Risk of Open Source Software Components in Telecom

International Conference on 5G Network Security

09 August 2023

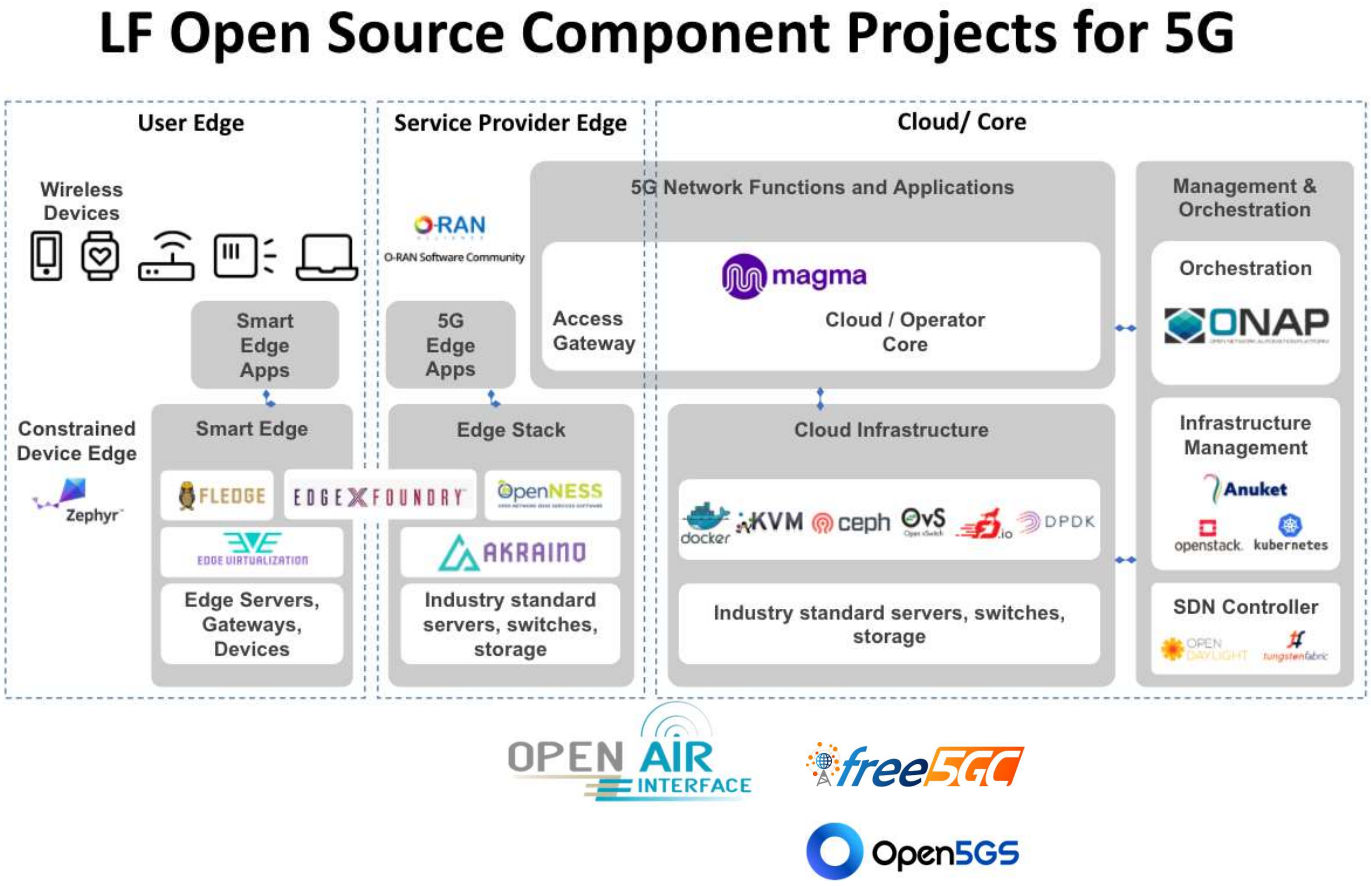
Aneesh Kumar K B

CDAC

Thiruvananthapuram

5G and Open Source Software Landscape

