

# The Lazy Guide to Open Source Software (OSS)

## Risk Assessment

Cyberface says “*Managing Risk is the best!*”



# CONTENT

- WHO IS THIS GUY?
- RISK MANAGEMENT ESSENTIALS
- WHAT IS OUR PROBLEM STATEMENT?
- YOU ARE DOING IT WRONG
- CONCLUSIONS

Astroface says  
“ALL SYSTEMS GO! ....for ASSESSING RISK!!!”

# WHO?



**CRob**, n, adj, and v

Pronunciation: U.S. (K-robe)

43rd level Dungeon Master

26th level Securityologist

Security Lorax, Cat-herder

Pirate-enthusiast & hat-owner



CRob\_at\_OpenSSF\_dot\_org



@SecurityCRob



@SecurityCRob@infosec.exchange



<https://github.com/SecurityCRob>



[The Security Unhappy Hour,  
Chips & Salsa](#)



[What is in the SOSS?  
https://www.linkedin.com/in/darthcrob/](#)



CyberFace wants to know when the eXceptional Reception is? He only came for the treats.

## The Rising Tide lifting all boats

- The OpenSSF is a **cross-industry collaboration** that brings together leaders to **improve the security** of open source software (OSS) by building a broader community, targeted initiatives, and best practices
- The OpenSSF brings together open source security initiatives under one foundation to accelerate work through cross-industry support. This includes working groups that address vulnerability disclosures, security tooling and more.
- OpenSSF is **committed to collaboration** and working both upstream and with existing communities to advance open source security for all.



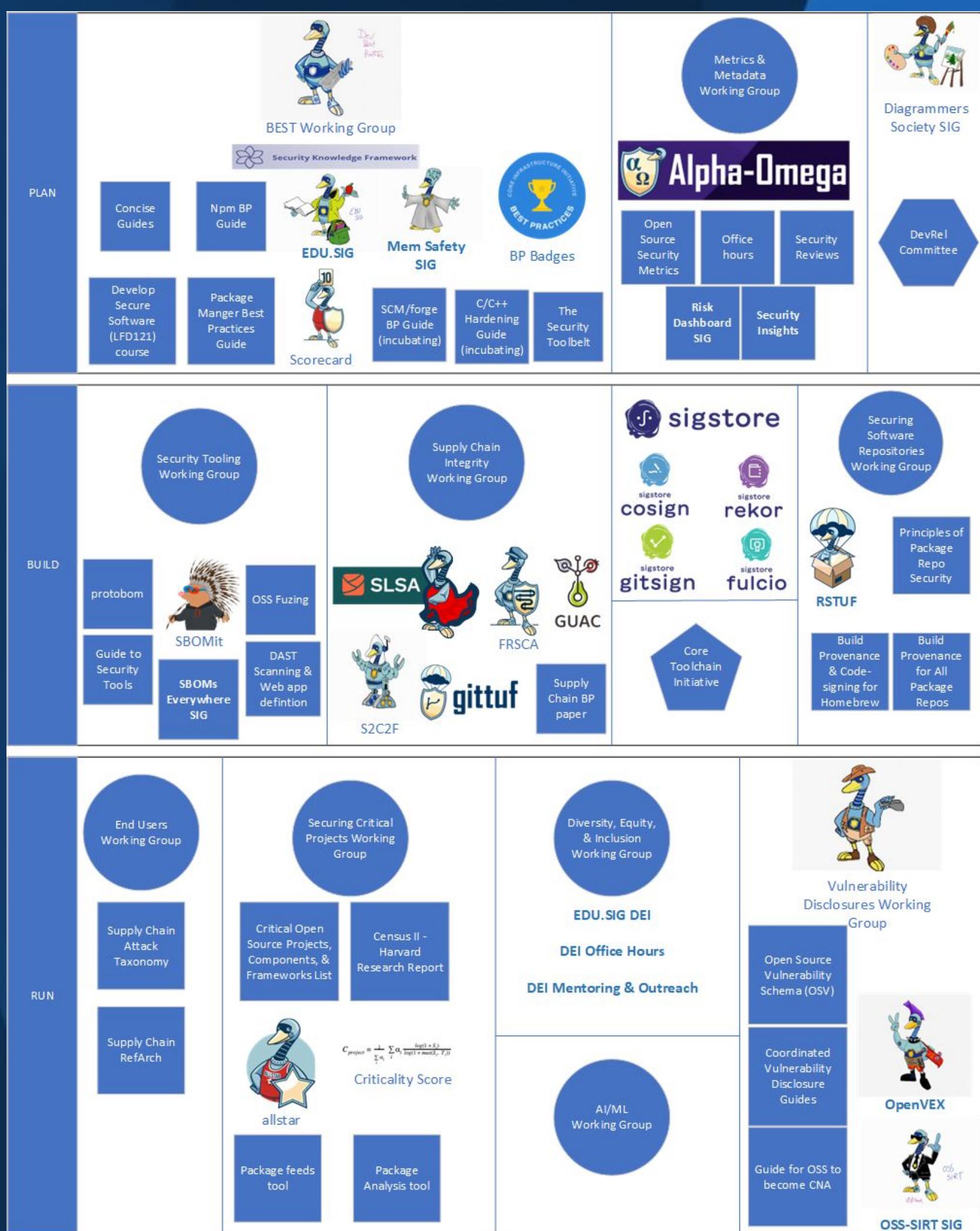


# A Gaggle of Geese!

We have numerous software projects, guidelines, specs, and experts to help both upstream OSS developers AND downstream OSS consumers

<https://openssf.org/community/openssf-working-groups/>

What's up with all the Geese?



