Open source, Check, Security, Check:
A checklist for securing open source projects

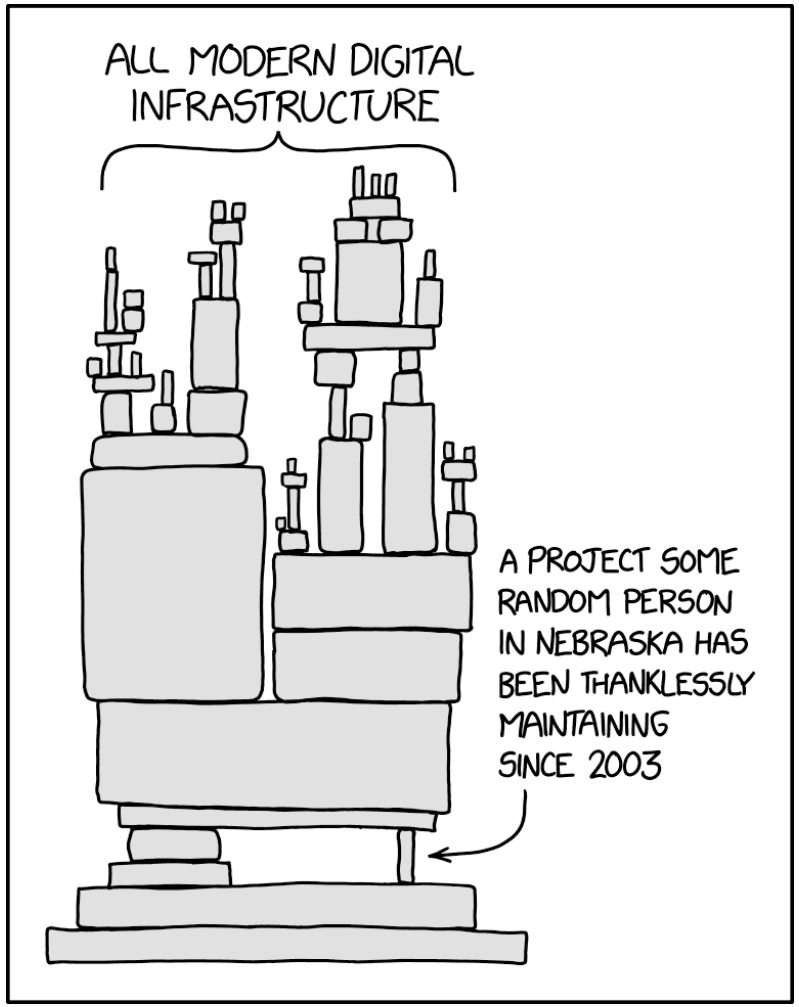

MutableSecurity
Seamless deployment and management of cybersecurity solutions


RETIRE


BILBAO
20 23
INNOVATION
HAPPENS HERE


Ubuntu
Security Podcast


DAY & NIGHT


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

YES,

- Large scale use in:
  - Profitable companies
  - Critical infrastructures
- Permissive licences
- Publicly reviewable code

BUT

- Unpaid maintainers
- Unmaintained, vulnerable projects
- Lack of ethical security testing
- Low-hanging fruits for threat actors