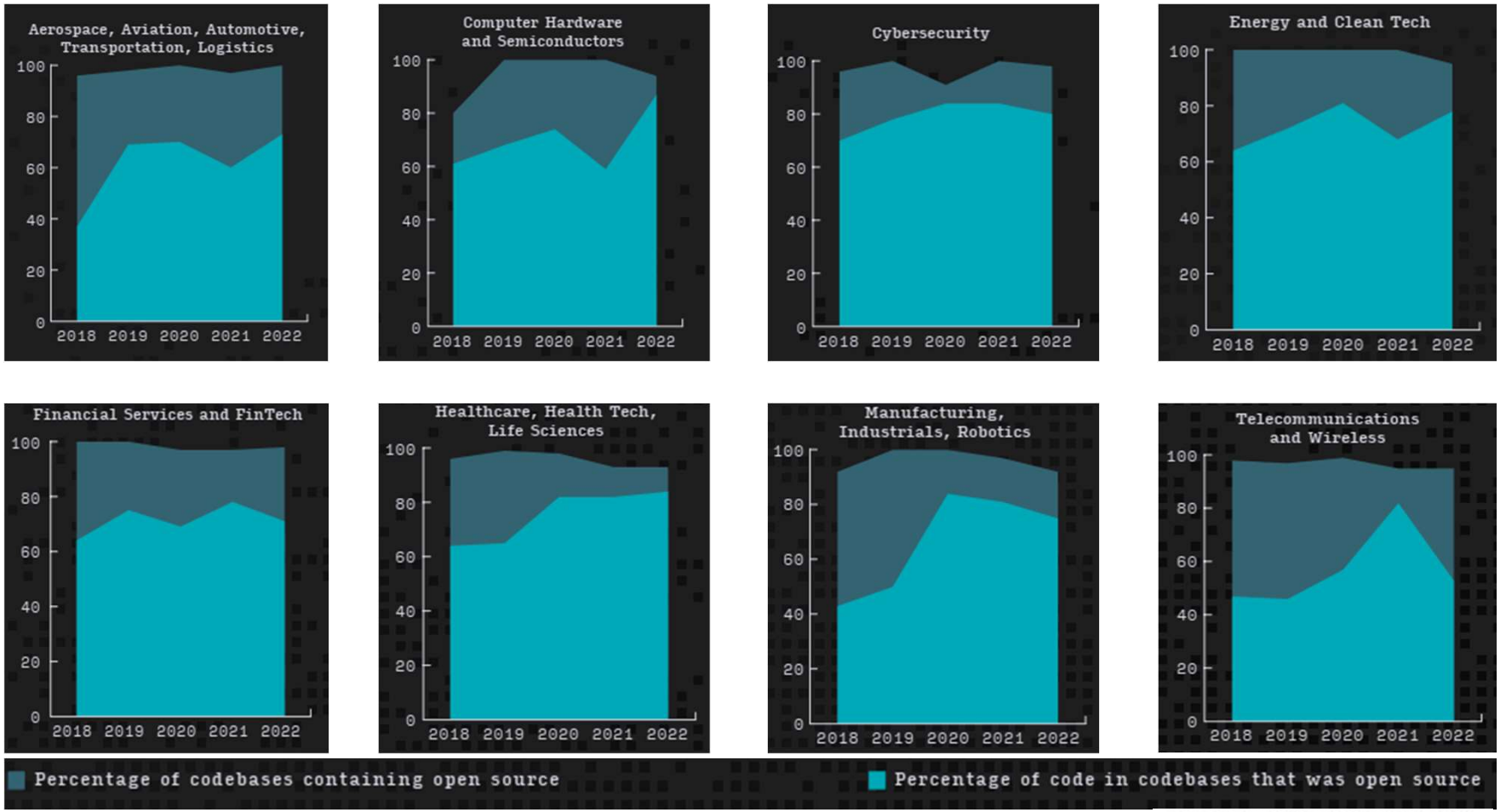
- 5G Networks introduce a major transformation in the Telecom Networks
- Networks are realized more of by software components



Open Source Software ecosystem is very vast. Isn't It?

