ARVOS

AI- and Risk-based Vulnerability Management
for Trustworthy Open Source Adoption

debricked

elastisys

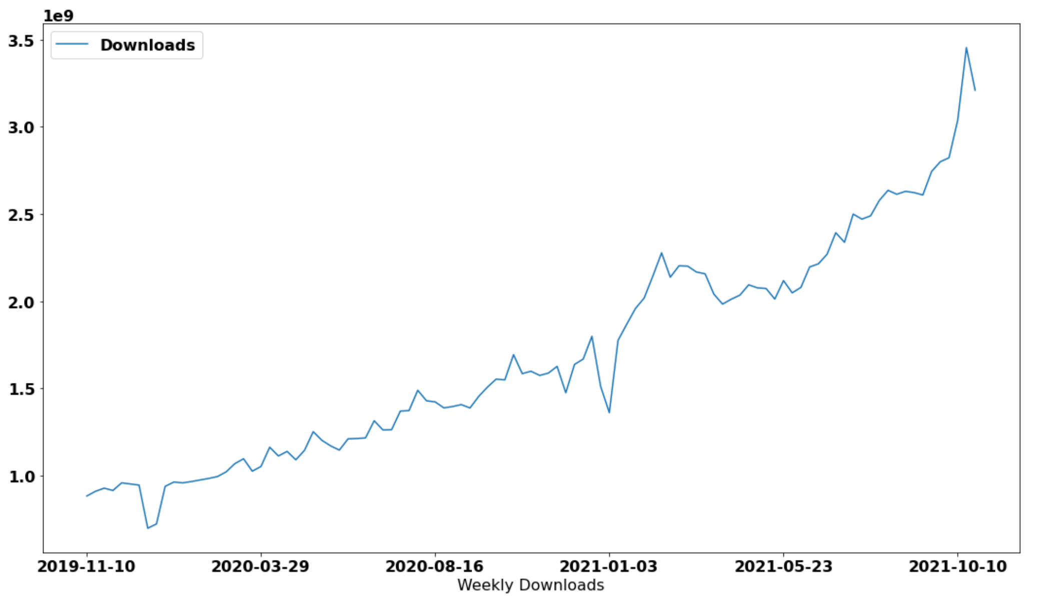
Emil Wåreus

Co-Founder & Head of Data Science

emil.wareus@debricked.com

PyPI OSS Downloads

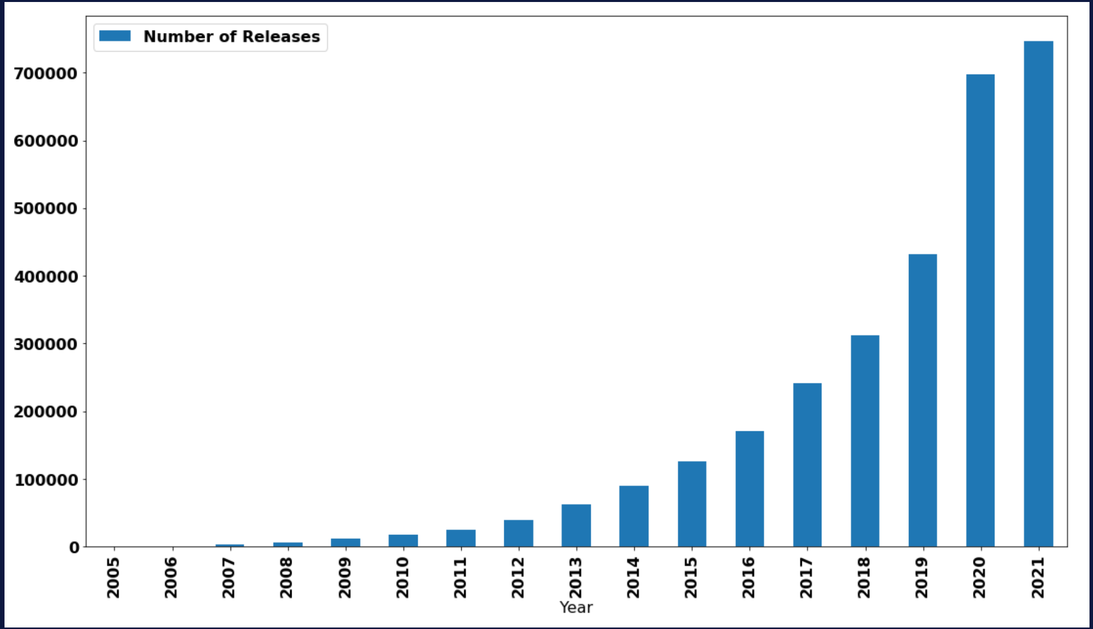
360 % growth
in 2 years



PyPI Release

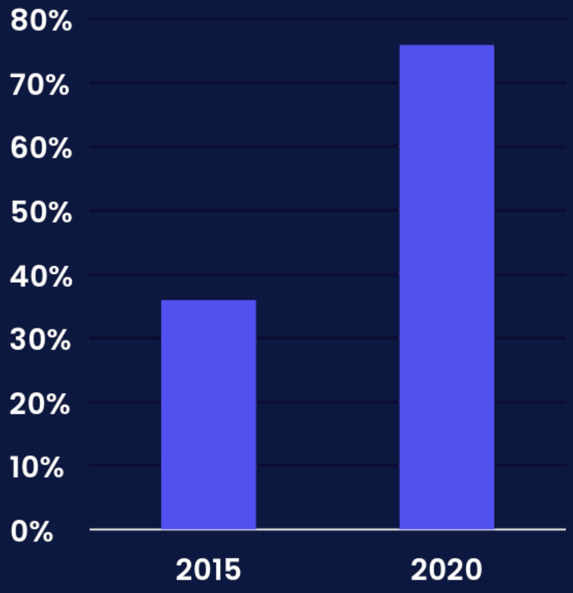
61% More releases from 2019 to 2020

2676% Growth since "Software is eating the World"



