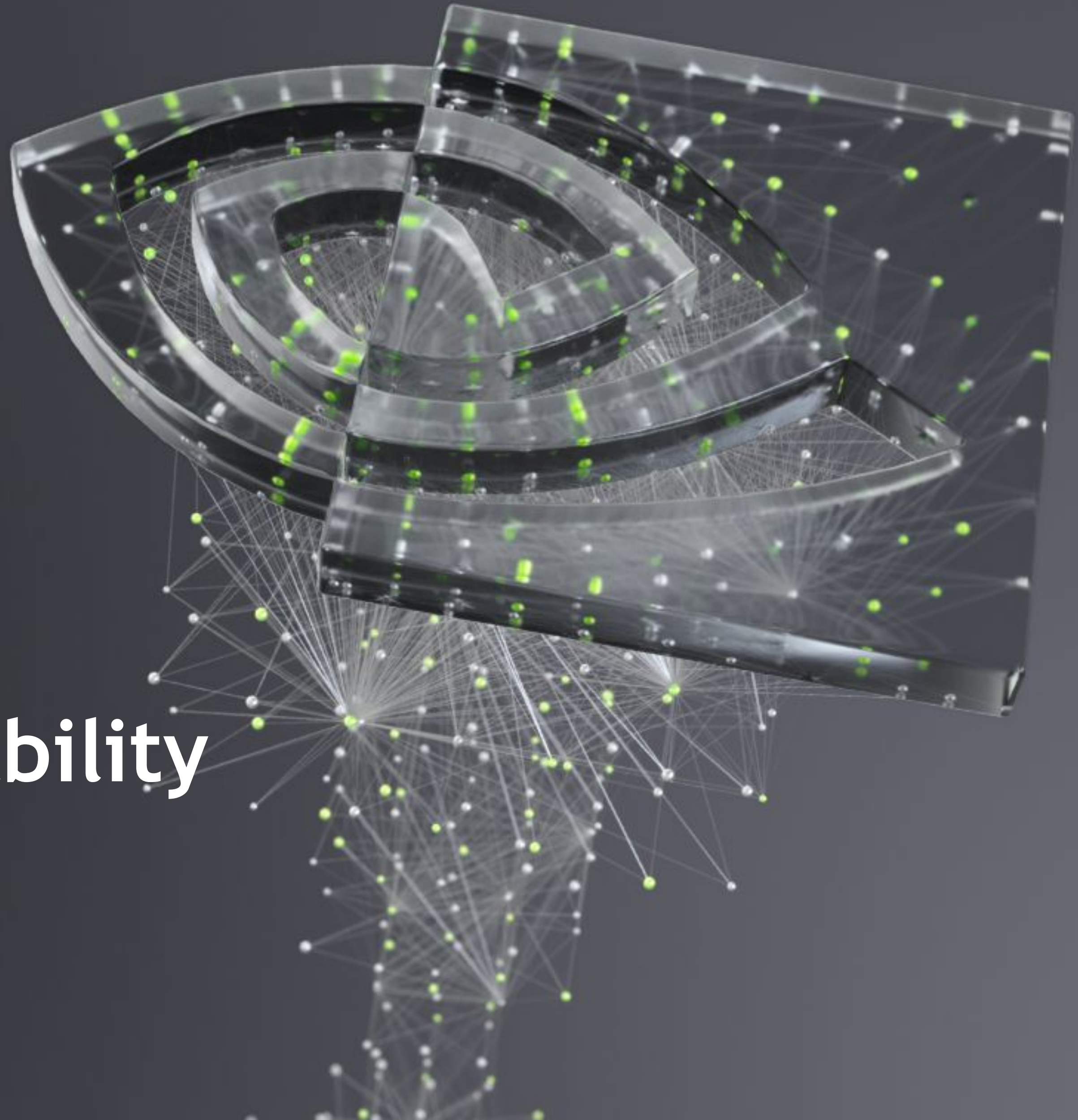


Automating Vulnerability Mapping from Tools

FIRST - PSIRT Technical Colloquium 2020 - March 4-5

About Us

NVIDIA Product Security Tools

Dee Annachhatre

NVIDIA Product Security

Tools Development

dannachhatre@nvidia.com

Jessica Butler

NVIDIA Product Security

Tools Development

jessicab@nvidia.com

Intro

Moving the Ball Forward

Pitfalls of manual process

All the Data - Oh My!

Cataloging Portfolio

Self-Service Registration tool

Mapping the Data - Oh Yeah!

Notifications

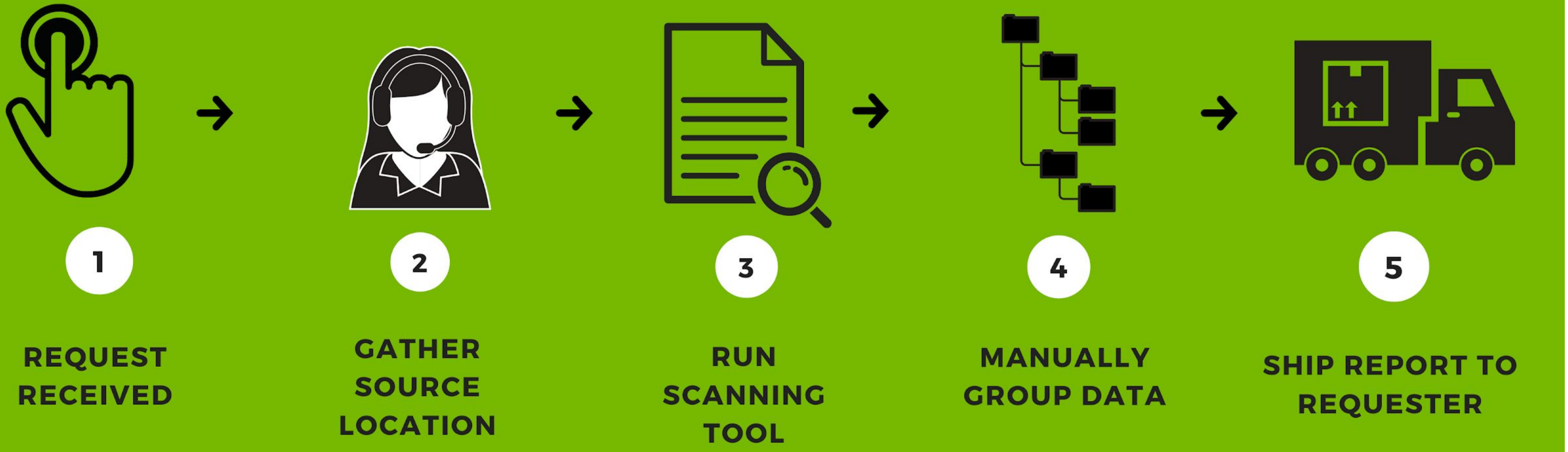
Issue Management - Yes, Please!