- Firmware Powering the hardware
- Device Drivers
- Operating System
- Hypervisors
- Container Software
- Virtual Machine Images
- Container Images
- Middleware
- Libraries
- APIs
- SDKs
- Toolsets including IDEs Compilers
- Applications/Services

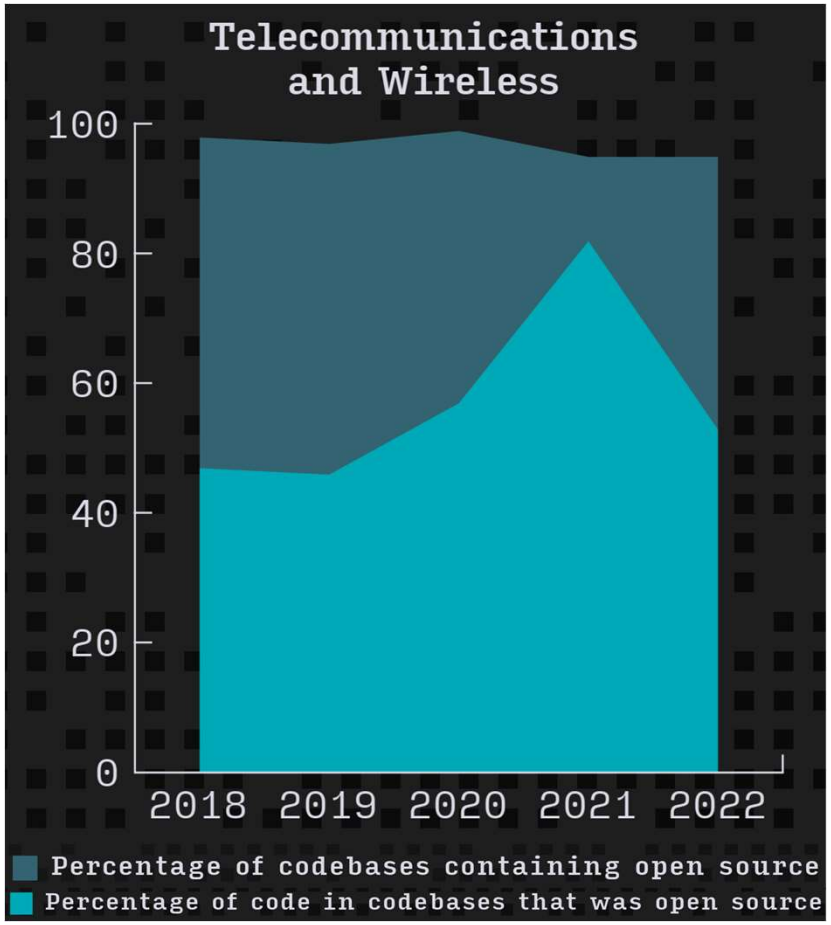
Open Source Adoption by Industry



Courtesy Synopsis, Inc.

Open Source Adoption by Telecom Industry

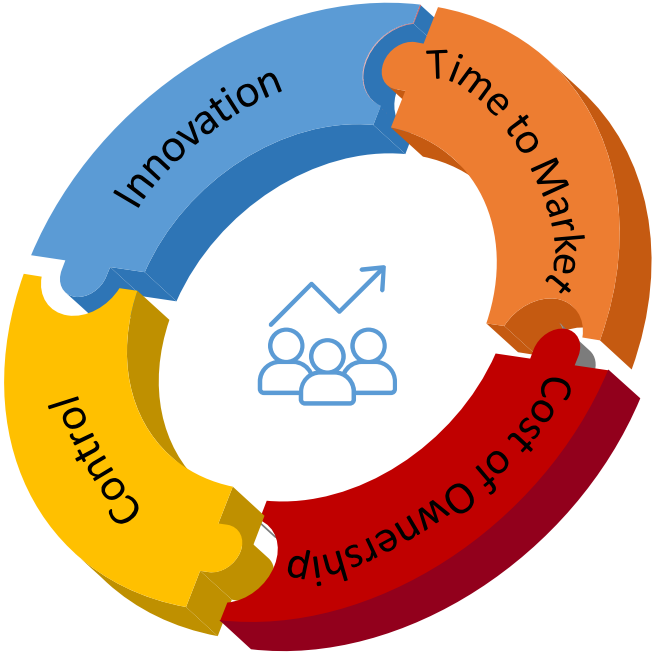
- About 95% of the projects/products contain open source component
- About 50% of the total codebase are from open source