Open Source Saturation is Increasing

76% of average industry code base is open source

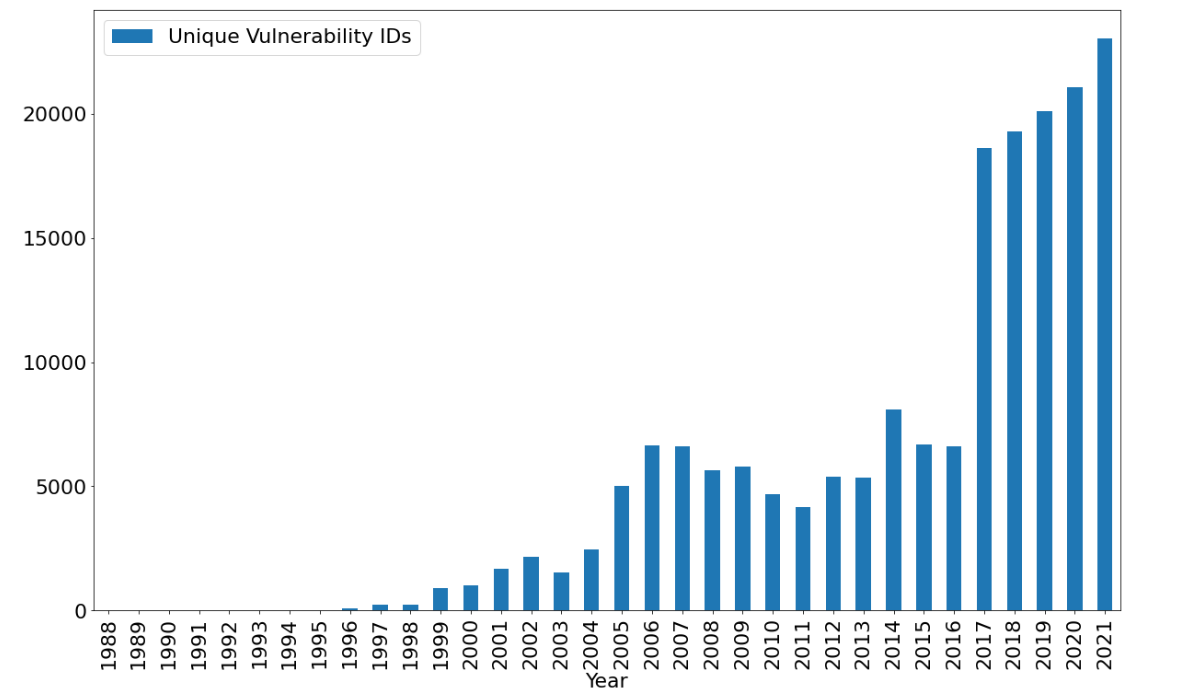


OPEN SOURCE SHARE OF CODEBASE

Vulnerabilities in Open Source

More vulnerabilities discovered each year

More alerts and work required for developers

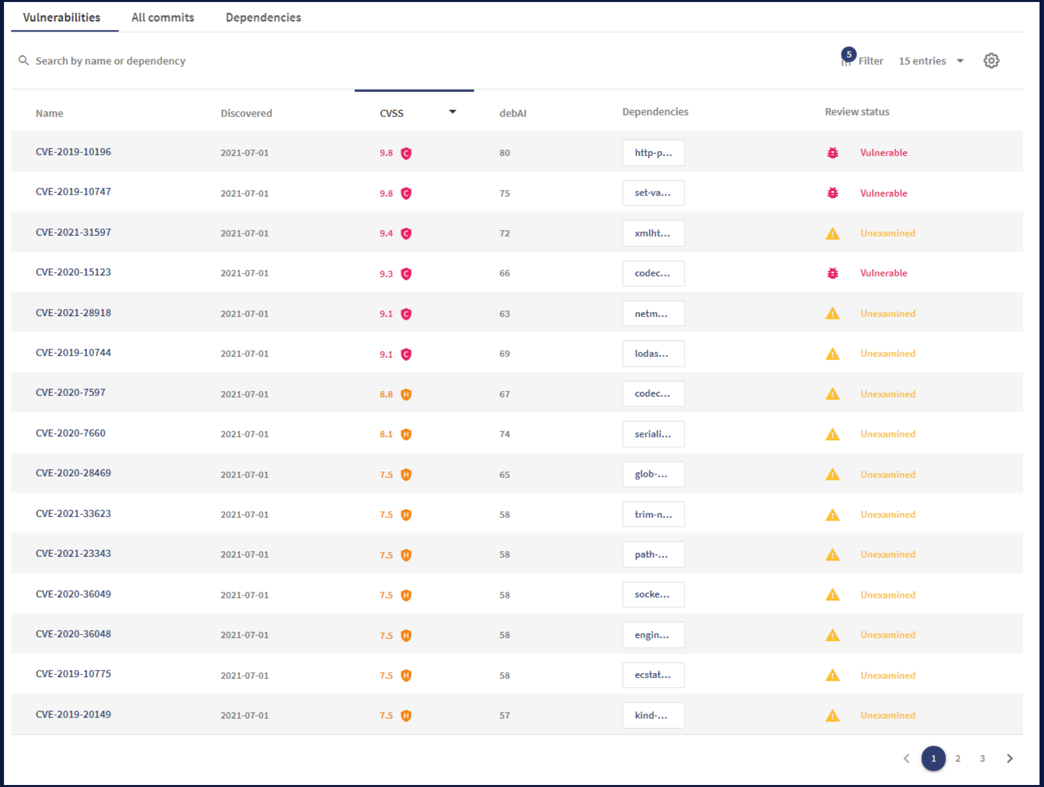


The “Cry Wolf” problem

Large lists of vulnerabilities to handle

Rich information on the vulnerability itself

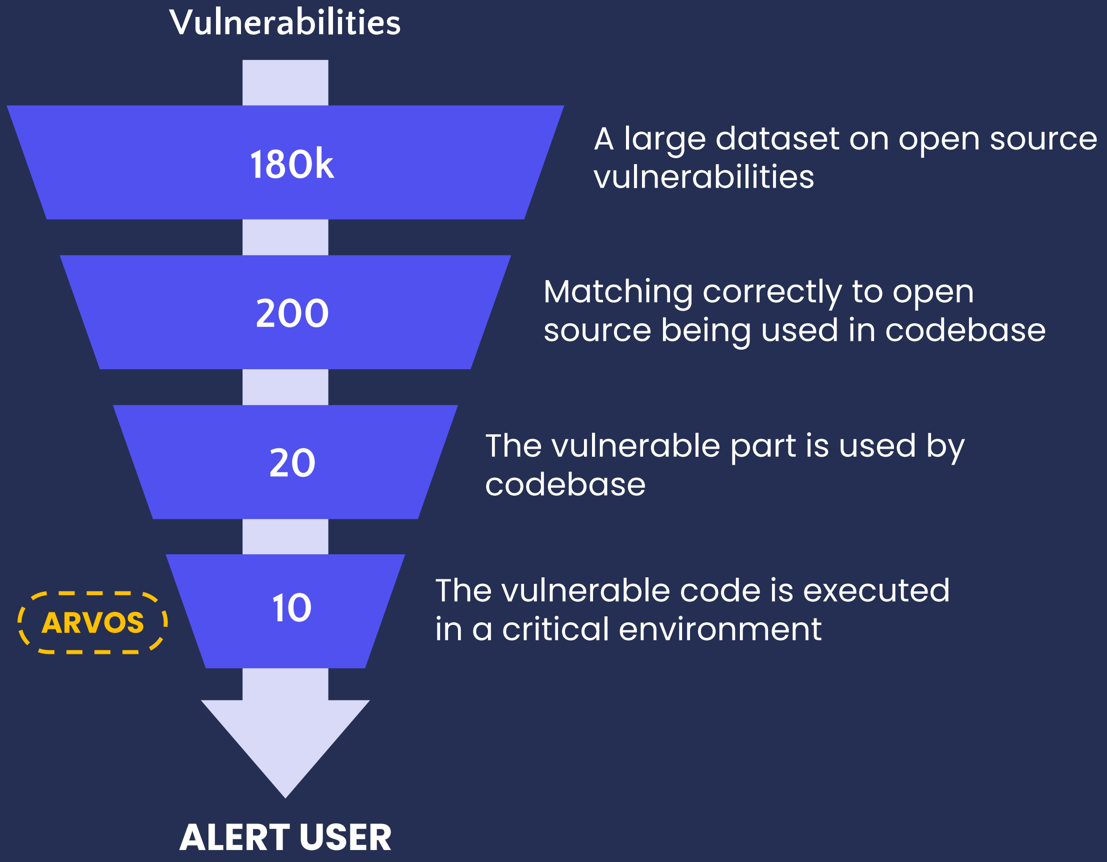
Poor contextualization to my code



Name	Discovered	CVSS	debAI	Dependencies	Review status