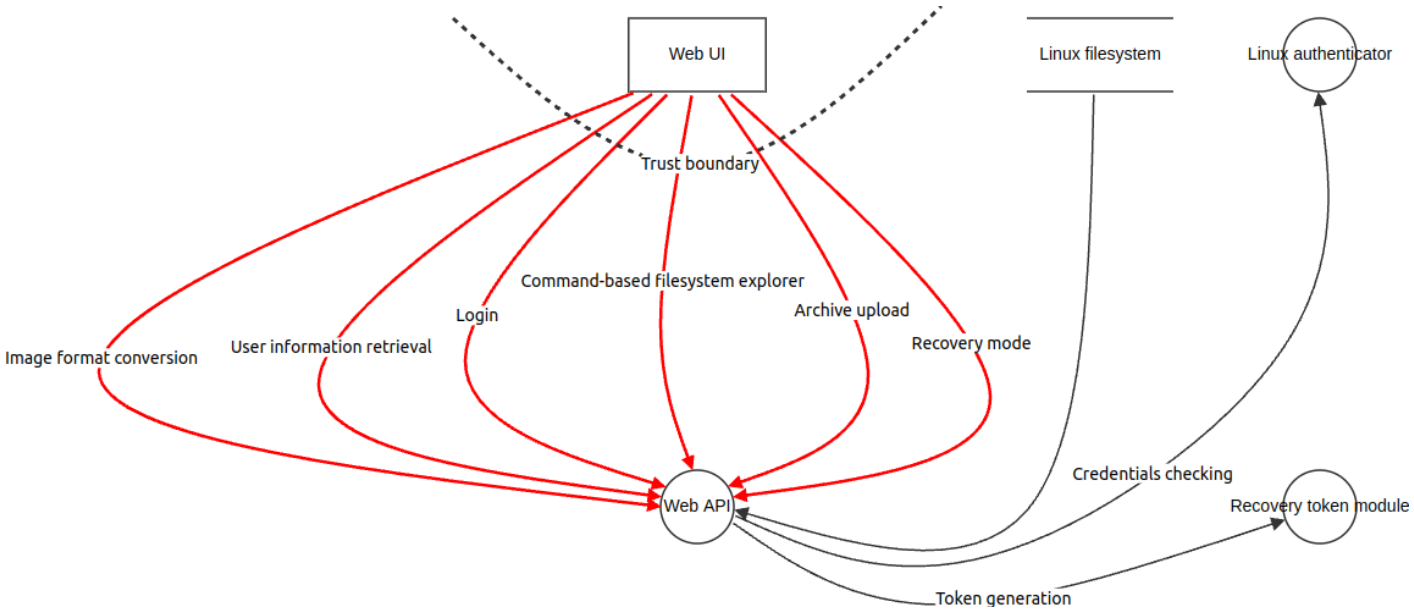

CHECK AFTER ENGINE START
1 Oil pressure CHECKED 1 pf
2 RPM 740 +/- 30 CHECKED 2 pf
3 Warm up time START 3 pf
Warm up: 2 minutes
Idle until Oil >50° and Coolant > 60°C
50% load
4 Fuel pumps (2) check OFF 4 pf
5 Fuel selectors (2) X-FEED 5 pf
6 Pitot heat ON, annunciation + Amps checked 6 pf
7 Pitot heat OFF 7 pf
8 PFD/MFD NORMAL MODE 8 pf
9 Avionics master ON 9 pf
FMS SETUP
I nitialize profile (AUX 4, MAP)
F light plan
R adios (COM, NAV, ADF, DME, CDI, BRG ½)
P erformance (speed bugs; Flight ID if applicable)
COMPLETED 10 pf
10 FMS setup
AUTOPILOT TEST
DISCONN press, check electric trim not working
AP ON, check annunciations and FD
DISCONN press, check AP off
GA button press, check FD commands climb
COMPLETED 11 pf
11 Autopilot test
12 Flood light
CODE / MODE
Diamond Aircraft Flight Training Division
rudder trim
AP DISC
PTT

~~Notations~~ Emoji time!