WAT

# TALKING ABOUT RISK FOR FUN AND PROFIT

SPOILERS - it's ALL about the business!

CyberFace says "Wordz matter!"

# WHY?

Risk Management's purpose is ensure effective & efficient strategy & strategic decisions so that the organization can deliver on their desired outcomes. The primary tool for providing this input to the strategy is risk assessments.

Risk-taking is an intrinsic component of business.



CyberFace says “*We all contribute to that risk management culture!*”

A **risk** is a possibility of damage or harm. A risk is described by a **threat** and a **vulnerability**. It is often documented as:

$$RISK = Threat \times Vulnerability$$

A **threat** is the potential cause of an incident that may result in harm to a system or organization.

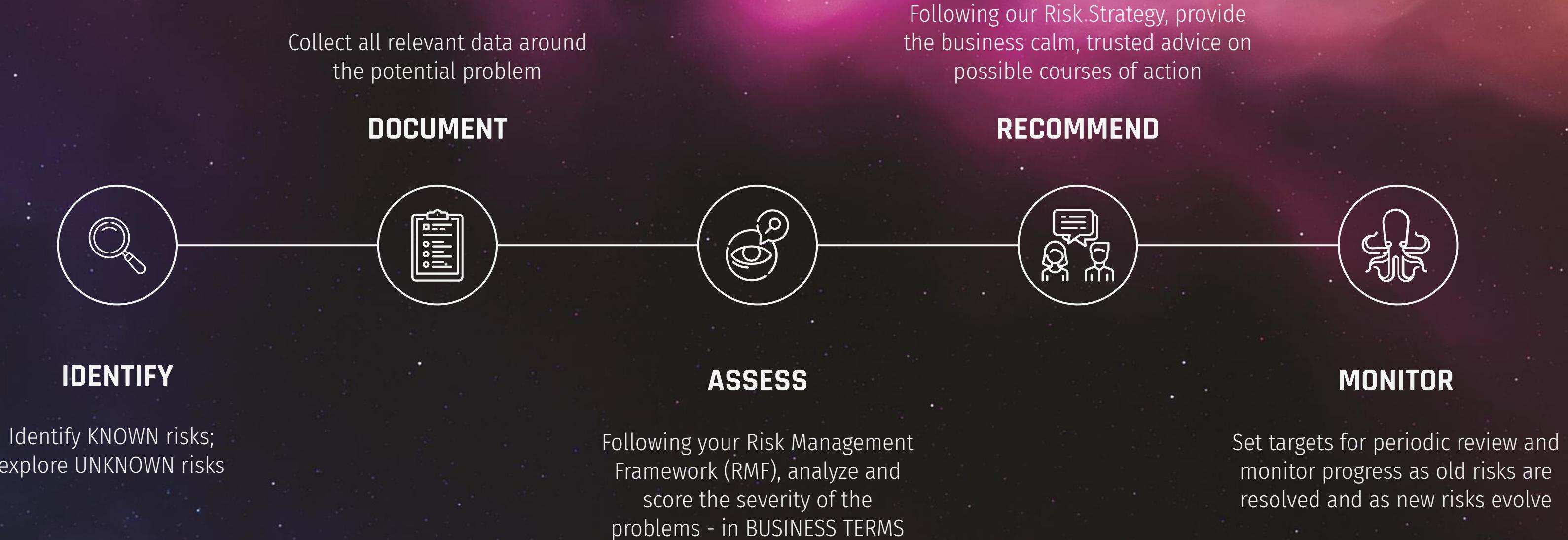
## WHAT IS RISK?

A **vulnerability** is a weakness of software, hardware, or online service that can be exploited and that has security implications.

It can also be described in terms of **Impact** and **Likelihood**

$$RISK = Likelihood \times Impact$$

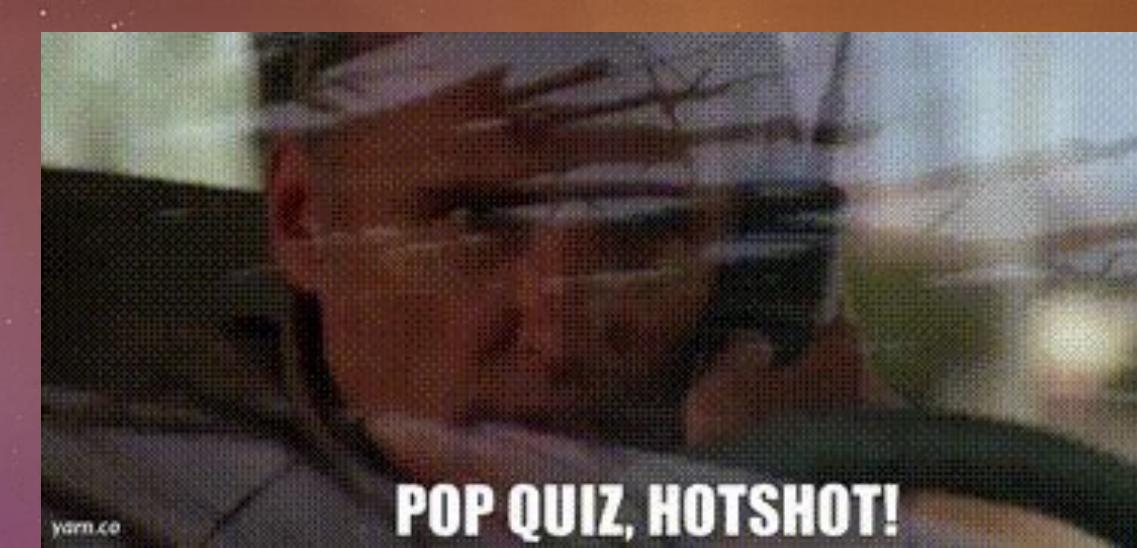
# RISK MANAGEMENT GOALS



# POP! Quiz, hotshots

How many folks here **use** open source software?