CVE-2019-10196	2021-07-01	9.8 C	80	http-p...	Vulnerable
CVE-2019-10747	2021-07-01	9.8 C	75	set-va...	Vulnerable
CVE-2021-31597	2021-07-01	9.4 C	72	xmlht...	⚠ Unexamined
CVE-2020-15123	2021-07-01	9.3 C	66	codec...	Vulnerable
CVE-2021-28918	2021-07-01	9.1 C	63	netm...	⚠ Unexamined
CVE-2019-10744	2021-07-01	9.1 C	69	lodas...	⚠ Unexamined
CVE-2020-7597	2021-07-01	8.8 H	67	codec...	⚠ Unexamined
CVE-2020-7660	2021-07-01	8.1 H	74	seriali...	⚠ Unexamined
CVE-2020-28469	2021-07-01	7.5 H	65	glob-...	⚠ Unexamined
CVE-2021-33623	2021-07-01	7.5 H	58	trim-n...	⚠ Unexamined
CVE-2021-23343	2021-07-01	7.5 H	58	path-...	⚠ Unexamined
CVE-2020-36049	2021-07-01	7.5 H	58	socke...	⚠ Unexamined
CVE-2020-36048	2021-07-01	7.5 H	58	engin...	⚠ Unexamined
CVE-2019-10775	2021-07-01	7.5 H	58	ecstat...	⚠ Unexamined
CVE-2019-20149	2021-07-01	7.5 H	57	kind-...	⚠ Unexamined

The 4 levels of precision

1. Are you using vulnerable OSS?
2. Are you calling the vulnerable part of the OSS?
3. Is the vulnerable part being called in a critical environment?
4. Is the vulnerability exploitable?