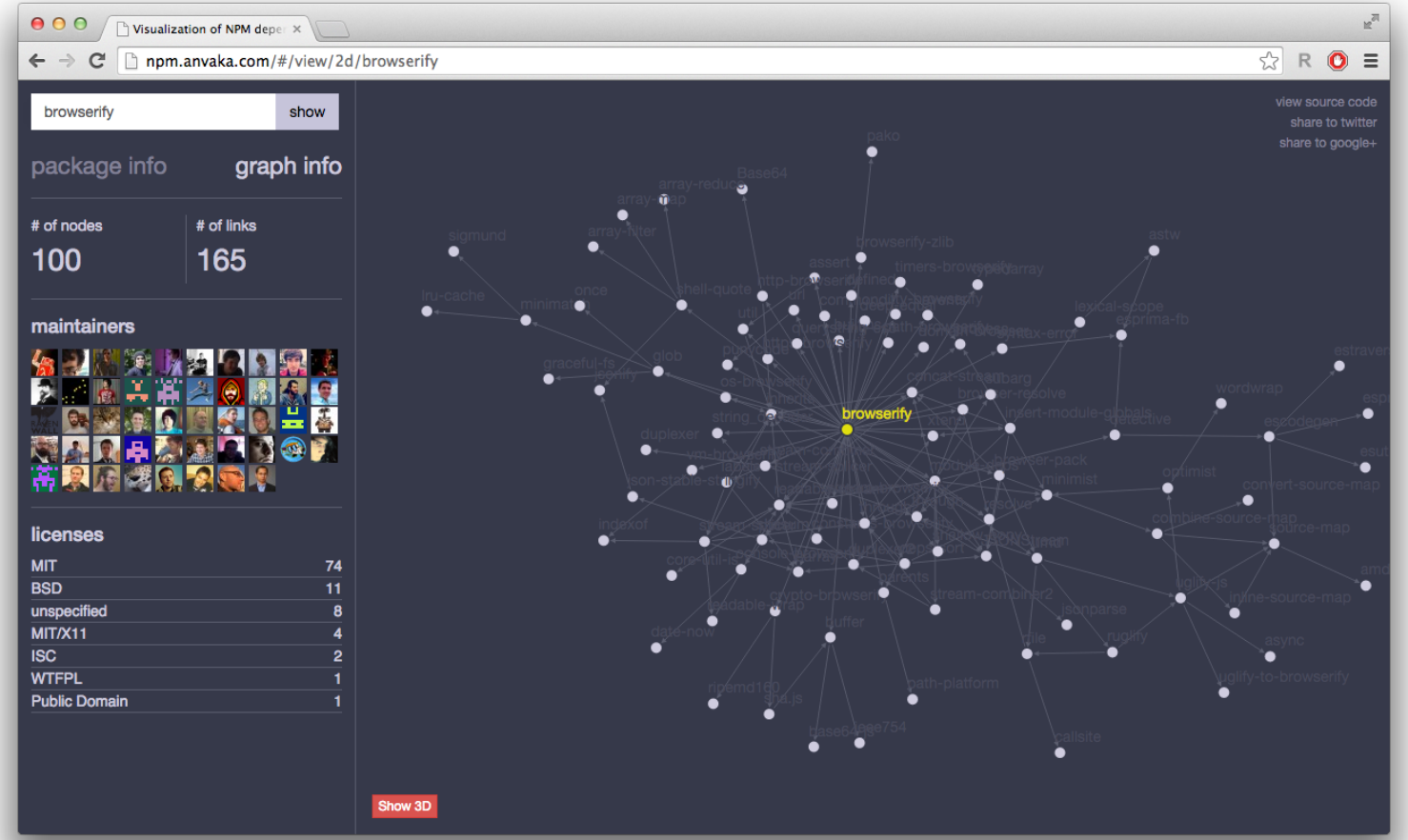
- ☑️ for (wanna-be) one-time activities
- 🔁 for recurrent activities
- 📦 for closed source friendly activities

I. Proactively find vulnerabilities

1. Create and maintain a threat model ☑️🔁📦


Web UI
Linux filesystem
Linux authenticator
Trust boundary
Command-based filesystem explorer
Login
Archive upload
Image format conversion
User information retrieval
Recovery mode
Credentials checking
Web API
Recovery token module
Token generation