(hint - do you own a smartphone or use the internet/cloud?)



**POP QUIZ, HOTSHOT!**

How many folks here **CONTRIBUTE** to open source software?

How much **money** do you think your company has **made/saved** because you decided to use **someone else's software** instead of writing it yourselves?

CISA estimates that OSS has generated over **\$6-8 TRILLION** dollars of value to the global economy.

Image [Source](#)

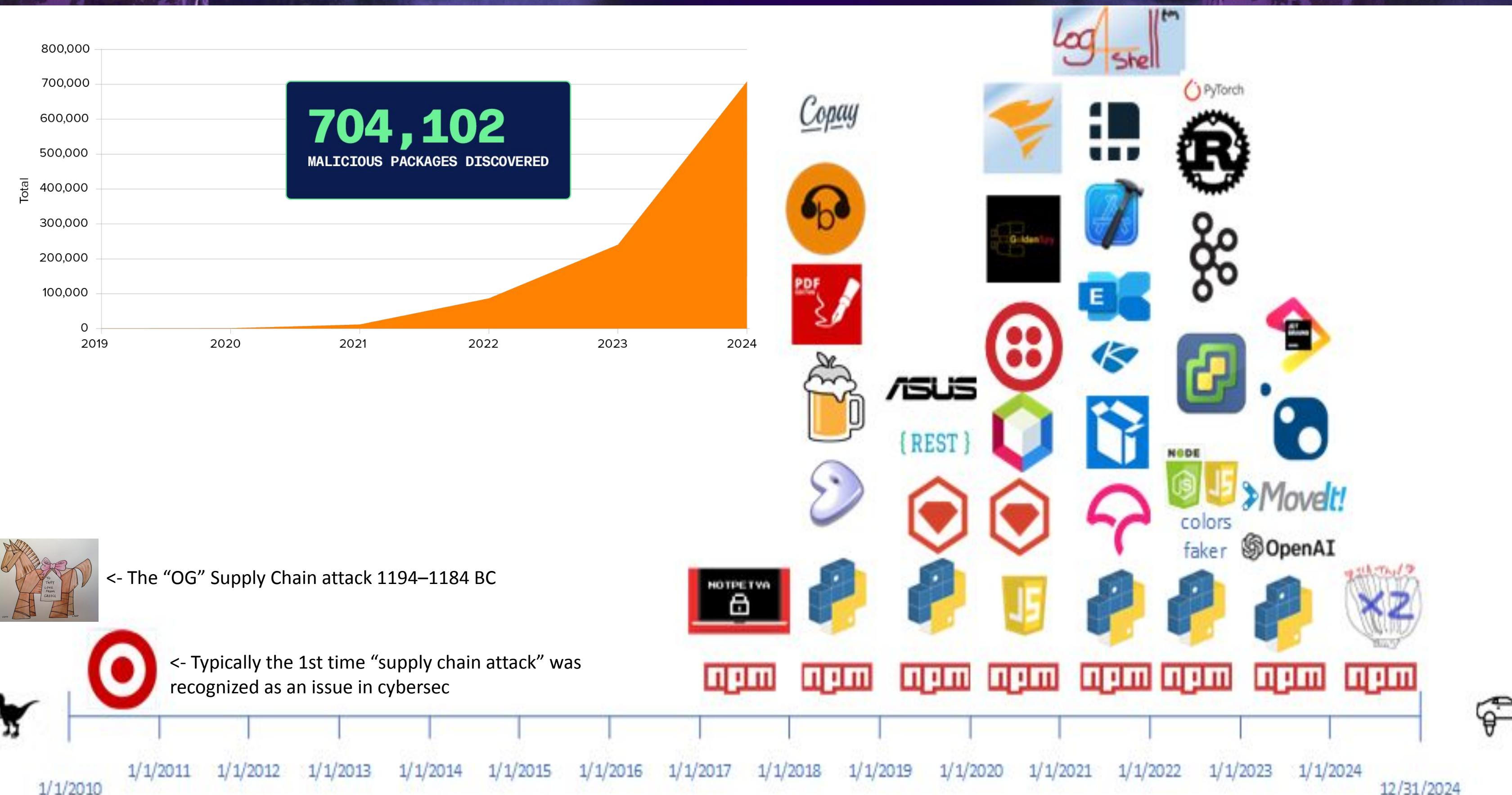
# Problem Statement

- Studies cite that anywhere between 90-98% of commercial software contains open source software(1)(2)
- Supply Chain attacks have risen 450%-600% over the last several years(3)
- Most Suppliers and Downstream Consumers do not understand upstream open source



- 1.) <https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html>
- 2.) <https://www.linuxfoundation.org/blog/blog/a-summary-of-census-ii-open-source-software-application-libraries-the-world-depends-on>
- 3.) <https://www.csoonline.com/article/3677228/supply-chain-attacks-increased-over-600-this-year-and-companies-are-falling-behind.htm>

# A Tour of many “popular” open source & supply chain attacks over the years



<https://www.sonatype.com/resources/vulnerability-timeline>

---

<https://www.reversinglabs.com/blog/a-partial-history-of-software-supply-chain-attack>

<https://linuxfoundation.eu/newsroom/the-rising-threat-of-software-supply-chain-attacks-managing-dependencies-of-open-source-projects>

