



The World's Identity Company



Managing Open Source Software Security in Your Organization

José Carlos Chávez
Security Software Engineer @ Okta

© Okta and/or its affiliates. All rights reserved.

**Secure Software by
Design 2024**

José Carlos Chávez

Security Software Engineer – **Okta**

- Peruvian
- Open Source enthusiast
- OWASP Coraza WAF Co-leader
- Loving father of 2
- Mathematician in quarantine



© Okta and/or its affiliates. All rights reserved.



@jcchavez

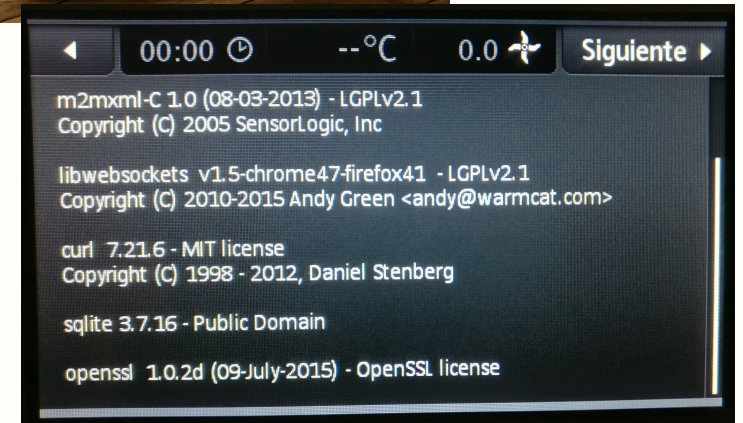
okta

Open Source



Why do we use Open Source Software in the enterprise?

1. It is free and publicly available.
2. It is flexible and general purpose.
3. It is (usually) stable.
4. Fosters ingenuity, creativity and innovation.
5. In many cases, it comes with a built-in community that brings support and continuously improves the source code.
6. It has shared maintenance costs across active users.
7. It is the future.



Open Source in numbers

Ecosystem	Total projects	Total project versions	YoY download growth
Java (Maven)	557K	12.2M	25%
Javascript (npm)	2.5M	37M	18%
Python	475K	4.8M	31%
.NET (NuGet Gallery)	367K	6M	43%
Totals/Averages	3.9M	60M	33%

Software Supply Chain Statistics, 2023
Sonatype 9th Annual State of the Software Supply Chain