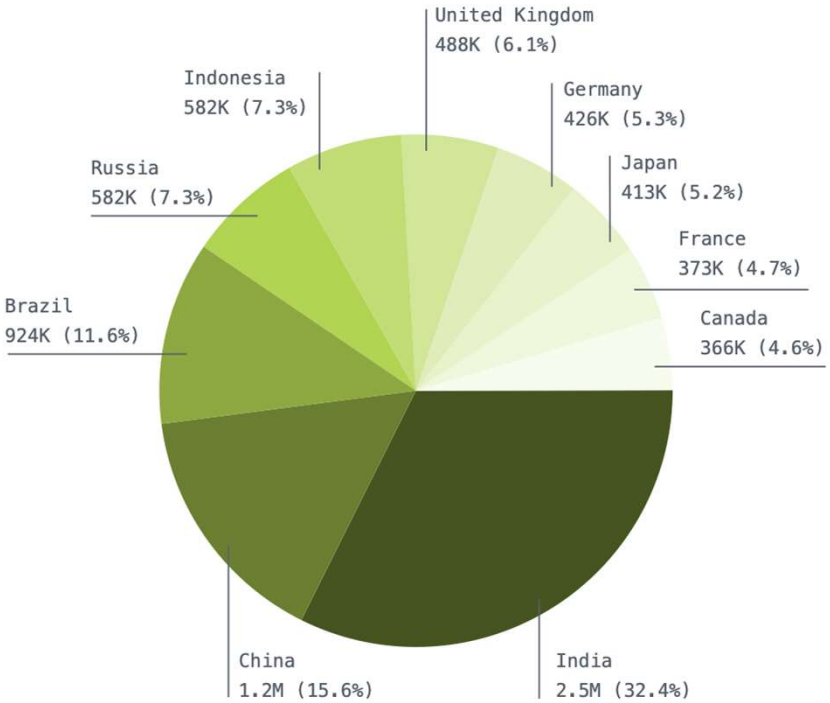
Courtesy Synopsis, Inc.

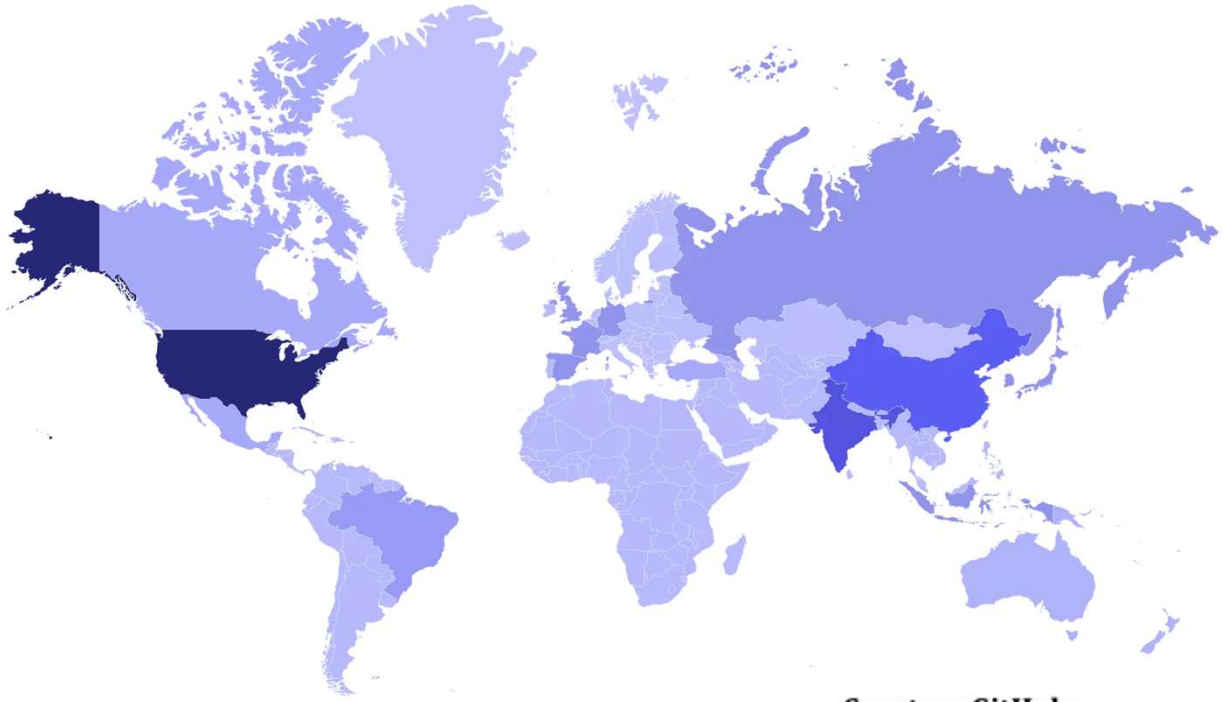
Motivating Factors for Open Source Adoption

- Accelerated Innovation
- Lower Time to Market
- Lower Total Cost of Ownership
- Increased Visibility and Control
- Simplify the Standardization Process
- Better Learning Opportunity



How Indian developers respond to this trend?



