONFIG
- MONITOR
Step 3: DEPENDENCY DEVOPS

- CODE: Open Source screening
- BUILD: Commit triggered scanning
- PKG: Build defined components
- RELEASE: Release tracking
- CONFIG: Platform dependencies
- MONITOR: Shift Left
Step 3:

DEPENDENCY DEVOPS

Shift Left

CODE
BUILD
PKG
RELEASE
CONFIG
MONITOR

Open Source screening
Commit triggered scanning
Build defined components
Release tracking
Platform dependencies
Continuous tracking and alerting
Step 4: Integrate for more data!

- Registration Service (Angular 7)
- Inventory DB (MySQL)
- Reporting Service (Kibana)
- Inventory Service (Flask, Celery)
- Team Build Pipeline (Jenkins)
- Internal security tool(s)
- Static analysis
- OSS scanning
Step 5: Metrics Drive Change

Severity Distribution for Vulnerabilities

Top 10 Repositories per Vulnerability Count
Managing YOUR WEB

**Portfolio**
Find the quickest way to populate and standardize.

**Scope**
Determine what is important and who owns it!

**Automation**
Integrate and automate. Meet developers where they are already working.

**Value**
Look for more interesting data. Entice your users with undeniable value. ;)

**Leverage**
Uncover the levers that control how your organization behaves with metrics!