96%

of the total codebases contained open source

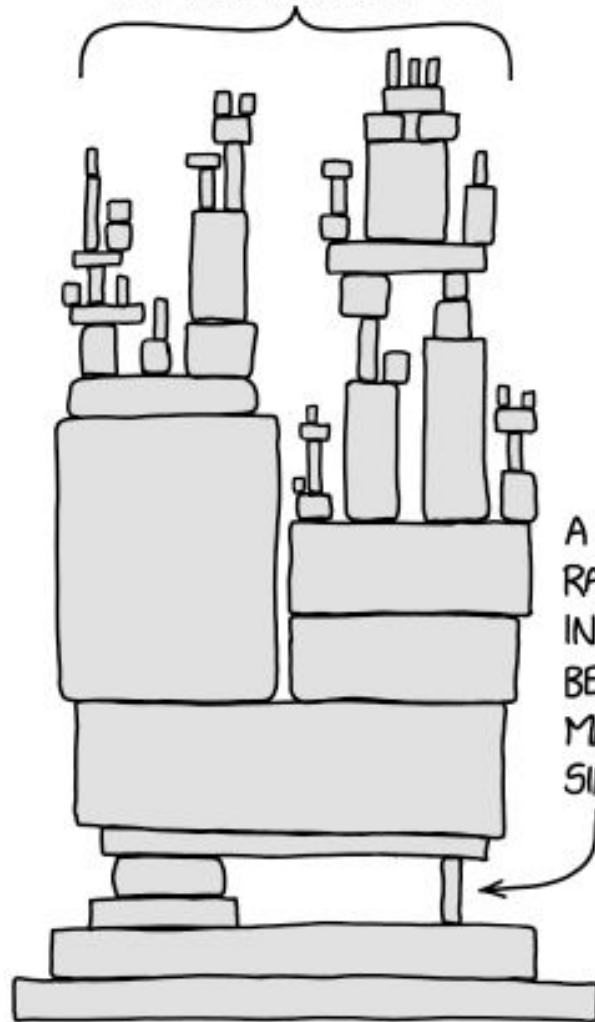
77%

of all code in the total codebases originated from open source

Synapsys OSSRA 2024



ALL MODERN DIGITAL
INFRASTRUCTURE



Open Source in numbers

Ecosystem	Total projects	Total project versions	YoY download growth
Java (Maven)	557K	12.2M	25%
Javascript (npm)	2.5M	37M	18%
Python	475K	4.8M	31%
.NET (NuGet Gallery)	367K	6M	43%
Totals/Averages	3.9M	60M	33%

Software Supply Chain Statistics, 2023
Sonatype 9th Annual State of the Software Supply Chain

Year	2024	2023	2022	2021
Q4	TBA	7,876 (+26%)	6,231 (+20%)	5,200
Q3	TBA	6,936 (+8%)	6,448 (+16%)	5,541
Q2	TBA	7,134 (+12%)	6,364 (+27%)	5,005
Q1	8,697 (+24%)	7,015 (+17%)	6,015 (+36%)	4,415
TOTAL	TBA	2,8961 (+15%)	25,059 (+24%)	20,161

Published CVE Records per year from cve.org

84% of codebases contained at least one open source vulnerability

54% increase in codebases containing high-risk vulnerabilities in the past year

Synapsys OSSRA 2024



OWASP OSS Top 10 risks

<https://owasp.org/www-project-open-source-software-top-10/>

OR1 – Known Vulnerabilities

A component version may contain vulnerable code

OR2 – Compromise of Legitimate Package

Attackers may compromise resources to inject malicious code

OR3 – Name Confusion Attacks

Attackers may create components whose names resemble names of legitimate component

OR4 – Unmaintained Software

A component or component version may not be actively developed

OR5 – Outdated Software

A project may use an old, outdated version of the component



OWASP Top 10 risks

<https://owasp.org/www-project-open-source-software-top-10/>

OR6 – Untracked Dependencies

Project developers may not be aware of a dependency on a component at all.

OR7 – License Risk

A component or project may not have a license at all.

OR8 – Immature Software

An open source project may not apply development best-practices.

OR9 – Unapproved Change

A component may change without developers being able to notice.

OR 10 – Under/over sized Dependency

A component may provide very little functionality or a lot of it.