# Outside of direct supply chain attacks, vuln counts have risen over the years

CVEs per year





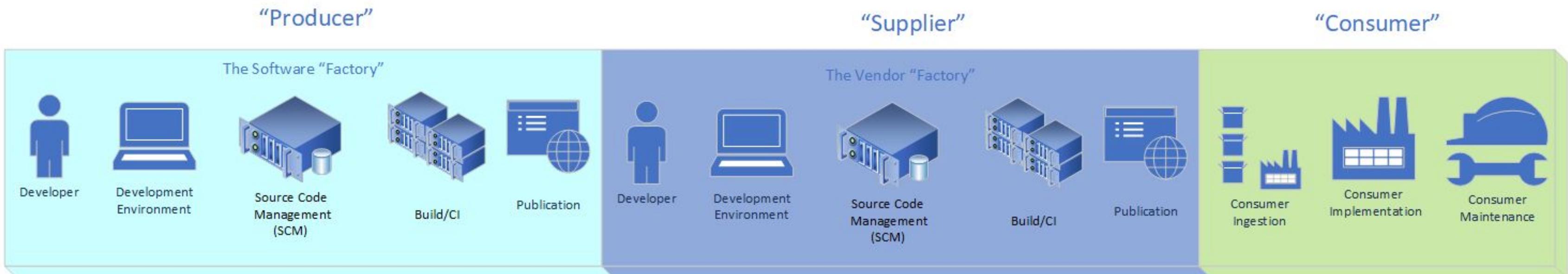
Let's Talk about  
what you're doing  
wrong



“I am an organization that pulls code straight from upstream”



“I am an organization that has my software supplied by a vendor”



# You never checked who wrote your software

Was it one person?  
Was it many people?

Did your vendor ACTUALLY write it,  
or did they embed Open Source  
into it?

IF they did that, what did they add in?



\*A CORE OSS tenant is that “all patches are welcome”, regardless of person/nation of origin

Does that person(s) understand software engineering?

Do they understand SECURE application development?

SOME ORGANIZATIONS may also care about country of origin\*

Image [Source](#)

## RISK:

3rd party software your company uses has unknown or unverified components that are of unknown pedigree, provenance, quality, and origin used within your operations and prod environment

### How this RISK can be managed:

- ONLY acquire software from known, vetted & trusted sources! (Pastebin is ....just the worst)
- Require your suppliers to give you an electronic, machine-readable **Software Bill of Materials** (SBOMs) that does multiple levels of dependency checking to see Pedigree and Provenance of the code as it travels through the software supply chain
- Use tools like OpenSSF Scorecard, which provides critical security health metrics about OSS projects -
- OpenSSF Concise Guide to Evaluating Open Source Software
- OpenSSF Secure Supply Chain Consumption Framework (S2C2F)
- 



Cyberface says “Did you know there are multiple SBOM types (Design, Source, Build, Analyzed, Deployed) as well as multiple electronic formats (SPDX, CycloneDX) to generate them in (and yes, there are some vendors that will default to FAXING you a PDF)...be warned”

# Brief SBOM tangent.....



<https://www.youtube.com/watch?v=sNjVQaK5QW4>

Cyberface says “Say ‘SBOM’ three times and Alan will appear!”

# You never checked that what they wrote is actually what you installed

Predominantly, OSS is about the source code, which is publicly available to view, test, reuse

How the software is compiled, tested, composed, built, and published is JUST as important as the software itself

\*When you deploy your code in production without testing



You will run into two of the hardest problems in all of computers:  
Naming & Versioning

Depending on how you consume software, there are things you can do to ensure integrity/authenticity

# What could **possibly** go wrong?

AV-100 - Develop & Advertise Distinct Malicious Package from scratch

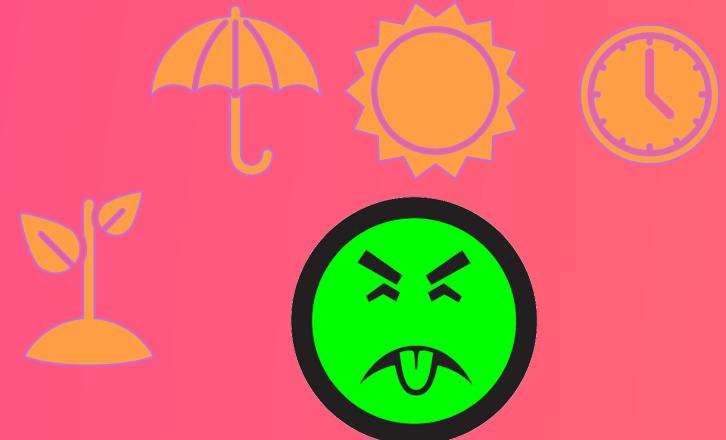
AV800 - Become the Maintainer



Moira  
Developer Queen



AV-302 Contribute as Maintainer  
AV-700 Compromise Maintainer System



AV-200 - Create Name Confusion with Legitimate Package

AV-403 - Tamper with Exposed Build System



AV-001 - Subvert Legitimate Package



AV-500 - Distribute Malicious version of Legitimate Package



AV-701 Exploiting Config Vuln  
Counterfeiting  
AV-702 Exploiting Software Vuln

...quite a lot, actually

## RISK:

3rd party software your company uses within your operations and prod environment has unknown or unverified build, test, and publication information

### How this RISK can be managed:

- Understand how the software you use is compiled, assembled, tested, and published
- Only used signed software artifacts (SBOM, packages, etc.) from trusted sources
- OpenSSF [Supply Chain Levels of Software Artifacts \(SLSA\)](#)
- OpenSSF [sigstore](#) for digital signing for integrity/authenticity checking
- Scanning, scanning, scanning
  - SAST, DAST, SCA, Fuzzing, dependency checking, known vuln scanning, etc.
- OpenSSF [C/C++ Compiler Hardening Options Guide](#)
- OpenSSF [Source Code Management Best Practices Guide](#)
- OpenSSF + CISA [Principles of Package Repository Security](#)
- OpenSSF [Concise Guide to Evaluating Open Source Software](#)