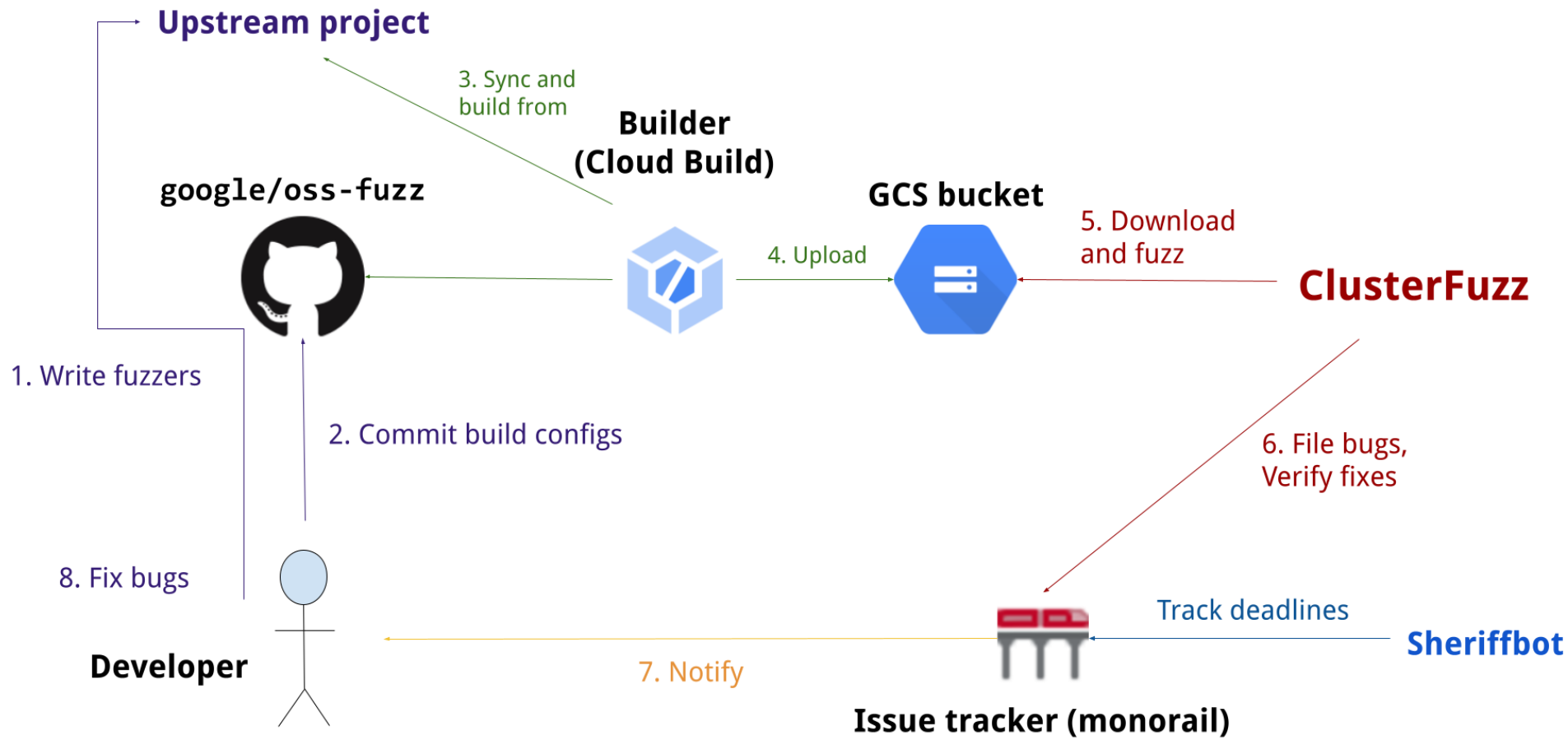
2. Check for vulnerabilities in your dependencies ☑️🔁


Visualization of NPM depe
npm.anvaka.com/#/view/2d/browserify
browserify
show
package info
graph info
# of nodes
100
# of links
165
maintainers
licenses
MIT 74
BSD 11
unspecified 8
MIT/X11 4
ISC 2
WTFPL 1
Public Domain 1
view source code
share to twitter
share to google+
Show 3D

3. Run security tools and constantly validate the warnings ☑️🔁📦

1. Run multiple tools
2. Aggregate the results (e.g., with the SARIF format)
3. Review the results
4. Suppress the false positives
5. Create automation for development environments and CI workflows

4. Integrate your project in OSS-Fuzz ☑️🔁


Upstream project
3. Sync and build from
Builder (Cloud Build)
google/oss-fuzz
GCS bucket
4. Upload
5. Download and fuzz
ClusterFuzz
1. Write fuzzers
2. Commit build configs
6. File bugs, Verify fixes
8. Fix bugs
Track deadlines
Sheriffbot
Developer
7. Notify
Issue tracker (monorail)

II. Secure your users

1. Design your software to be secure by default ☑️🔁📦

Chromium Blog

News and developments from the open source browser project

A safer default for navigation: HTTPS

Tuesday, March 23, 2021

Starting in version 90, Chrome's address bar will use *https://* by default, improving privacy and even loading speed for users visiting websites that support HTTPS. Chrome users who navigate to websites by manually typing a URL often don't include "http://" or "https://". For example, users often type "example.com" instead of "https://example.com" in the address bar. In this case, if it was a user's first visit to a website, Chrome would previously choose *http://* as the default protocol[1]. This was a practical default in the past, when much of the web did not support HTTPS.