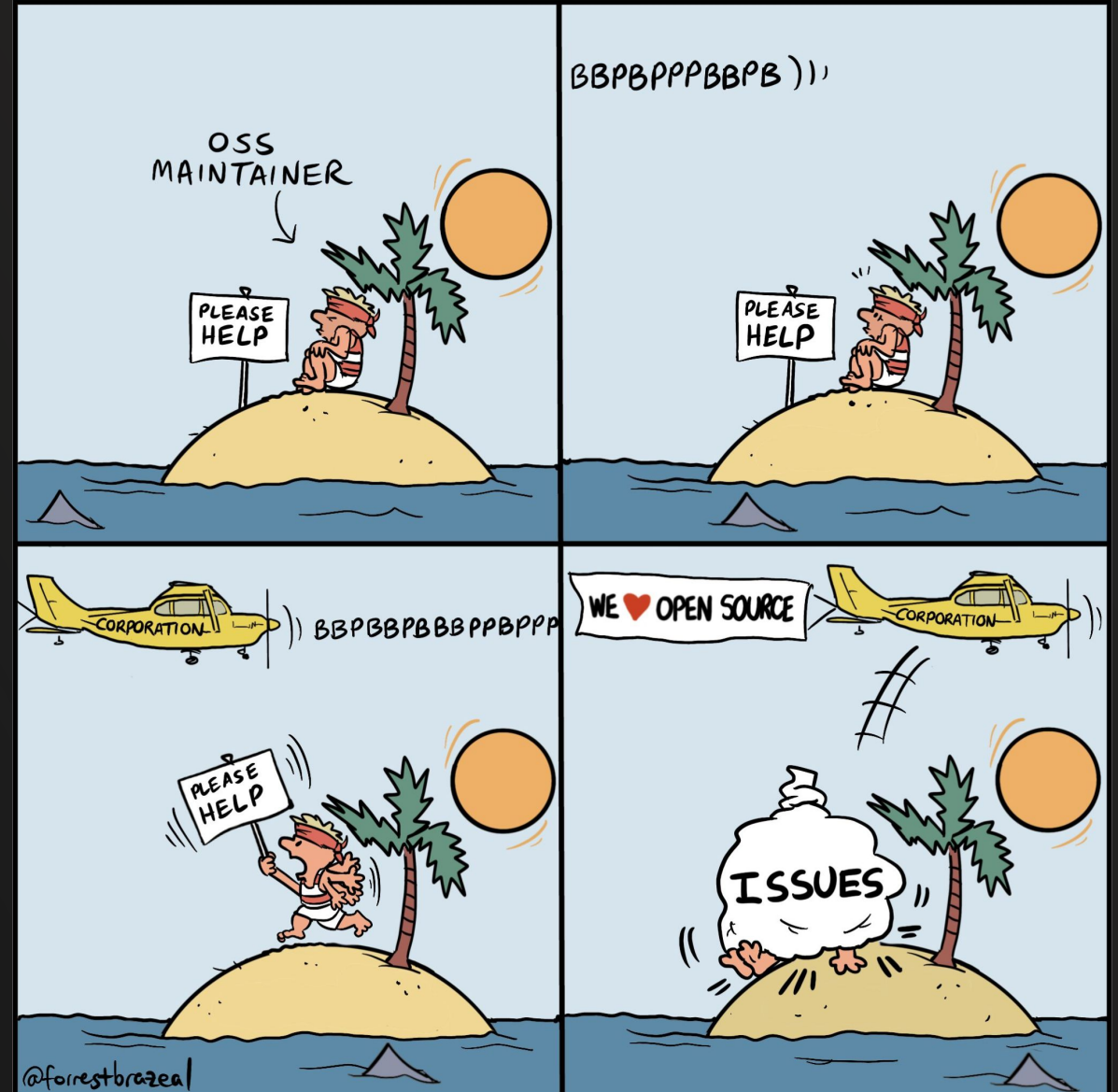
Strategies for protection

A. Maintain an inventory of Open Source Software in use:

1. Define policies on how to choose a package and monitor its health over time e.g. [OSSF scorecard](#).
 - Explicit policies in favour of ***Do it yourself*** and against ***Don't reinvent the wheel*** are important.
2. Automated analysis of Supply Chain to identify dependencies and build dependency graphs.
3. Maintain up to date SBOMs.
4. Get involved with crucial projects being used in your products.