Calculating Risk

open source scanning

VULNERABILITY DETECTION

MANUAL PROCESS



1 REQUEST RECEIVED →

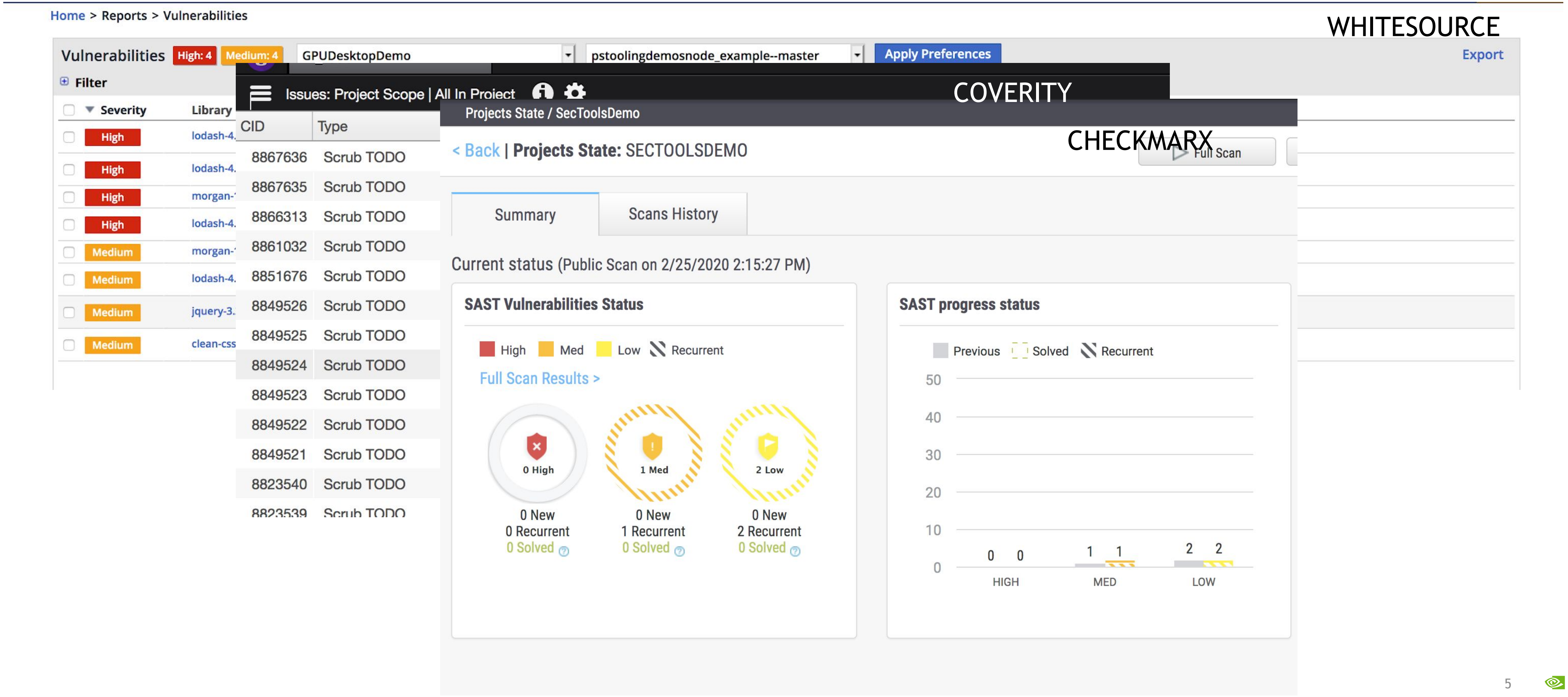
2 GATHER SOURCE LOCATION →

3 RUN SCANNING TOOL →

4 MANUALLY GROUP DATA →

5 SHIP REPORT TO REQUESTER

All the Data - OH MY!



WHITESOURCE

Home > Reports > Vulnerabilities

Vulnerabilities High: 4 Medium: 4 GPUDesktopDemo pstoolingdemosnode_example--master Apply Preferences Export

⊞ Filter

□	Severity	Library
□	High	lodash-4.
□	High	lodash-4.
□	High	morgan-
□	High	lodash-4.
□	Medium	morgan-
□	Medium	lodash-4.
□	Medium	jquery-3.
□	Medium	clean-css

COVERITY

Issues: Project Scope | All In Project

CID	Type
8867636	Scrub TODO
8867635	Scrub TODO
8866313	Scrub TODO
8861032	Scrub TODO
8851676	Scrub TODO
8849526	Scrub TODO
8849525	Scrub TODO
8849524	Scrub TODO
8849523	Scrub TODO
8849522	Scrub TODO
8849521	Scrub TODO
8823540	Scrub TODO
8823539	Scrub TODO

CHECKMARX

Projects State / SecToolsDemo

< Back | **Projects State:** SECTOOLSDEMO

Full Scan

Summary | Scans History

Current status (Public Scan on 2/25/2020 2:15:27 PM)

SAST Vulnerabilities Status

🟥 High 🟧 Med 🟨 Low ▨ Recurrent

Full Scan Results >

0 High	1 Med	2 Low
0 New	0 New	0 New
0 Recurrent	1 Recurrent	2 Recurrent
0 Solved	0 Solved	0 Solved

SAST progress status

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

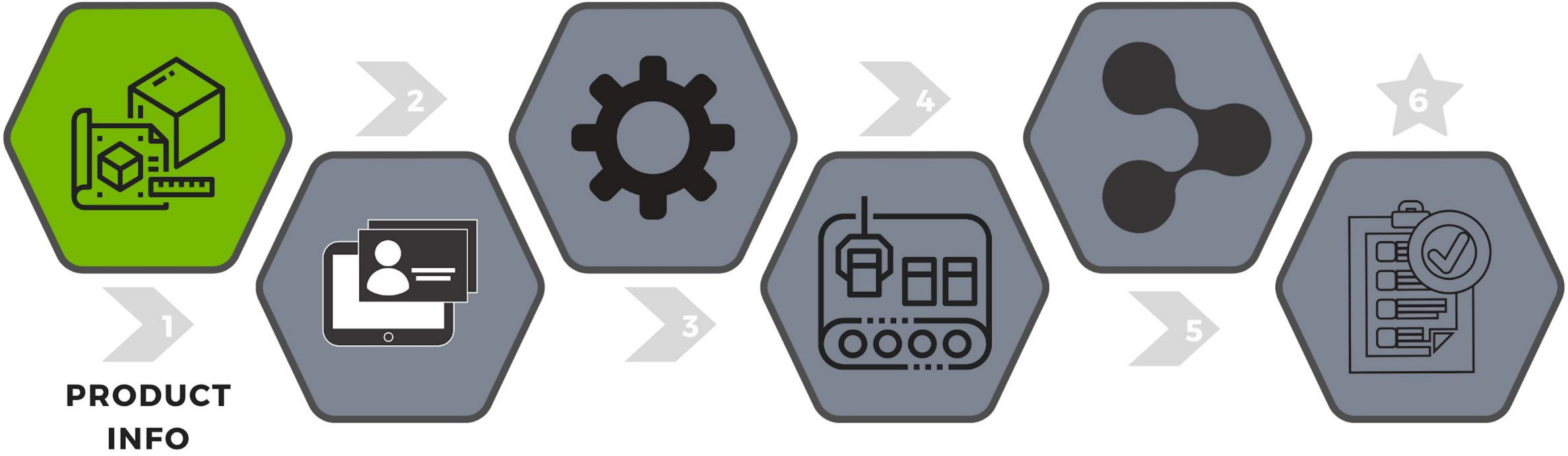
OPEN SOURCE SOFTWARE

- Versioning detection
- Vulnerability mapping
- Fix recommendations
- Fix verification

