

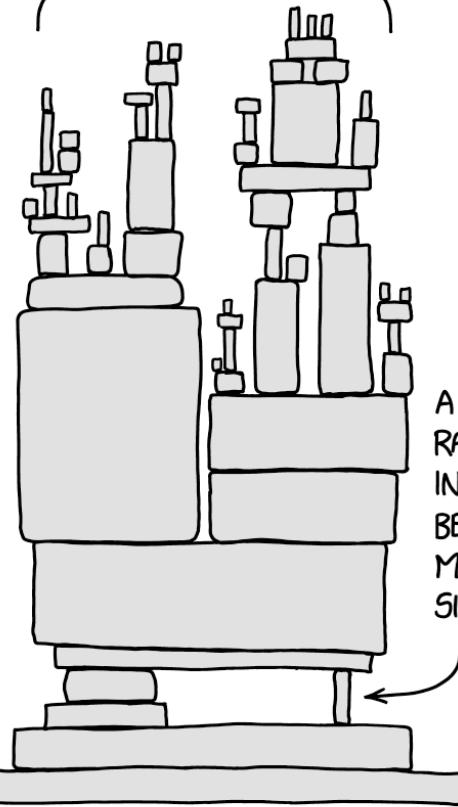
Open source, Check, Security, Check:

A checklist for securing open source projects

@iosifache

- Ex-builder at MutableSecurity
- Ex-security engineer @ Romanian Army and Canonical
- Security engineer in Snap Inc.
- Open source maintainer
- GSoC mentor for OpenPrinting
- Enthusiast of good coffee, long runs/hikes, and quality time