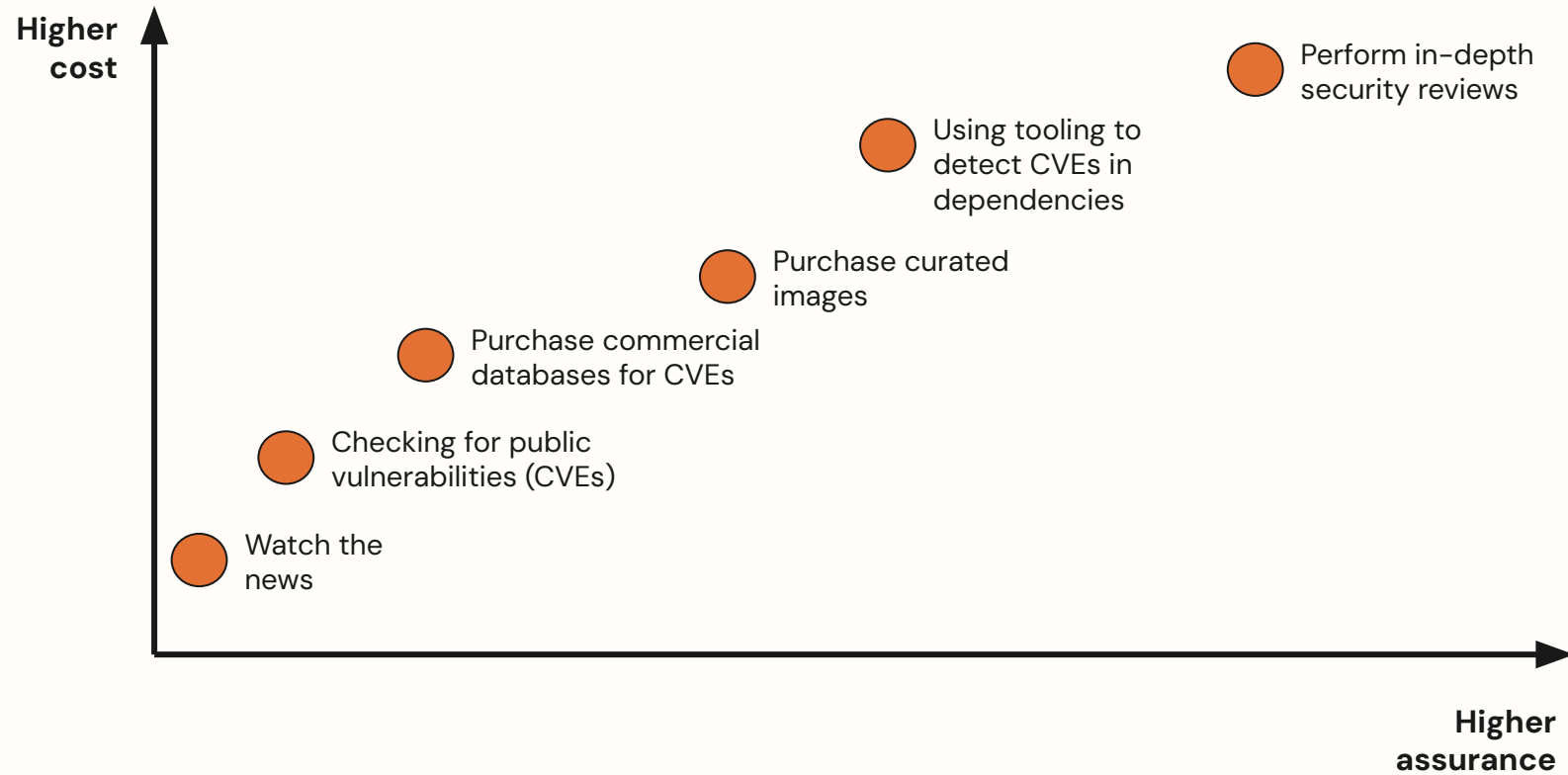


Don't be that company



Strategies for protection

B. Ensure OSS is secure



Strategies for protection

B. Ensure OSS is secure: **In depth security reviews**

1. Public vulnerabilities
2. Static analysis
3. Dynamic analysis
4. Secure configuration
5. Project health
6. Active monitoring of all the outcomes



Strategies for protection

C. Respond to Security Vulnerabilities

1. **Routinary ones (99%):** teams get alerted when they are found and a manual/automated fix is put in place.
2. **Intrincated ones (1%):**
 - requires triage and identify affected components
 - rollout mitigation if possible e.g. Web Application Firewall, Rate limiting, etc.
 - coordinate a solution
 - rollout a solution (could take weeks e.g. due to breaking changes or deprecations)



Conclusions

Keep these tips in mind to help craft a more engaging presentation.

1. Keep an accurate inventory of the open source being used – automation is essential.
2. Use high-quality vulnerability data sources (not only CVEs)
3. Leverage existing processes to respond to OSS vulnerabilities
4. Do not rely on a single layer of defence
5. Building your program will take time and iterations.
6. Keep a closer eye on the high-risk OSS being used.
7. Consider adopting practices that keep OSS components up to date – again, automation is essential.
8. Stay on top of the open source supply chain – monitoring is required.



Questions?

You can also reach me at

- josecarlos.chavez@okta.com
- <https://www.linkedin.com/in/jcchavez/>
- <https://twitter.com/jcchavez>



Recommended readings

[Threats, Risks, and Mitigations in the Open Source Ecosystem](#) – Michael Scovetta, Microsoft

[2024 Open Source Security and Risk Analysis Report](#) – Synopsis

[9th Annual State of Software the Supply Chain](#) – Sonatype

[CERT/CC Vulnerability Notes Database](#) – CMU SEI

[Why the future of manufacturing will rely on open source](#) – Francis Chow, Redhat

[Top Cybersecurity Statistics for 2024](#) – Jacob Fox, Cobalt



Thank you!