Self Service

REGISTRATION

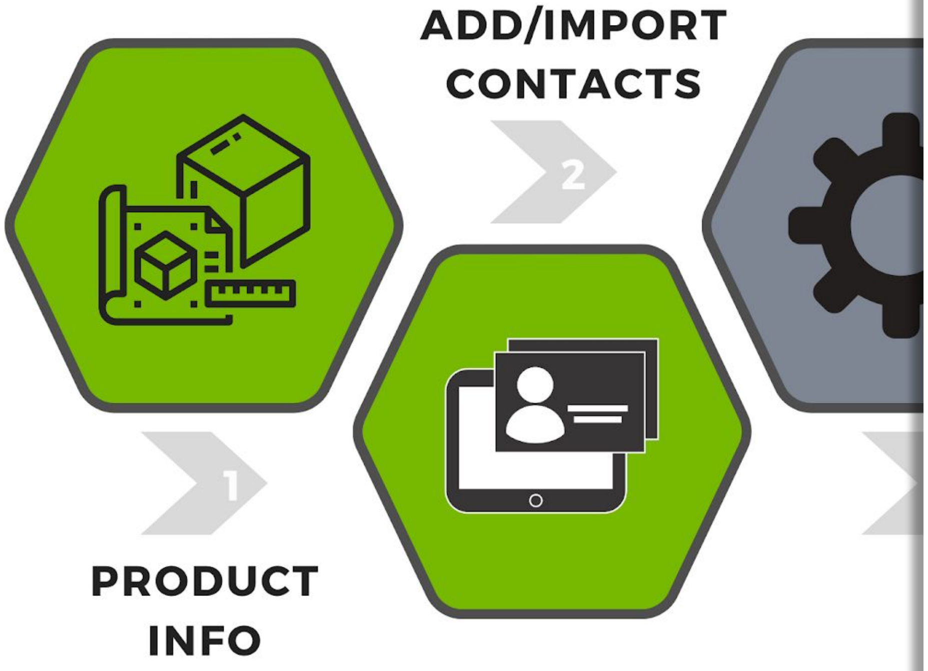
Automating component mapping via build tool synchronization

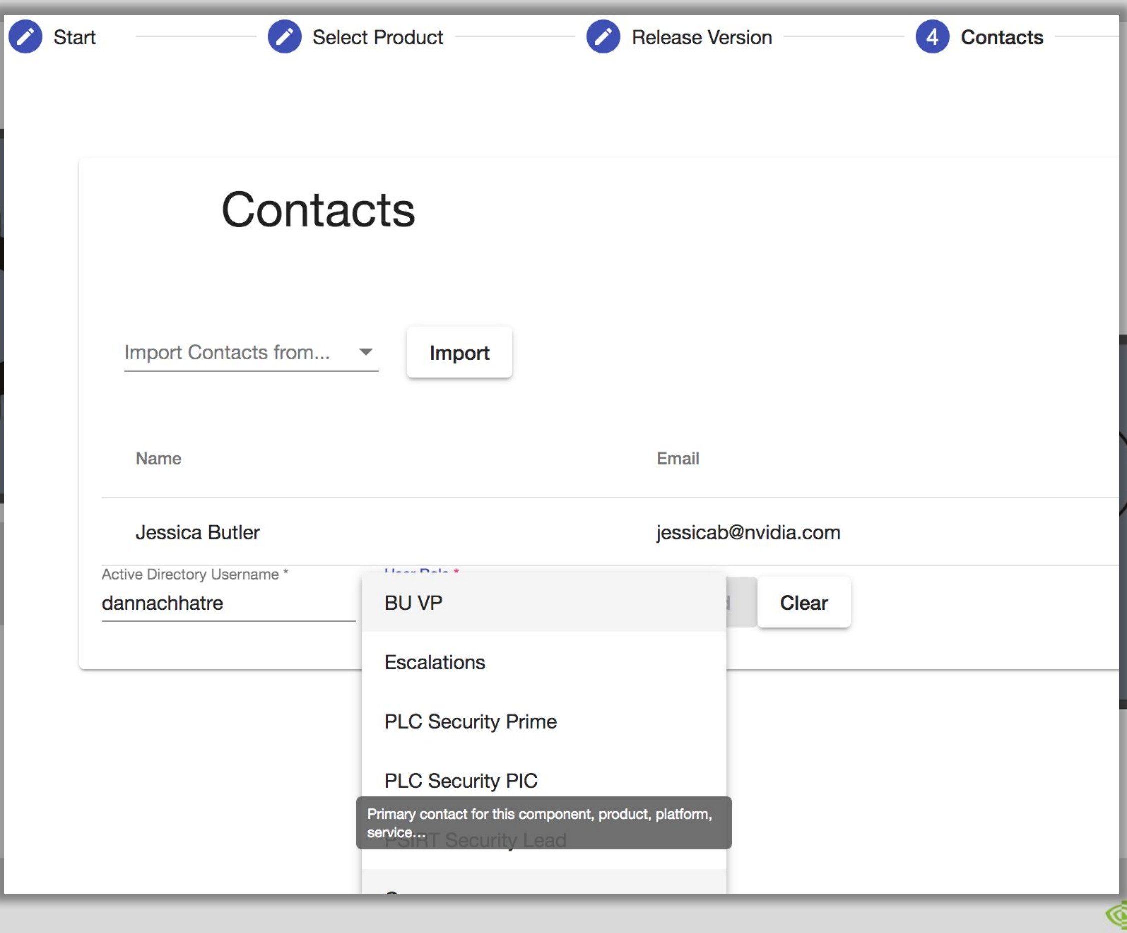


Self Service

REGISTRATION

Automating component mapping via build tool synchronization





Contacts

Name	Email
Jessica Butler	jessicab@nvidia.com

Active Directory Username *: dannachhatre

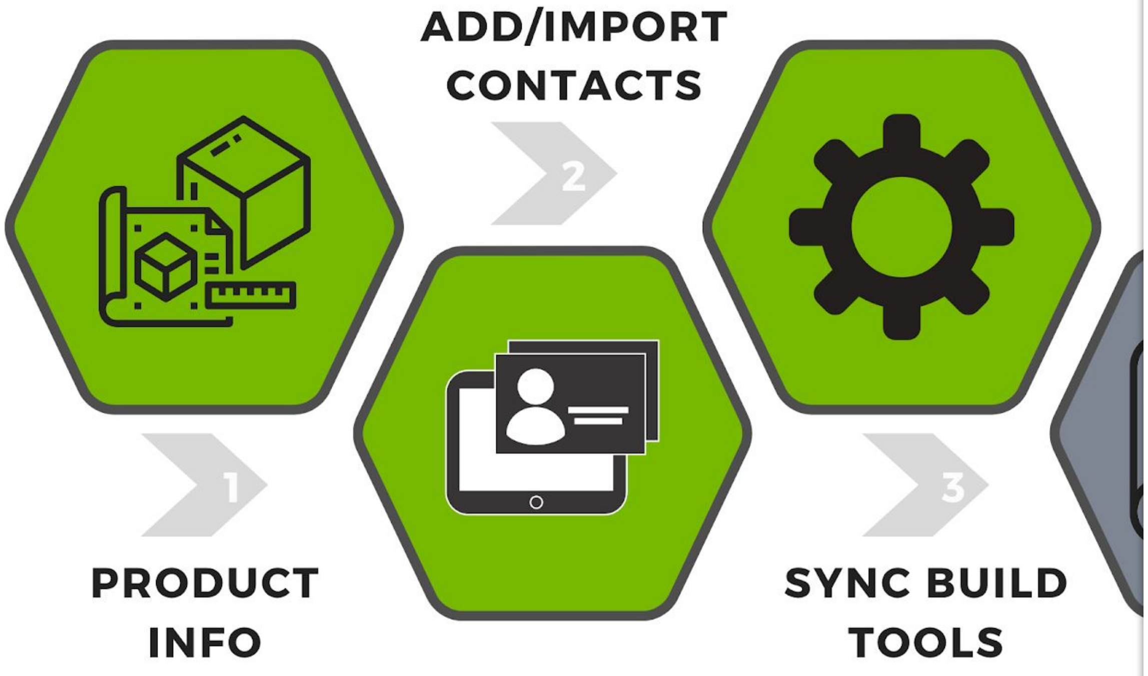
User Role *

Primary contact for this component, product, platform, service...