2. Have security recommendations for users ☑️🔁📦

Node.js Security Best Practices

Intent

This document intends to extend the current threat model and provide extensive guidelines on how to secure a Node.js application.

Document Content

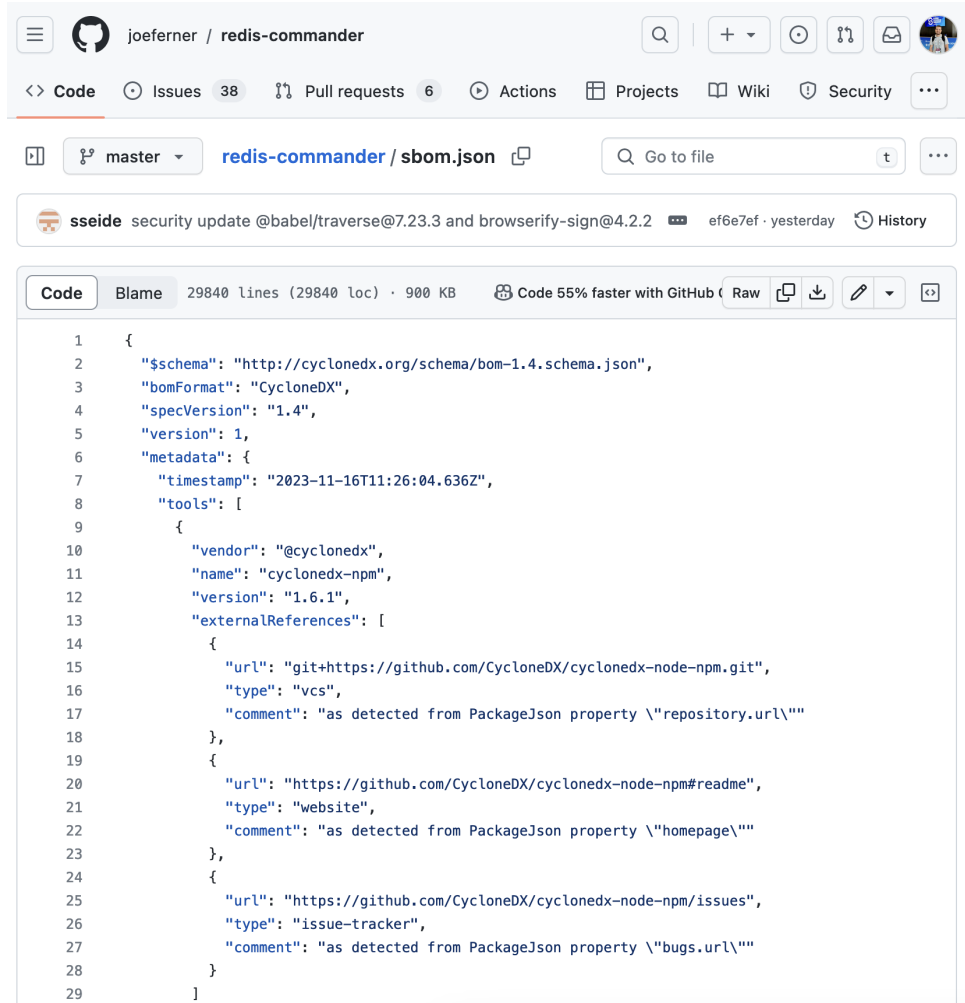
- Best practices: A simplified condensed way to see the best practices. We can use this issue or this guideline as the starting point. It is important to note that this document is specific to Node.js, if you are looking for something broad, consider OSSF Best Practices.
- Attacks explained: illustrate and document in plain English with some code example (if possible) the attacks that we are mentioning in the threat model.
- Third-Party Libraries: define threats (typosquatting attacks, malicious packages...) and best practices regarding node modules dependencies, etc...

Threat List

Denial of Service of HTTP server (CWE-400)

This is an attack where the application becomes unavailable for the purpose it was designed due to the way it processes incoming HTTP requests. These requests need not be deliberately crafted by a malicious actor: a misconfigured or buggy client can also send a pattern of requests to the server that result in a denial of service.