Cyberface says “*Sometimes HOW something is made is JUST AS important as what it does.*”



# You never checked how they secure the software (or maybe they don't)

FUN FACT - The majority of OSS software projects are maintained by a single maintainer

A large portion are “weekend warriors” that don’t have tools, resources, or peers to assist with the maintenance of the project



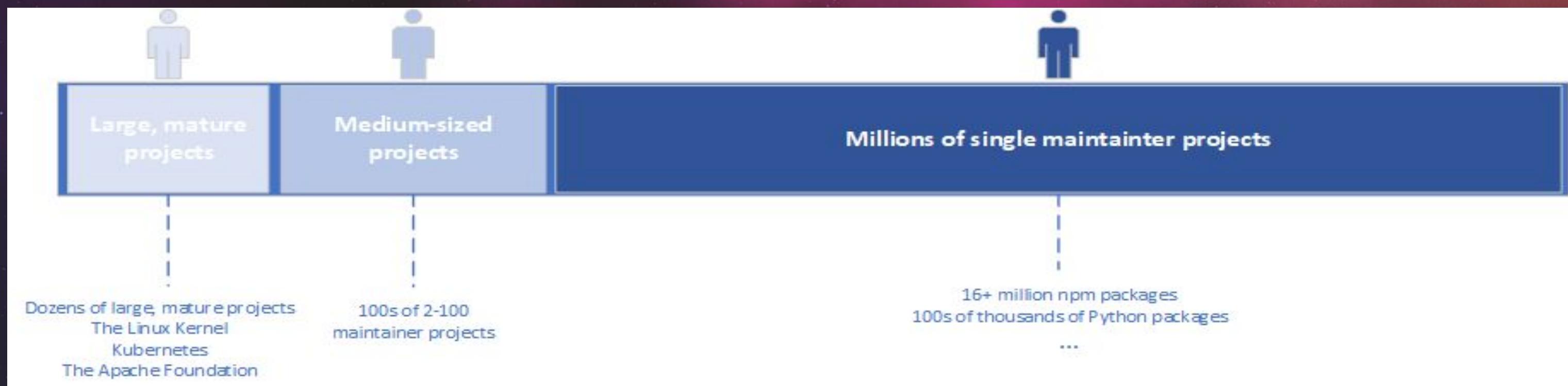
To learn more about Diana the Weekend Warrior:

<https://github.com/ossf/toolbelt/tree/main/personas#name-diana-the-weekend-warrior>

Depending on how you consume software, there are things you can do to ensure integrity/authenticity

# Open Source Developer Spectrum

- There are MANY motivations for why people choose to create and collaborate with open source software (1).
  - Many large and mature open source projects have specialists or teams that help manage security for the effort.
  - The majority of open source projects are single-maintainer projects that do not have access to those resources, nor an understanding of how (2).



- (1) - [https://www.linuxfoundation.org/hubfs/LF%20Research/MaintainerSecurityBPs\\_011724.pdf?hsLang=en](https://www.linuxfoundation.org/hubfs/LF%20Research/MaintainerSecurityBPs_011724.pdf?hsLang=en)
- (2) - <https://anchore.com/blog/open-source-is-bigger-than-you-imagine/>

## RISK:

3rd party software your company uses within your operations and prod environment was created and is maintained with unknown SDLC practices