Self Service

REGISTRATION

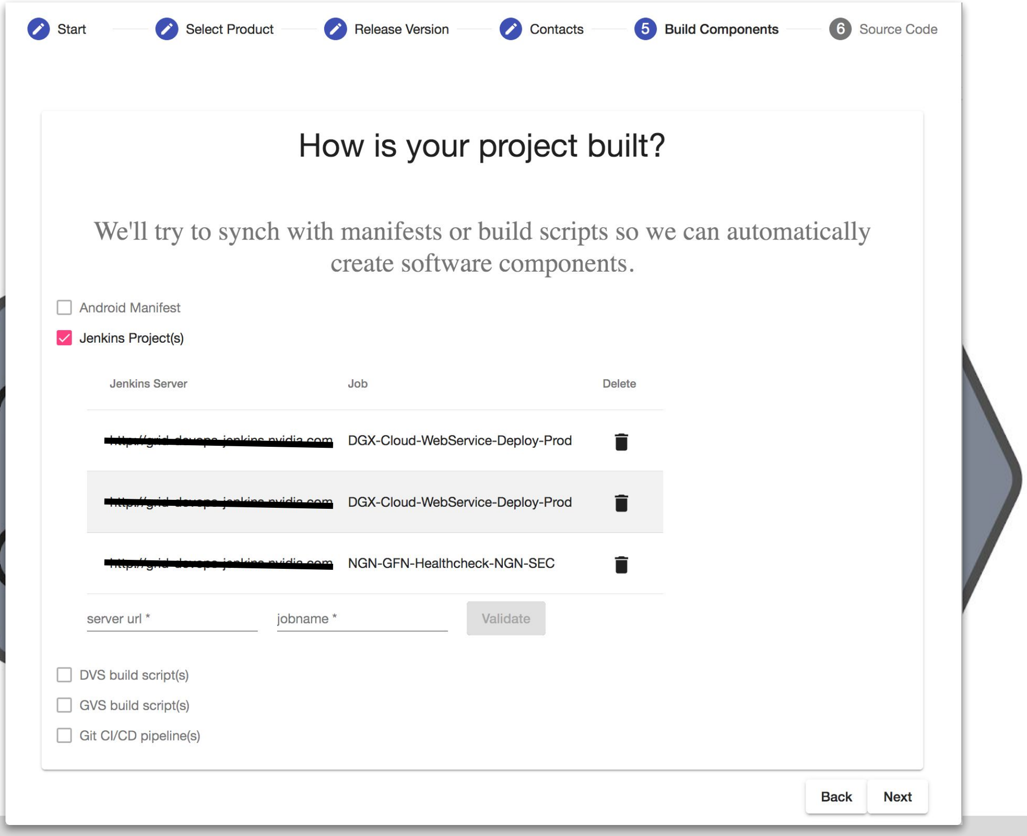
Automating component mapping via build tool synchronization



1 PRODUCT INFO

2 ADD/IMPORT CONTACTS

3 SYNC BUILD TOOLS



How is your project built?

We'll try to synch with manifests or build scripts so we can automatically create software components.

☐ Android Manifest

☑ Jenkins Project(s)

Jenkins Server	Job	Delete
[redacted]	DGX-Cloud-WebService-Deploy-Prod	🗑
[redacted]	DGX-Cloud-WebService-Deploy-Prod	🗑
[redacted]	NGN-GFN-Healthcheck-NGN-SEC	🗑

server url * ____ jobname * ____ Validate

☐ DVS build script(s)

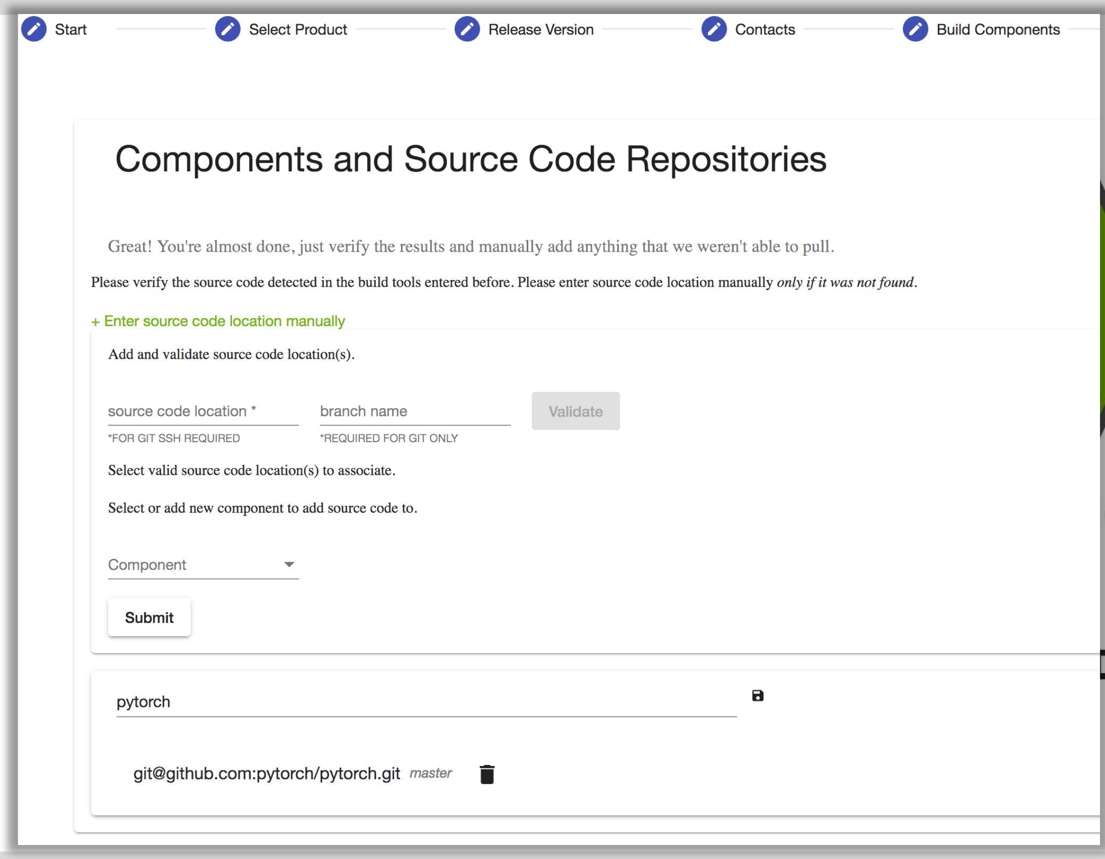
☐ GVS build script(s)

☐ Git CI/CD pipeline(s)

Self Service

REGISTRATION

Automating component mapping via build tool synchronization



Start — Select Product — Release Version — Contacts — Build Components

Components and Source Code Repositories

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

Please verify the source code detected in the build tools entered before. Please enter source code location manually *only if it was not found*.