A large dataset on open source vulnerabilities

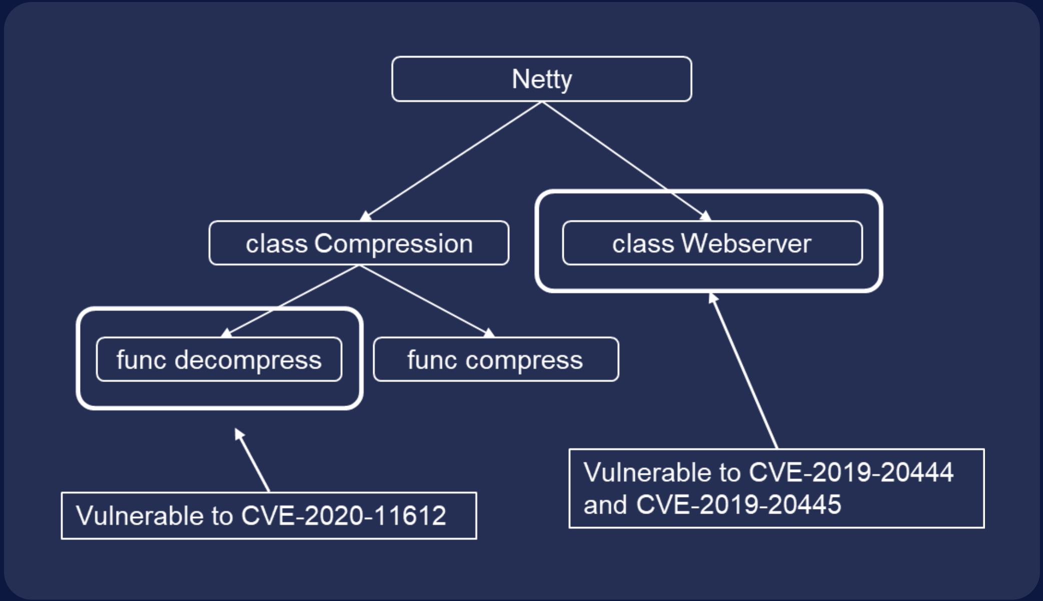
Matching correctly to open source being used in codebase

The vulnerable part is used by codebase

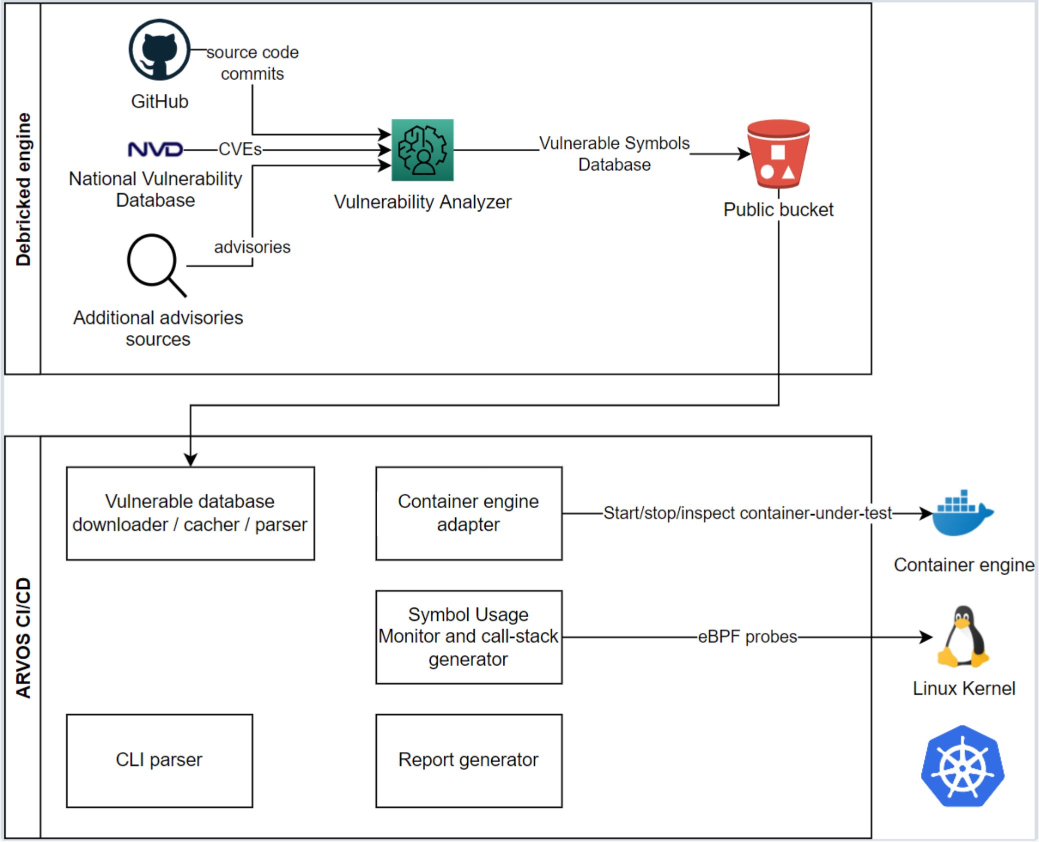
The vulnerable code is executed in a critical environment