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



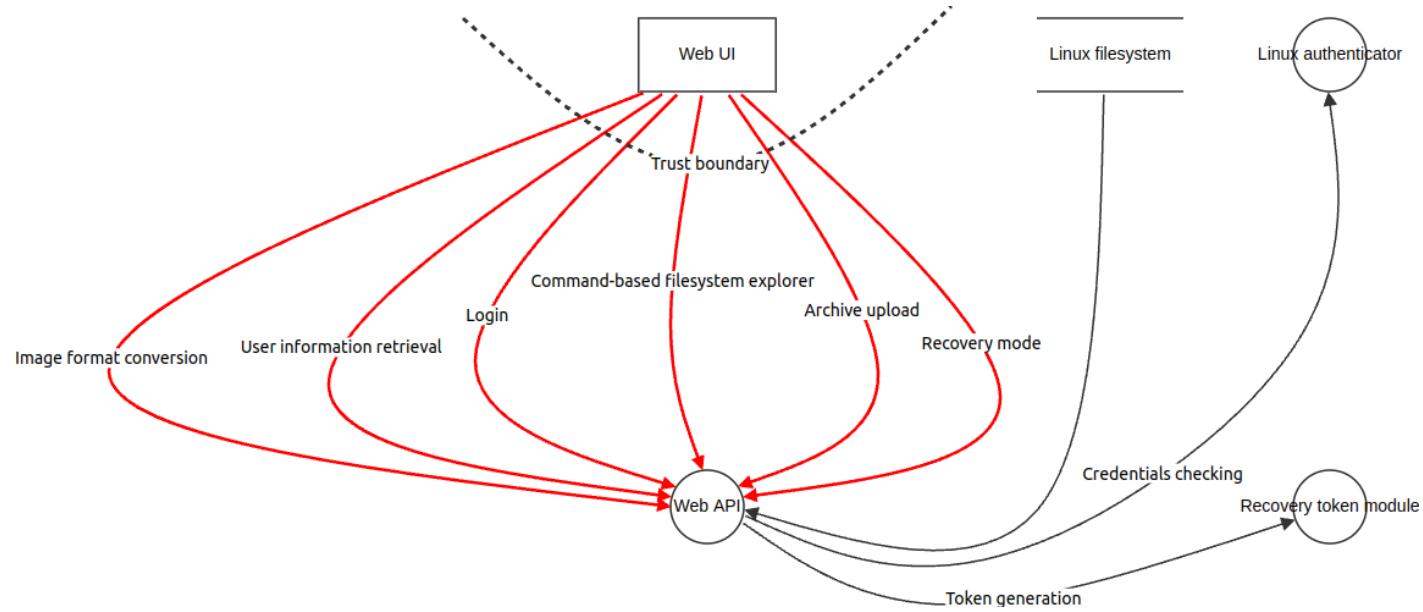
Notations

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   



2. Check for vulnerabilities in your dependencies  

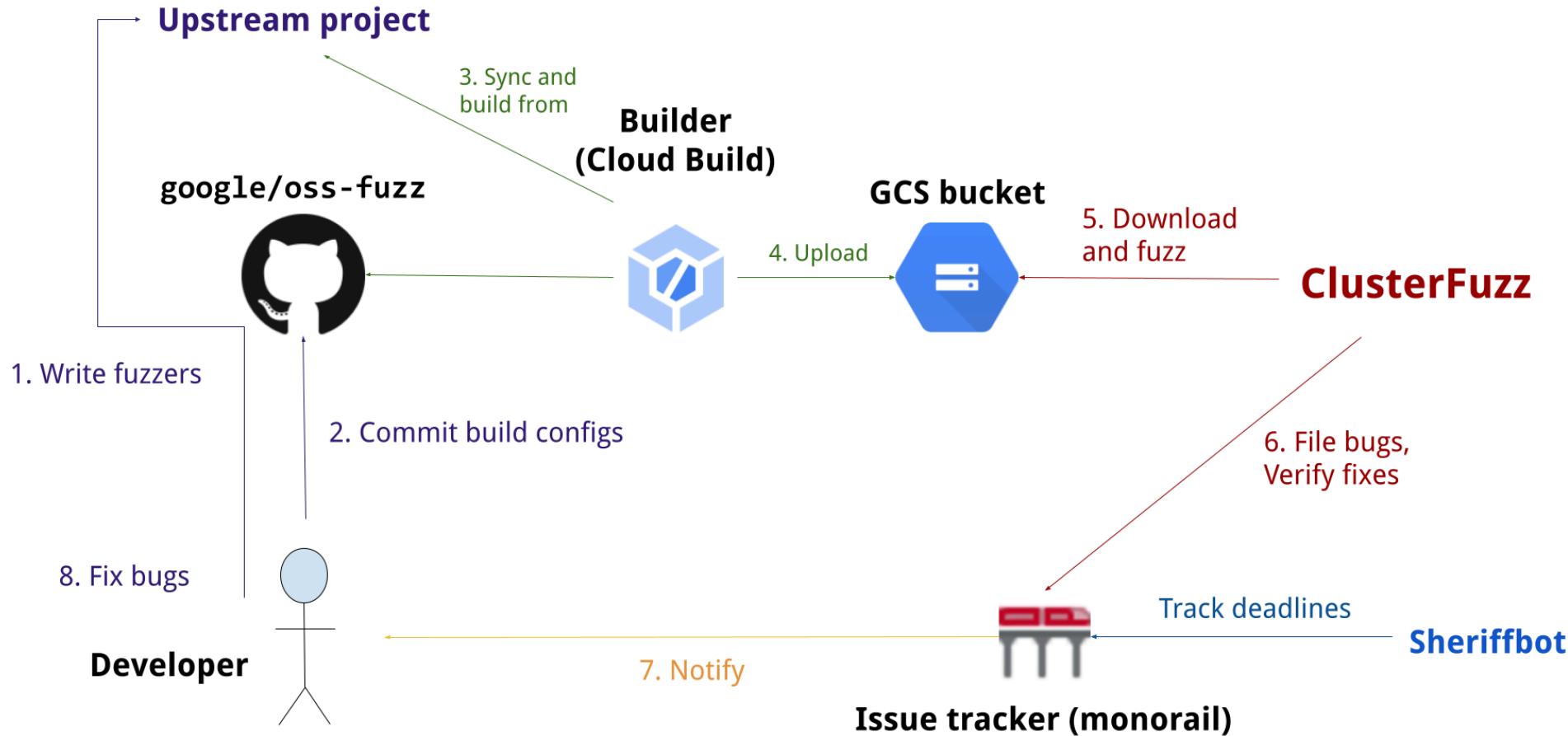


3. Run security tools and constantly validate the warnings   

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 ✓





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 protocol¹. This was a practical default in the past, when much of the web did not support HTTPS.