+ Enter source code location manually

Add and validate source code location(s).

source code location * | branch name | Validate

*FOR GIT SSH REQUIRED | *REQUIRED FOR GIT ONLY

Select valid source code location(s) to associate.

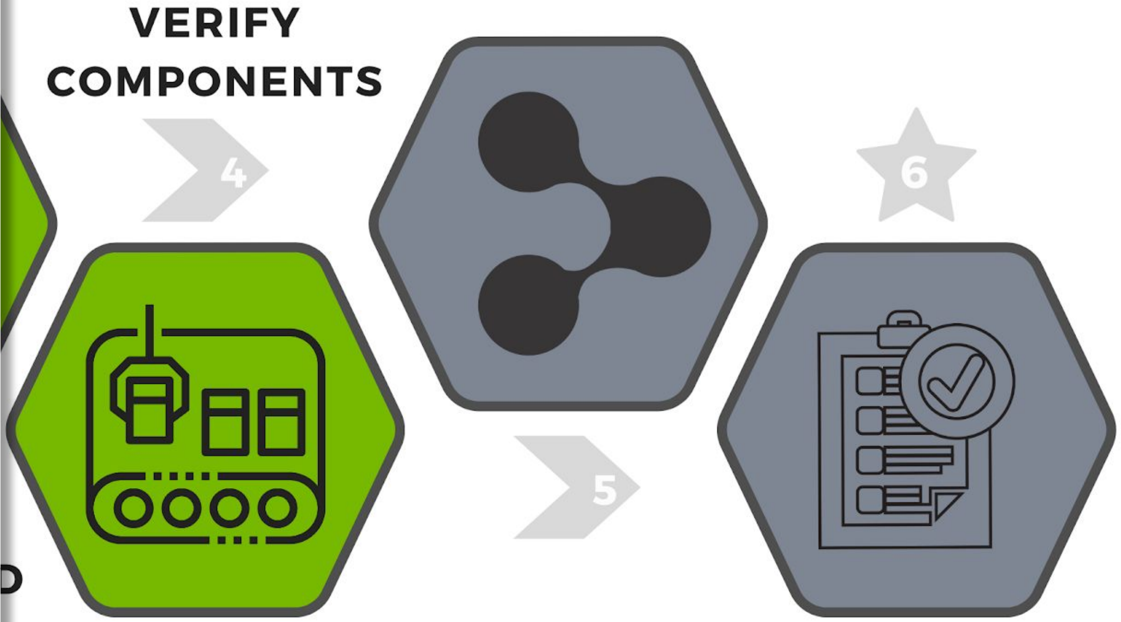
Select or add new component to add source code to.

Component

Submit

pytorch

git@github.com:pytorch/pytorch.git *master*

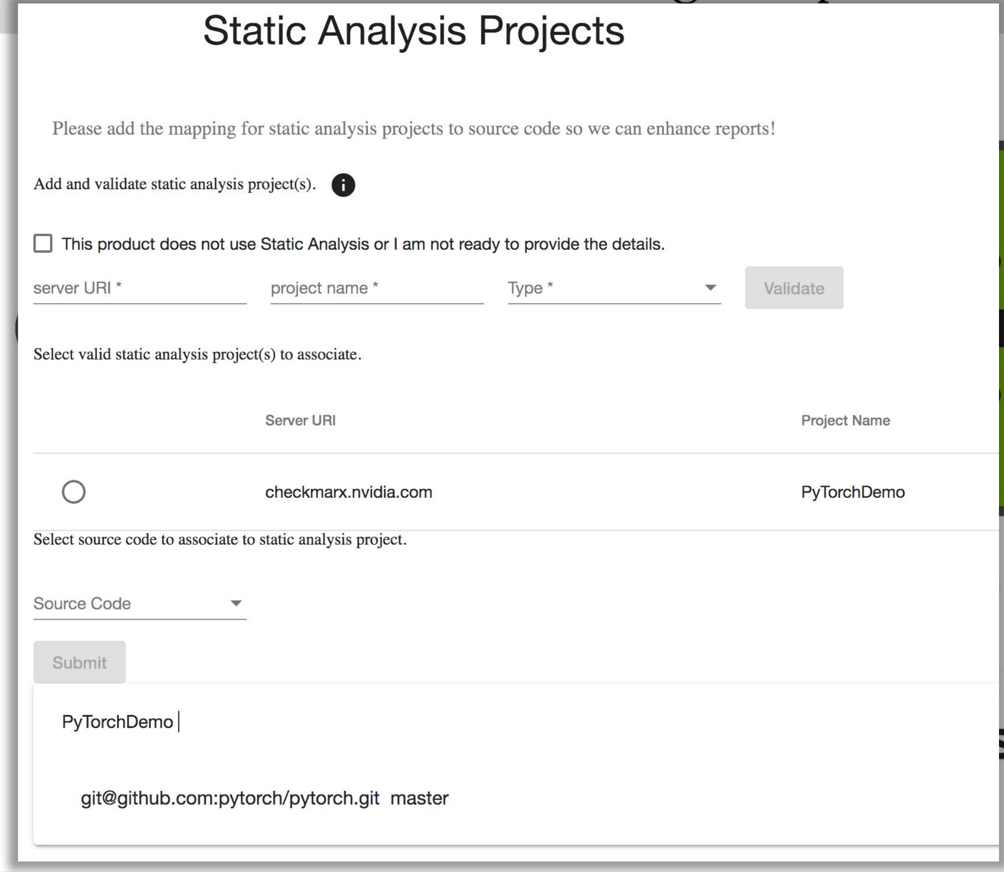


VERIFY COMPONENTS

Self Service

REGISTRATION

Automating component mapping via build tool synchronization



Static Analysis Projects

Please add the mapping for static analysis projects to source code so we can enhance reports!

Add and validate static analysis project(s).

□ This product does not use Static Analysis or I am not ready to provide the details.

server URI * | project name * | Type * | Validate

Select valid static analysis project(s) to associate.