### How this RISK can be managed:

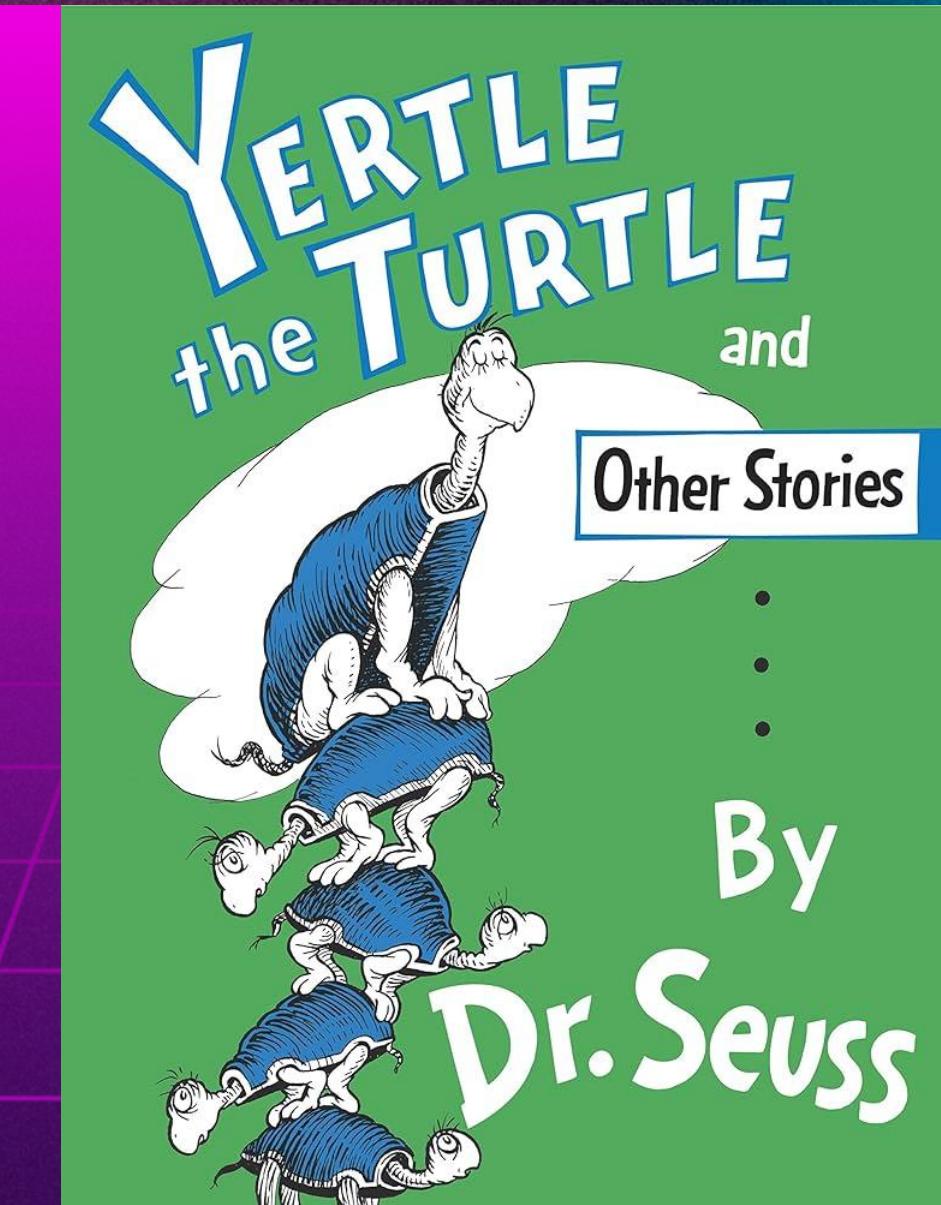
- For software that matters to you, find out what the project's development processes, tooling, lifecycle activities are
- Check for the [OpenSSF Best Practices Badge](#), which provides verification of project's SDL activities
- OpenSSF [Concise Guide to Developing More Secure Software](#)
- OpenSSF [Secure Supply Chain Consumption Framework \(S2C2F\)](#)

Cyberface says “*Education is the SINGLE greatest tool in a secure developer’s toolkit. Once you learn something, you ALWAYS have access to that knowledge!*”



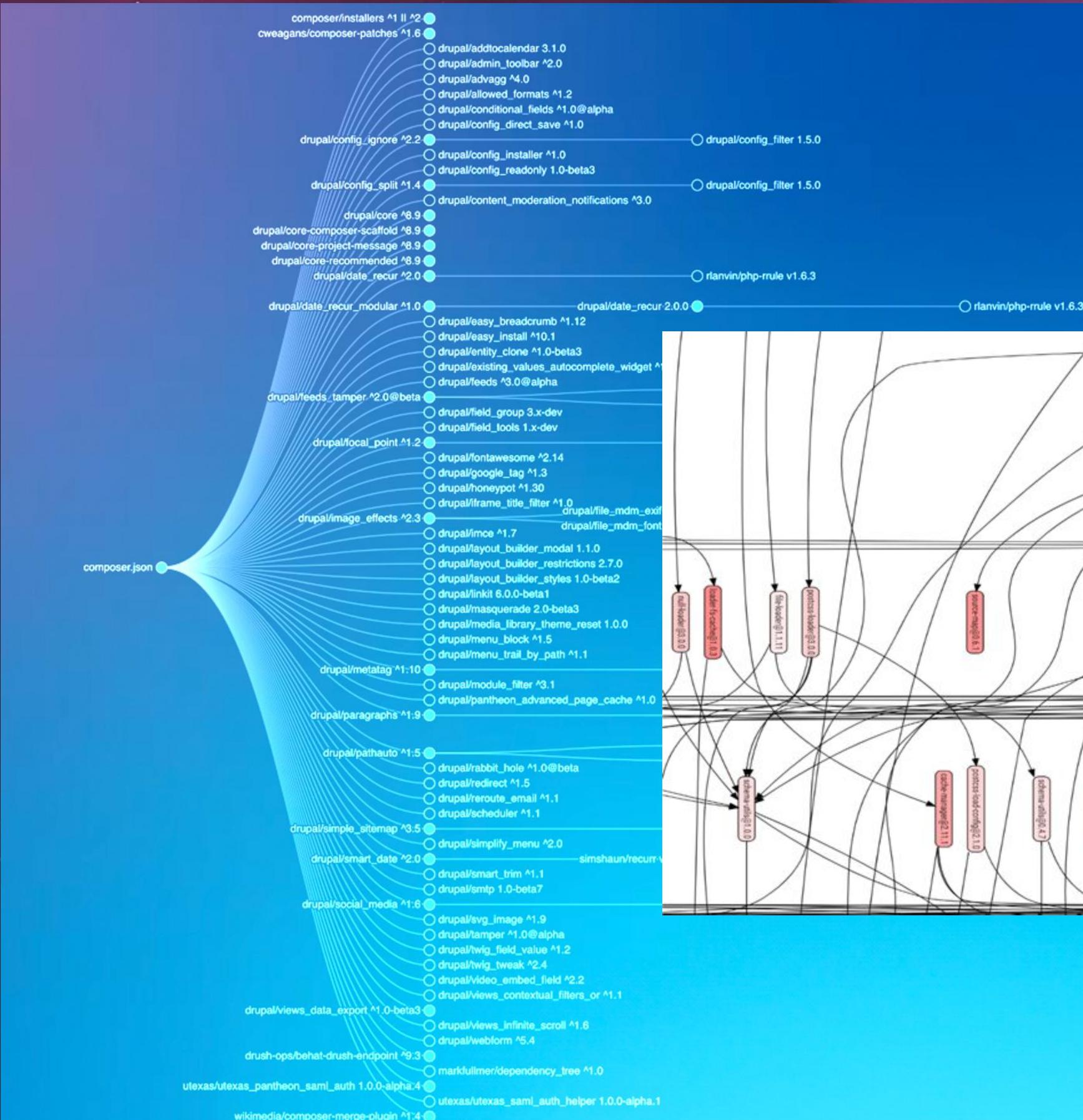
# You don't really know what software you have installed or what it installed to make itself work