3. Create SBOMs ☑️📦


joeferner / redis-commander
Code
Issues 38
Pull requests 6
Actions
Projects
Wiki
Security
master
redis-commander / sbom.json
Go to file
sseide security update @babel/traverse@7.23.3 and browserify-sign@4.2.2
ef6e7ef · yesterday
History
Code
Blame
29840 lines (29840 loc) · 900 KB
Code 55% faster with GitHub
Raw
{
"$schema": "http://cyclonedx.org/schema/bom-1.4.schema.json",
"bomFormat": "CycloneDX",
"specVersion": "1.4",
"version": 1,
"metadata": {
"timestamp": "2023-11-16T11:26:04.636Z",
"tools": [
{
"vendor": "@cyclonedx",
"name": "cyclonedx-npm",
"version": "1.6.1",
"externalReferences": [
{
"url": "git+https://github.com/CycloneDX/cyclonedx-node-npm.git",
"type": "vcs",
"comment": "as detected from PackageJson property \"repository.url\""
},
{
"url": "https://github.com/CycloneDX/cyclonedx-node-npm#readme",
"type": "website",
"comment": "as detected from PackageJson property \"homepage\""
},
{
"url": "https://github.com/CycloneDX/cyclonedx-node-npm/issues",
"type": "issue-tracker",
"comment": "as detected from PackageJson property \"bugs.url\""
}
]

III. Establish a security reporting process

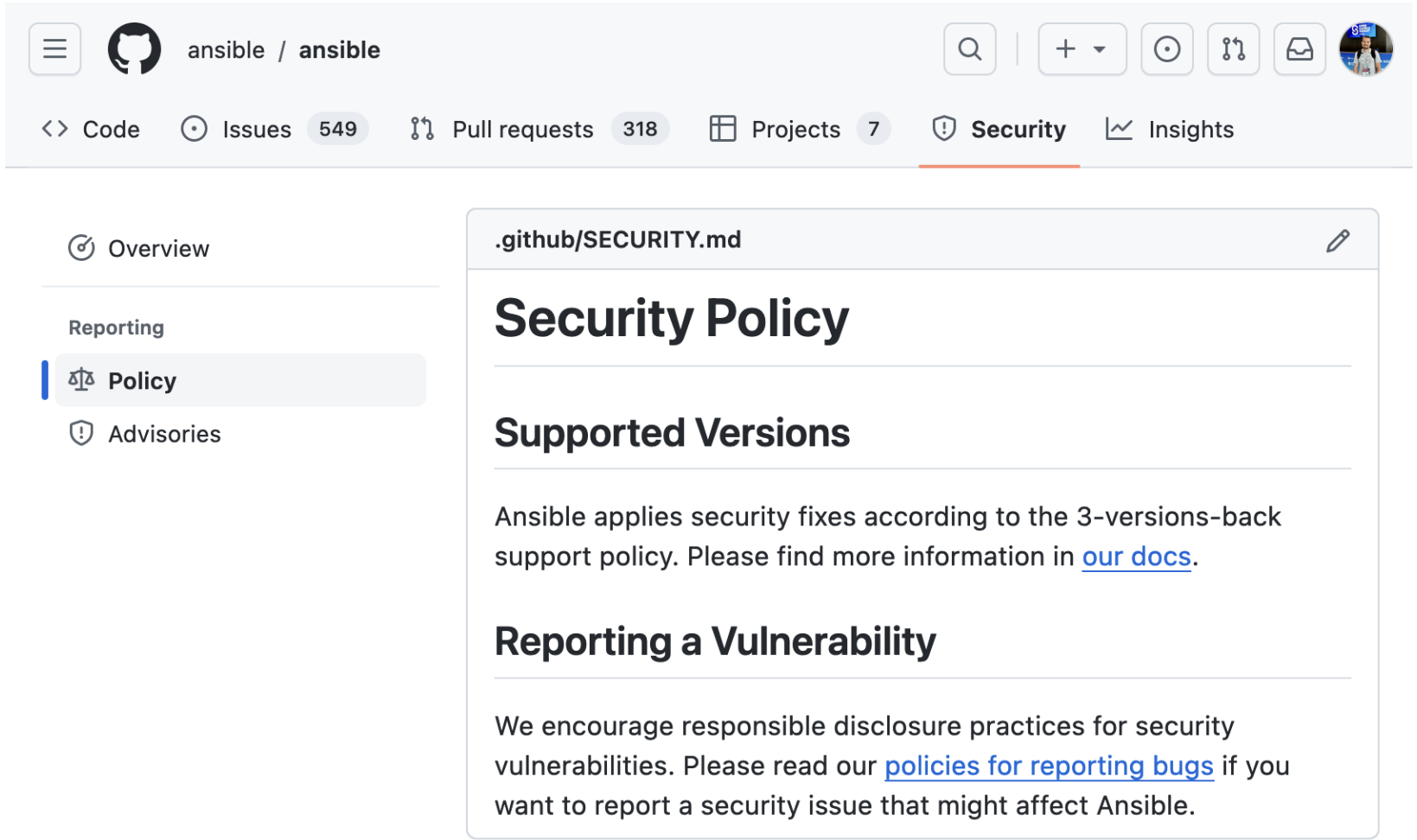
1. Have a standardised, documented process for responding to vulnerabilities ☑️📦