	Server URI	Project Name
○	checkmarx.nvidia.com	PyTorchDemo

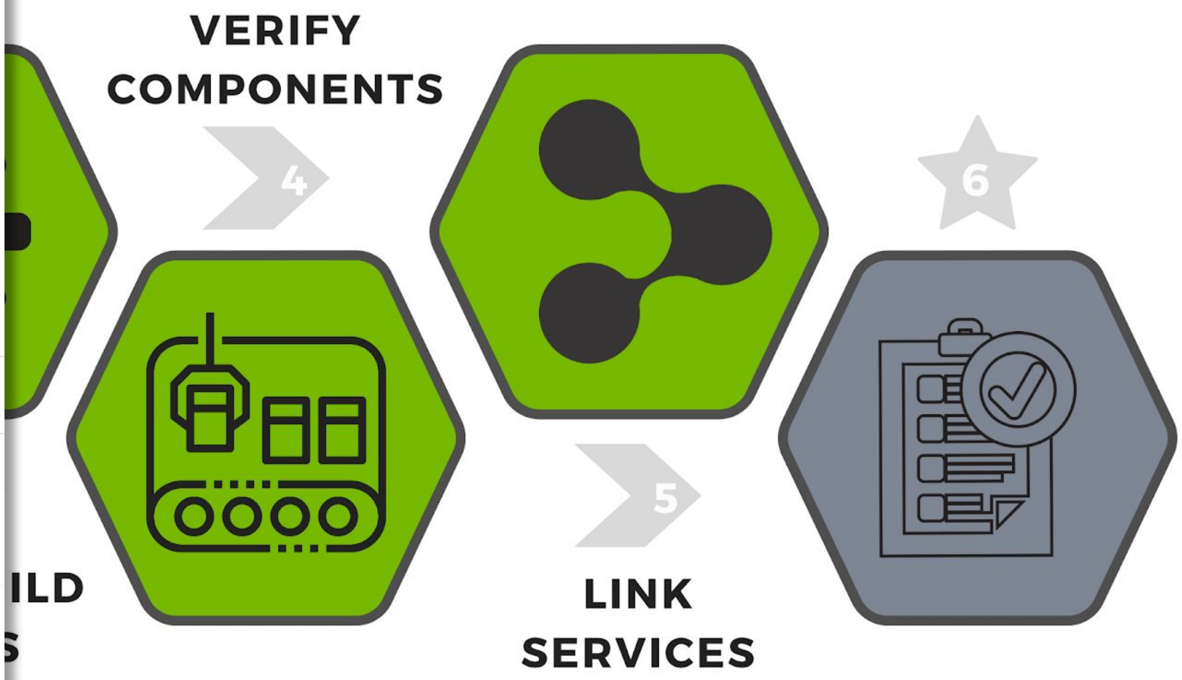
Select source code to associate to static analysis project.

Source Code

Submit

PyTorchDemo

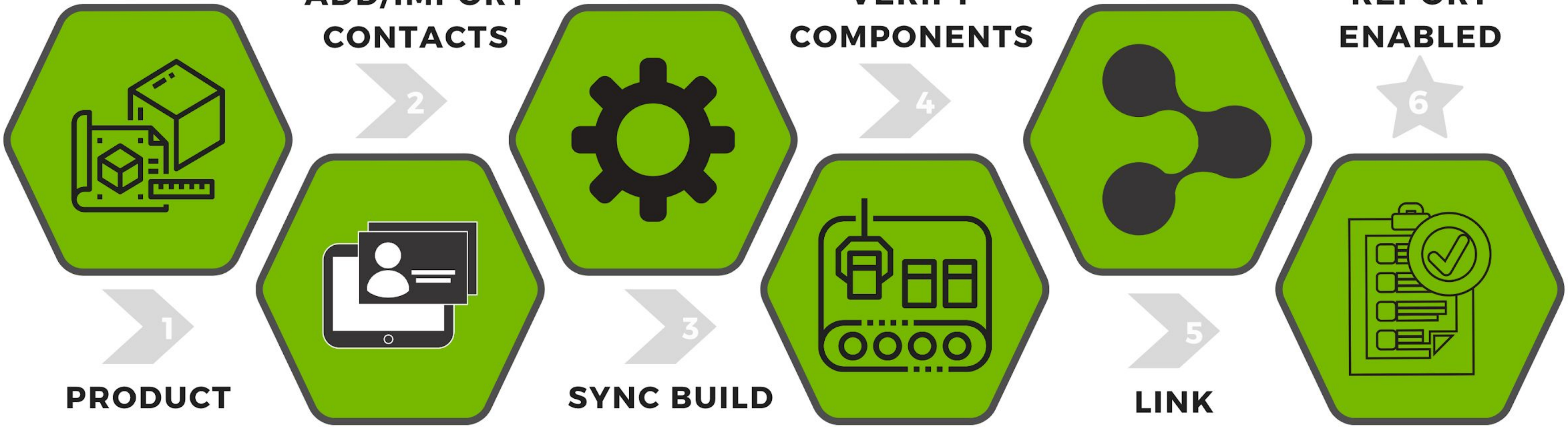
git@github.com:pytorch/pytorch.git master



Self Service

REGISTRATION

Automating component mapping via build tool synchronization



1 PRODUCT INFO

2 ADD/IMPORT CONTACTS

3 SYNC BUILD TOOLS

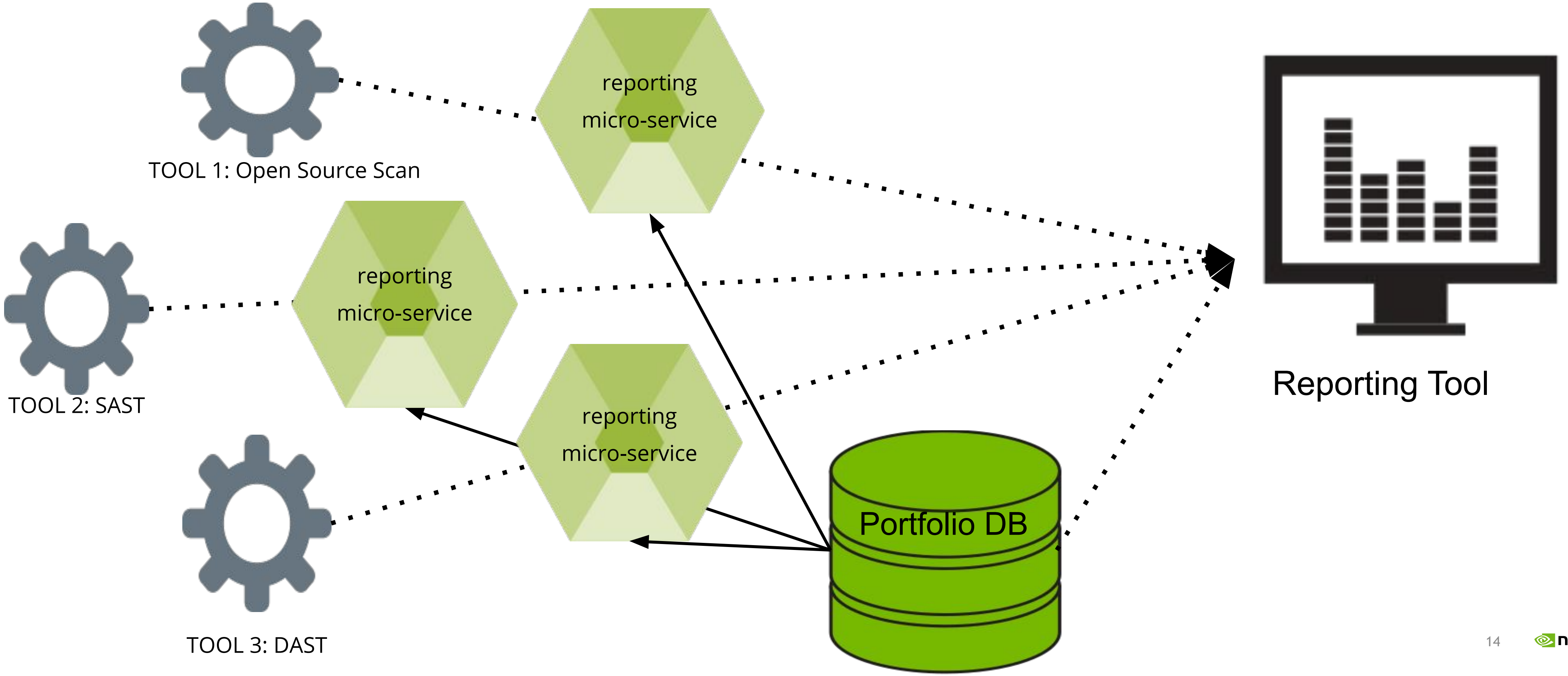
4 VERIFY COMPONENTS

5 LINK SERVICES

6 REPORT ENABLED

SELF-SERVICE REGISTRATION DEMO

Mapping the Data



Notifications

OSS Scanning Service

	Description	Use Case	User Type
Initial Report	Dashboard of OSS vulnerability distribution and other details including recommended fix	Triggered the morning after successful registration	Owner, PSIRT Lead
Weekly Report	Updated dashboard for scan results	Triggered every Monday morning	Owner, PSIRT Lead
New CVE Report	Dashboard with newly discovered CVEs for packages in Portfolio	Triggered when a new CVE is introduced for an OSS package in Portfolio	Owner, PSIRT Lead
Package Discovery alert	Email about a new undisclosed vulnerability associated with an OSS package	Triggered manually by PSIRT team about an undisclosed vulnerability associated with a particular OSS package	Owner, PSIRT Lead
Portfolio Notification	Notifications regarding updates to the product catalog hierarchy	Registered build has been deleted/modified Underlying repositories have been deleted, Registered owners are invalidated	Owner, PSIRT Lead