Depending on the language and ecosystem, that ONE component you ingest could have 1 to  $n$  dependencies just to make it work



FUN FACT - oftentimes your dependencies ALSO have dependencies that may not get documented (pulled in at build or runtime)

# ....So many dependencies



[https://www.drupal.org/project/composer\\_dependency\\_tree](https://www.drupal.org/project/composer_dependency_tree)

<https://www.gatsbyjs.com>

Thanks to Intel's Ryan Ware & Katharine Druckman for letting me open source these images!

....so many

## RISK:

3rd party software your company uses within your operations and prod environment has unknown or unverified transitive and intransitive dependencies

### How this RISK can be managed:

- Reduce the attack surface of your software and eliminate unneeded dependencies wherever possible
- Use tools like the OpenSSF's [GUAC](#) (Graph for Understanding Artifact Composition) to show dependency tree graphs to analyze “hidden” or “unadvertised” software components -
- OpenSSF [Concise Guide to Evaluating Open Source Software](#)
- OpenSSF [Secure Supply Chain Consumption Framework \(S2C2F\)](#) - Documents levels of maturity to ease your journey of ingesting OSS and so that you can improve over time

[Image Source](#)



Astroface says “*Removing software dependencies and packages you don’t need ultimately makes less work for you in the future! Future-you will thank you!!*”

# You never checked that the dev made any kind of commitment to fix problems in the software

Free software doesn't always include free support and maintenance.

Upstream typically supports latest version and *maybe one version back* with updates.



COMPLETELY EXPECTED “ripped from the headlines” reference

Back to the Diana persona, many projects truly are “as is” with no commitment to fix defects or vulnerabilities.

Our “pal” Jia Tan of XZ-Utils fame, highlights a common pattern of upstream maintainer burnout being exploited by bad actors

# OSS Licensing “Fun Facts”

Here is a snippet from the Apache 2.0 License, which is broadly used, and very similar to other OSS licenses in these two paragraphs.

As always, **your mileage varies**, read the licenses for the software you consume/use for specifics.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, **Licensor provides the Work** (and each Contributor provides its Contributions) on an **"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND**, either express or implied, including, without limitation, any warranties or conditions of **TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE**. You are solely responsible **for** determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. **In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages**, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the **Work** (including but not limited to damages **for** loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

## RISK:

The 3rd party software you use within your operational and production environment has a different perspective on how the software is supported and maintained than what your requirements state.

### How this RISK can be managed:

- Ensure that software you consume have documented support policies that meet your organizational requirements. As needed, consider finding commercial support for open source packages critical to your business. **Better yet, consider contributing back upstream to projects you use**
- Check upstream repositories for a [security.md](#), security policy, or vulnerability disclosure policy (VDP) to understand how the project will address vuln reports (or if they won't!)
- This signal can be found in [Scorecard](#), [Best Practices Badge](#), [Allstar](#), [Security Insights](#) and others
- Check critical projects' historic advisories, git repo commits to understand how reports are addressed

Cyberface says “Actors like ‘Jan Tai’ have been conducting social engineering attacks like what happened with XZ-Utils for many years now. Originally the attackers would only embed malware and BitCoin miners, but now Nation State Actors see this as a prime attack pattern (Solarwinds, MS Exchange hack [Forest Blizzard], and many others). Enjoy!”



# You aren't monitoring upstream for CVEs, bug fixes, or other activity

Open Source Projects are like snowflakes....each one is delicate, beautiful, and completely different from every other one

There are multiple ways you can monitor upstream activity to get the information you need to manage your software



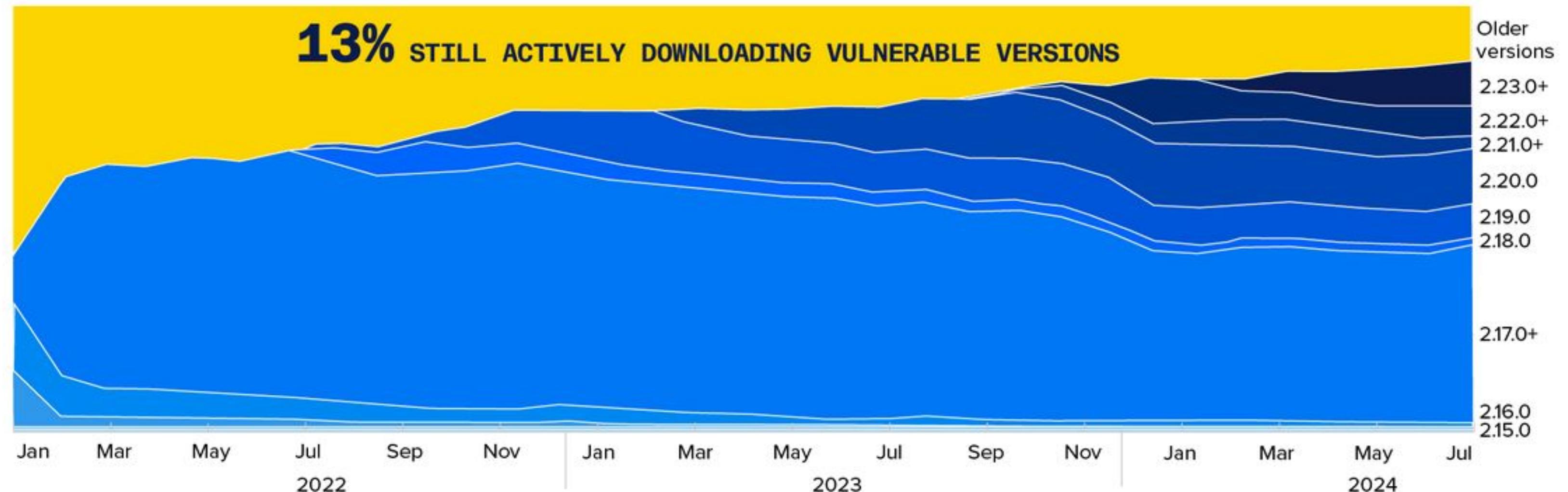
Also, software ages more like milk than wine.....