Disclosure policy

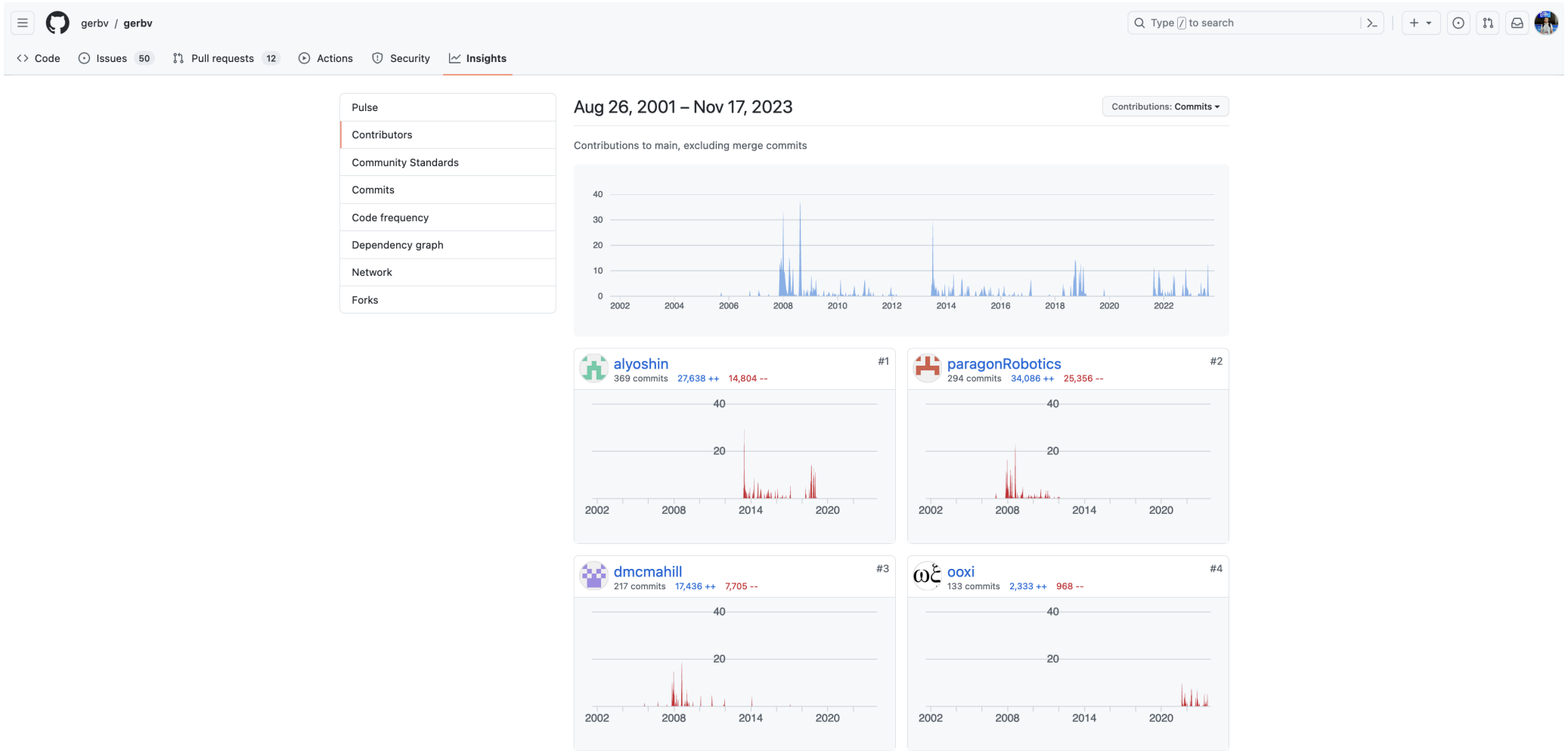
Here is the security disclosure policy for Node.js