Issue Management

Tool Policies

1. Customize based on product types
2. Slowly increase strength for priority
3. Automate scan setup and configure

De-duplicate Bugs

1. Define filters for bug system
2. Reporting MSAs detect issue(s)
3. Portfolio DB determines owner(s)
4. Issue MSA validates new bug (and opens)

Prioritization

1. Portfolio DB detects code reuse
2. OSS Reporting MSA maps vulns
3. Report dashboard pinpoints biggest ROI

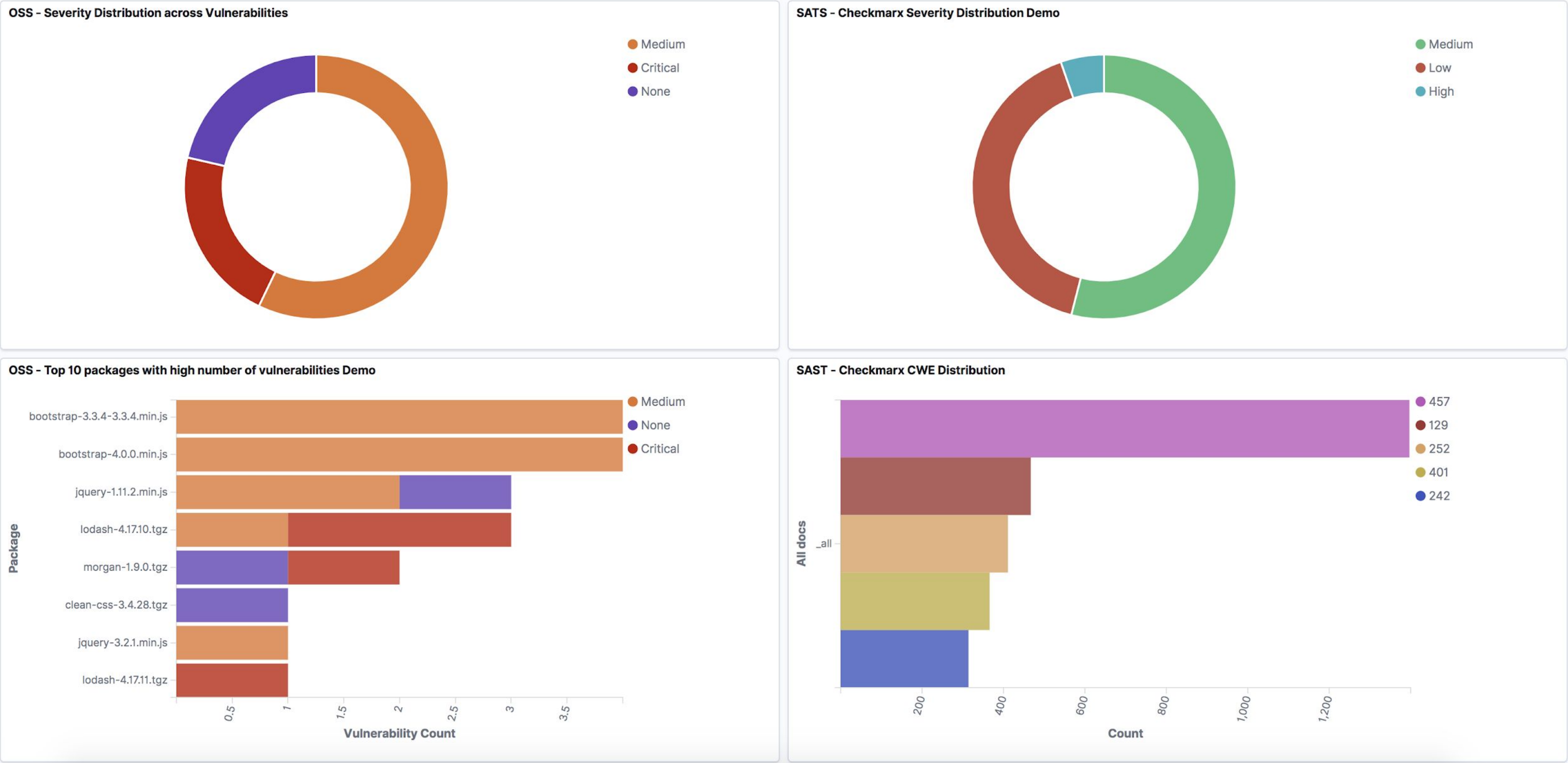
Whitelisting

1. Define format and location for list
2. Use team processes for approval
3. Require approval based on time
4. Synch to Reporting for validation

REPORT DEMO

Reports

Overall Security Risk Profile



OSS - Severity Distribution across Vulnerabilities

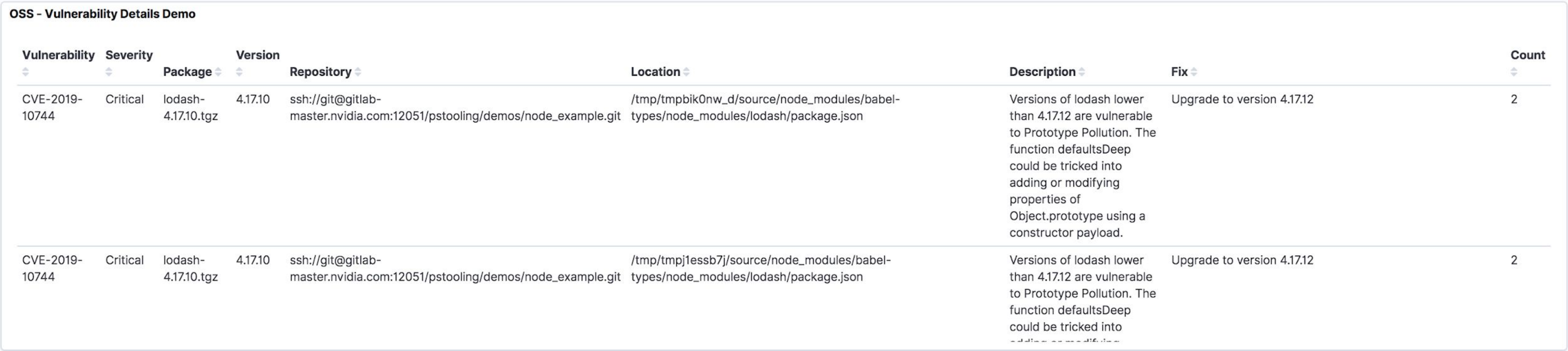
SATS - Checkmarx Severity Distribution Demo