2. Have security recommendations for users   



Docs

v20.9.0 API LTS

v21.2.0 API

ES6 and beyond

Guides ARCHIVED

Dependencies

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 t ...

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 ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉

```
1  {
2   "$schema": "http://cyclonedx.org/schema/bom-1.4.schema.json",
3   "bomFormat": "CycloneDX",
4   "specVersion": "1.4",
5   "version": 1,
6   "metadata": {
7     "timestamp": "2023-11-16T11:26:04.636Z",
8     "tools": [
9       {
10         "vendor": "@cyclonedx",
11         "name": "cyclonedx-npm",
12         "version": "1.6.1",
13         "externalReferences": [
14           {
15             "url": "git+https://github.com/CycloneDX/cyclonedx-node-npm.git",
16             "type": "vcs",
17             "comment": "as detected from PackageJson property \"repository.url\""
18           },
19           {
20             "url": "https://github.com/CycloneDX/cyclonedx-node-npm#readme",
21             "type": "website",
22             "comment": "as detected from PackageJson property \"homepage\""
23           },
24           {
25             "url": "https://github.com/CycloneDX/cyclonedx-node-npm/issues",
26             "type": "issue-tracker",
27             "comment": "as detected from PackageJson property \"bugs.url\""
28           }
29         ]

```

III. Establish a security reporting process

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

[About](#)[Project Governance](#)[Previous Releases](#)[Security Reporting](#)

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 (selected) 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  

Security Team

Table of Contents

- [Node.js Bug Bounty Program](#)
- [Current Initiatives](#)
- [Current Project Team Members](#)
- [Emeritus Members](#)
- [Code of Conduct](#)
- [Moderation Policy](#)

This team is *not* responsible for managing or responding to security reports against Node.js itself. That responsibility remains with the [Node.js TSC](#).

Node.js Bug Bounty Program

The program is managed through the HackerOne platform at <https://hackerone.com/nodejs> with further details.

Current Initiatives

Initiative	Champion	Status	Links
Automate Security release process	@marco-ippolito / @RafaelGSS	In Progress	Issue #860
Node.js maintainers: Threat Model	Group effort	In Progress	Issue #1333
Audit build process for dependencies	@mhawson	TODO	Issue #1037

Current Project Team Members

- [fraxken](#) - Thomas Gentilhomme
- [marco-ippolito](#) - Marco Ippolito
- [mhawson](#) - Michael Dawson
- [RafaelGSS](#) - Rafael Gonzaga
- [ulisesGascon](#) - Ulises Gascon

4. Be transparent and verbose with the reported vulnerabilities  

Vulnogram CVE CVSS 4.0

NEW Open Download Post to CVE.org

CVE-yyyy-nnnn.. Load

Editor 7 Source Preview CVE Portal

CVE ID * CVE-yyyy-nnnn or pick from existing

Title eg., Memory leak in Linux Filesystem

Public at 24/08/2024, 12:30 GMT+2

Problem types

eg., CWE-20 Improper Input Validation

+ Problem type

Impacts

eg., CAPEC-130 Excessive Allocation

+ Impact

Affected products *

Enter a vendor and product OR a package and a collection

Vendor or project eg., Linux	Product name eg., Linux Kernel	Platforms eg., x86, Android, Windows, MacOS, ..		
Package collection URL eg., https://wordpress.org/plugins	Package name eg., kernel	Source repository (OSS) eg., https://git.kernel.org		
Modules, components, or features eg., filesystem	Source-code file (OSS) eg., hello.c	Program routines (OSS)		
Versions (exact versions or ranges)				
Affected? <input checked="" type="checkbox"/> y <input type="checkbox"/> n <input type="checkbox"/> ?	Version (or start of a range) eg., 1.2.0; 0 means no lower limits	< Version (range) eg., 1.2.8, 1.2.*	<= Version (range) eg., 1.2.7, 1.2.*	status changes (patches, split ranges)
+ Version		+ item		versionType eg. semver, mav
Default status (for versions not specified above) <input checked="" type="checkbox"/> y <input type="checkbox"/> n <input type="checkbox"/> ?				
+ Product				
CVE Description Auto Generate *				



The Open Source 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](https://github.com/iosifache/oss_fortress) as a GitHub repository
- ossfortress.io as a wiki





Say hi!