Finding the Vulnerable Functionality

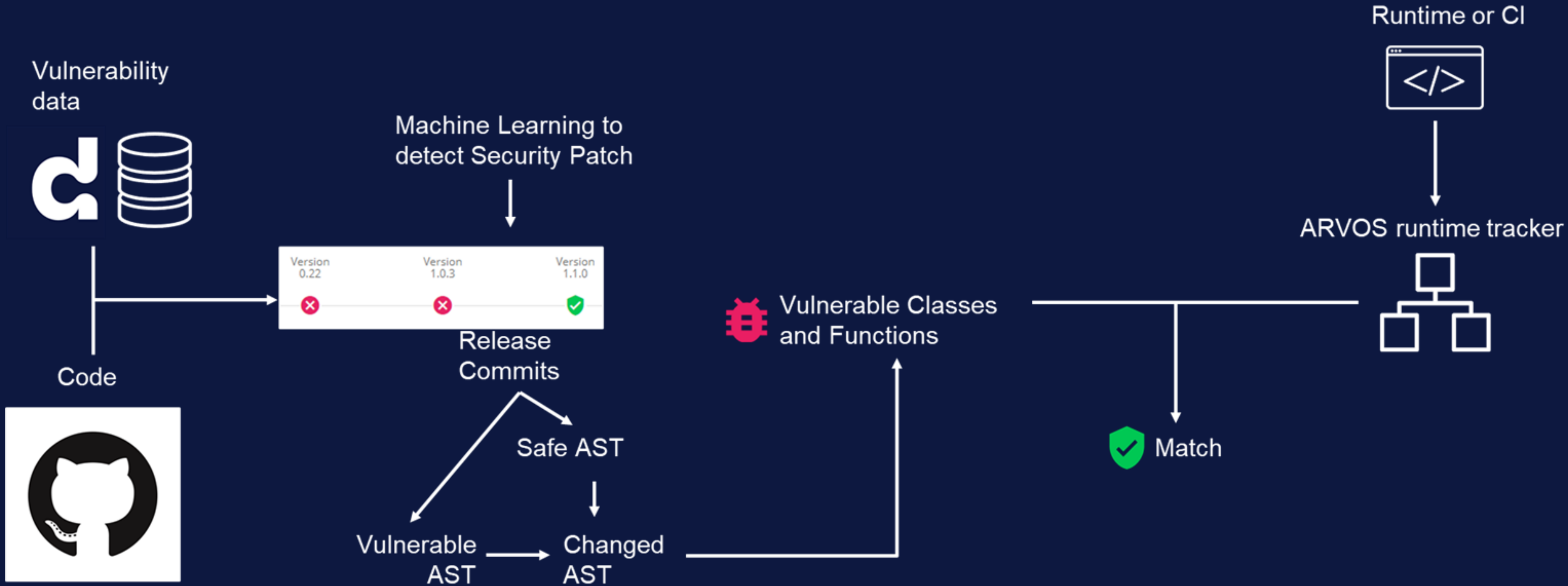
Only a part of the OSS project is affected by the affected vulnerability


Netty
class Compression
class Webserver
func decompress
func compress
Vulnerable to CVE-2020-11612
Vulnerable to CVE-2019-20444 and CVE-2019-20445

ARVOS architecture


Debricked engine
GitHub
source code commits
NVD
CVEs
National Vulnerability Database
advisories
Additional advisories sources
Vulnerability Analyzer
Vulnerable Symbols Database
Public bucket
ARVOS CI/CD
Vulnerable database downloader / cacher / parser
Container engine adapter
Start/stop/inspect container-under-test
Container engine
Symbol Usage Monitor and call-stack generator
eBPF probes
Linux Kernel
CLI parser
Report generator

Finding the Vulnerable Functionality



Vulnerability data

Code

Machine Learning to detect Security Patch

Release Commits

Safe AST

Vulnerable AST

Changed AST

Vulnerable Classes and Functions

Runtime or CI

ARVOS runtime tracker

Match

Key Findings

We have validated the importance of deep contextualisation of vulnerabilities (precision level 2 and 3)

CISO/CTO/Managers want to shift left, and perform scan in CI only

Developers see a lot of value to track in production, CI, and as a debugging tool

We should develop a good “core” that can be extended to all theses use cases

Thank you!

emil.wareus@debricked.com

https://github.com/arvos-dev