OSS - Top 10 packages with high number of vulnerabilities Demo

SAST - Checkmarx CWE Distribution

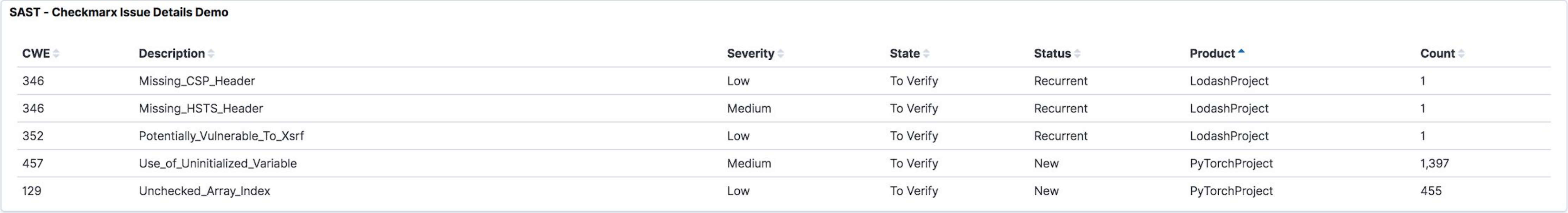
Reports

Security Risk Profile Details



OSS - Vulnerability Details Demo

Vulnerability	Severity	Package	Version	Repository	Location	Description	Fix	Count
CVE-2019-10744	Critical	lodash-4.17.10.tgz	4.17.10	ssh://git@gitlab-master.nvidia.com:12051/pstooling/demos/node_example.git	/tmp/tmpbik0nw_d/source/node_modules/babel-types/node_modules/lodash/package.json	Versions of lodash lower than 4.17.12 are vulnerable to Prototype Pollution. The function defaultsDeep could be tricked into adding or modifying properties of Object.prototype using a constructor payload.	Upgrade to version 4.17.12	2
CVE-2019-10744	Critical	lodash-4.17.10.tgz	4.17.10	ssh://git@gitlab-master.nvidia.com:12051/pstooling/demos/node_example.git	/tmp/tmpj1essb7j/source/node_modules/babel-types/node_modules/lodash/package.json	Versions of lodash lower than 4.17.12 are vulnerable to Prototype Pollution. The function defaultsDeep could be tricked into	Upgrade to version 4.17.12	2



SAST - Checkmarx Issue Details Demo

CWE	Description	Severity	State	Status	Product	Count
346	Missing_CSP_Header	Low	To Verify	Recurrent	LodashProject	1
346	Missing_HSTS_Header	Medium	To Verify	Recurrent	LodashProject	1
352	Potentially_Vulnerable_To_Xsrf	Low	To Verify	Recurrent	LodashProject	1
457	Use_of_Uninitialized_Variable	Medium	To Verify	New	PyTorchProject	1,397
129	Unchecked_Array_Index	Low	To Verify	New	PyTorchProject	455

Calculating Risk

Things to THINK on...

- Number of issues should be normalized
    - more source code
    - more open source in use
- Risk should incorporate multiple factors
    - severity
    - count
    - type
    - disclosure date
    - publicity
    - higher risk products
- Risk should be easily visualized
    - ie. a product with 3 Med issues should look different than product with 10 Med issues, etc

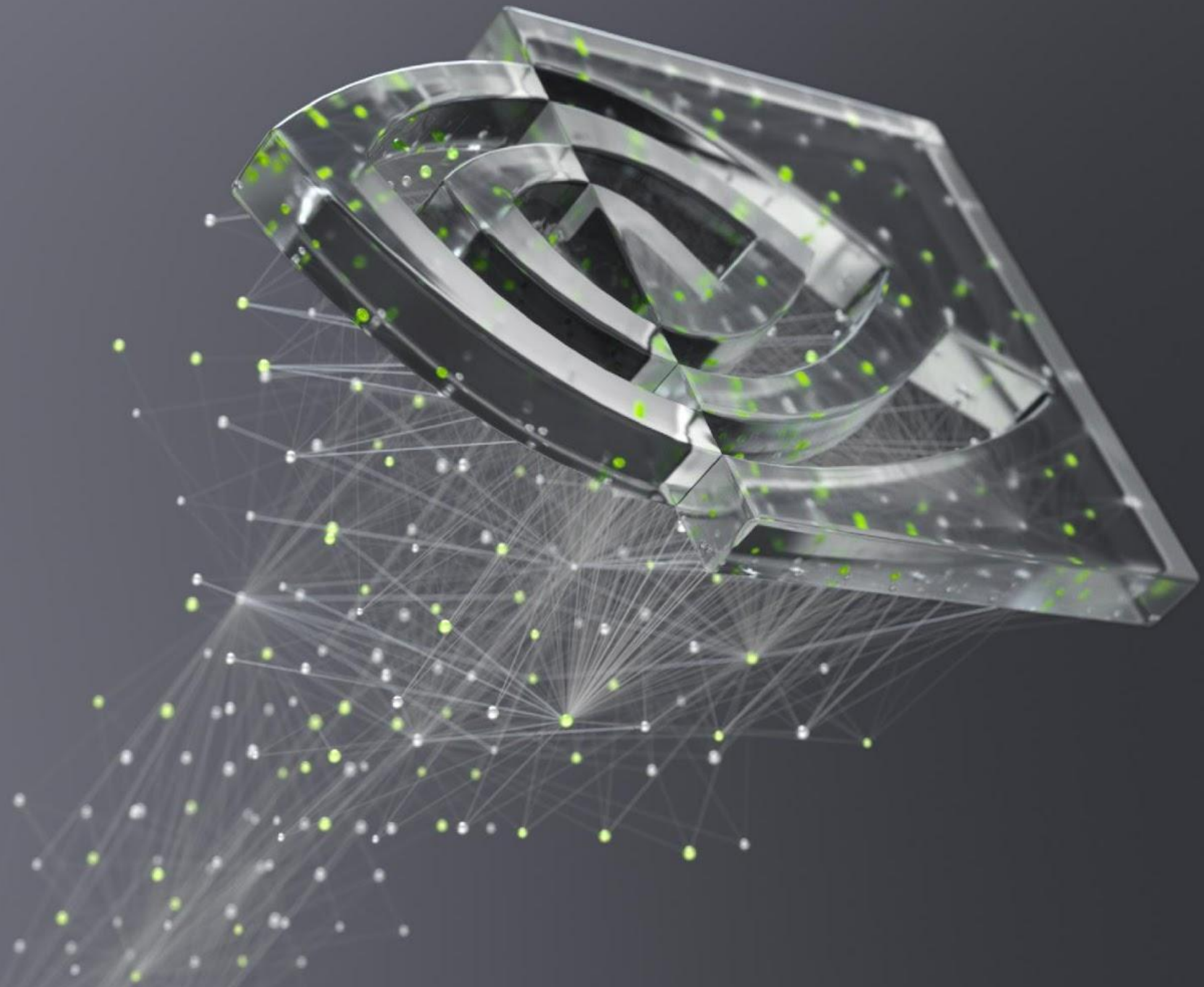
THANK YOU!

Comments, Questions, Follow-UP!

We love chatting with other Security Tools developers to knowledge share. Please contact us if you're interested in learning more or sharing!!

dannachhatre@nvidia.com

jessicab@nvidia.com



NVIDIA