Another sad issue “ripped from the headlines” that is COMPLETELY avoidable...

**96% of vulnerable open source downloads had a newer, non-vulnerable version available**

Log4Shell was fixed  
on Dec 6, 2021

Monthly Log4j Downloads on Maven Central



Sonatype 10<sup>th</sup> Annual State of the Software

Brian Fox, CTO & Co-founder of Sonatype “**PLEASE stop  
downloading the old, vulnerable versions of log4j! Log4Shell has  
been fixed for years now!**”

<https://www.sonatype.com/resources/log4j-vulnerability-resource-center>  
<https://www.sonatype.com/state-of-the-software-supply-chain/Introduction>



## RISK:

You do not understand how the 3rd party software you use within your operational and production environment intakes, triages, prioritizes, fixes, and communicates security vulnerabilities and other functional updates and don't have it plugged into your SDLC practices.

### How this RISK can be managed:

- Find out how critical software you use communicates changes and updates. SUBSCRIBE to those channels. Add these signals to your SDLC to ensure integration.
- The OpenSSF's OSV (Open Source Vulnerability database and schema) is one of the best places to monitor for OSS vulnerabilities
- Databases like NVD may not have complete, accurate, or timely data about all known CVEs + their metadata
- oss-security lists are the BEST thing to monitor for traditional OSS projects as CVEs go PD
- Tools like VEX (or the awesome OSS' OpenVEX) help maintainers and vendors notify their downstream of the affectedness of their software to vulns quickly (VEX works GREAT with SBOMs!)
- OpenSSF SIREN (new!) is a new list that will help advertise loCs and TTPs for oss projects that are being actively exploited

Cyberface says “A good friend of mine once said that ‘Knowing is HALF the battle!’”



# CONCLUSION

Open Source Software is a vibrant and energetic source of ideas and innovation than can be used by downstream to power their dreams....

Using it is NOT FREE, and requires patience, discipline, and lots of additional hard work, including....

# Parting Thoughts to Manage Your Risk of Using OSS

1. Understand what OSS you are using today (through scans, SBOMs, code review, etc.); conduct due-dilligence to learn more about how it is written, stored, compiled, and published
2. Reduce your attack surface when it comes to # of installed packages and dependencies
3. Combine your asset inventories with SBOMs with vuln scanner info and VEX statements to get a better picture of where your OSS risk lies
4. Use the latest, patched versions of software. Old software = vulnerable software
5. Monitor your upstream suppliers and OSS projects for updates (particularly CVE notifications)
6. **CVSS != Risk** A number or colour is not Risk. You need to apply your org's Risk Appetites, Tolerances, and RM methodology to understand what vulns matter to YOU most
7. If you use OSS, consider supporting OSS

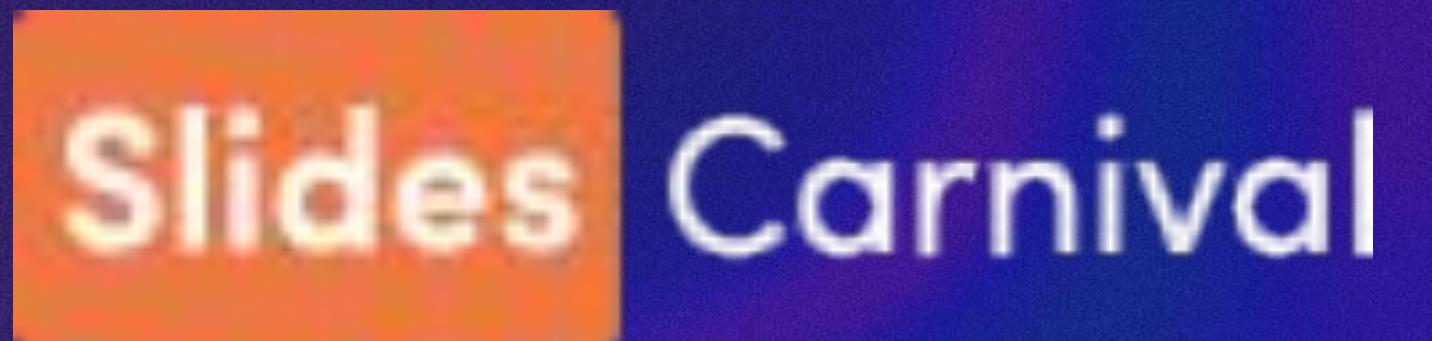
# THANK

# YOU!



CyberFace says “*This isn’t the end, this is only the BEGINNING! (mwahaha)*”

# CREDITS



This presentation template is free for everyone to use thanks to the following:

SlidesCarnival for the presentation template  
Pixabay, Pexels for the photos