- The security report is received and is assigned a primary handler. This person will coordinate the fix and release process. The problem is confirmed and a list of all affected versions is determined. Code is audited to find any potential similar problems. Fixes are prepared for all releases which are still under maintenance. These fixes are not committed to the public repository but rather held locally pending the announcement.
- A suggested embargo date for this vulnerability is chosen and a CVE (Common Vulnerabilities and Exposures (CVE®)) is requested for the vulnerability.
- On the embargo date, the Node.js security mailing list is sent a copy of the announcement. The changes are pushed to the public repository and new builds are deployed to nodejs.org. Within 6 hours of the mailing list being notified, a copy of the advisory will be published on the Node.js blog.
- Typically the embargo date will be set 72 hours from the time the CVE is issued. However, this may vary depending on the severity of the bug or difficulty in applying a fix.
- This process can take some time, especially when coordination is required with maintainers of other projects. Every effort will be made to handle the bug in as timely a manner as possible; however, it's important that we follow the release process above to ensure that the disclosure is handled in a consistent manner.

2. Create a security policy ☑️📦


ansible / ansible
Code
Issues 549
Pull requests 318
Projects 7
Security
Insights
Overview
Reporting
Policy
Advisories
.github/SECURITY.md
Security Policy
Supported Versions
Ansible applies security fixes according to the 3-versions-back support policy. Please find more information in our docs.
Reporting a Vulnerability
We encourage responsible disclosure practices for security vulnerabilities. Please read our policies for reporting bugs if you want to report a security issue that might affect Ansible.

3. Find backup security responders ☑️📦


gerbv / gerbv
Type / to search
Code
Issues 50
Pull requests 12
Actions
Security
Insights
Pulse
Contributors
Community Standards
Commits
Code frequency
Dependency graph
Network
Forks
Aug 26, 2001 – Nov 17, 2023
Contributions: Commits
Contributions to main, excluding merge commits
40
30
20
10
0
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
alyoshin
#1
369 commits 27,638 ++ 14,804 --
paragonRobotics
#2
294 commits 34,086 ++ 25,356 --
dmcmahill
#3
217 commits 17,436 ++ 7,705 --
ooxi
#4
133 commits 2,333 ++ 968 --

4. Be transparent and verbose with the reported vulnerabilities 🔁📦

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-4508	Third Party Advisory
https://github.com/gerbv/gerbv/commit/dfb5aac533a3f9e8ccd93ca217a753258cba4fe5	Patch
https://github.com/gerbv/gerbv/issues/191	Exploit Issue Tracking Third Party Advisory
https://lists.debian.org/debian-lts-announce/2023/09/msg00040.html	

Known Affected Software Configurations Switch to CPE 2.2

Configuration 1 (hide)

cpe:2.3:a:gerbv_project:gerbv:*:*:*:*:*:*:*:*	From (including)	Up to (including)
Show Matching CPE(s)	2.4.0	2.10.0

The OSS Fortress

- Workshop for finding software vulnerabilities using open source tools
- Vulnerable-by-default Python and C web application
- Tasks (and solutions) for linting, code querying, secret scanning, dependency scanning, fuzzing, and symbolic execution
- `iosifache/oss_fortress` on GitHub
- `ossfortress.io` as a wiki


That's all Folks!

`ping @iosifache`

- Website: iosifache.me
- GitHub: @iosifache
- X: @iosifache
- LinkedIn: @iosifache