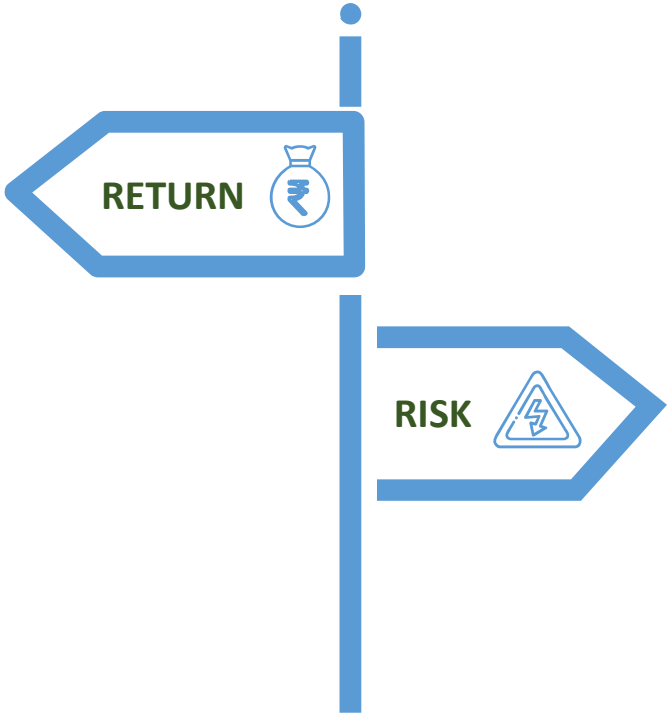


Courtesy GitHub

"India's population on our platform alone totalled 9.75 million people in the past year—and more than 2.5 million new people in India joined GitHub in 2022. If this trajectory continues, we predict that Indian users will match the current United States GitHub developer population by 2025" - GitHub

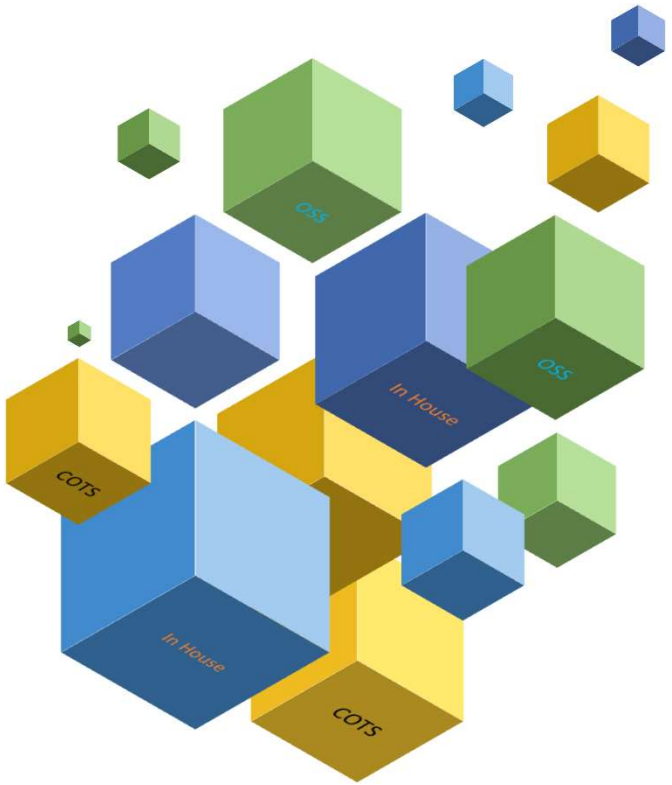
What is its impact on 5G and beyond Development?

- Benefits of opens source are obvious and enticing
- Opens up great opportunity for startups and SMBs by reducing the entry barrier of new technology business
- Great boost for the software industry for the development of 5G and beyond solutions of telecom domain
- Faster standardization through rapid prototyping and early feedback



System Development Strategies

- Complete In house development
- COTS Software
- Open Source Software
- Third Party Software
- Managed Services(SaaS, IaaS, PaaS)



Certain alarming facts from Open Source Domain

Heartbleed computer virus stayed undetected for 2 years - India Today

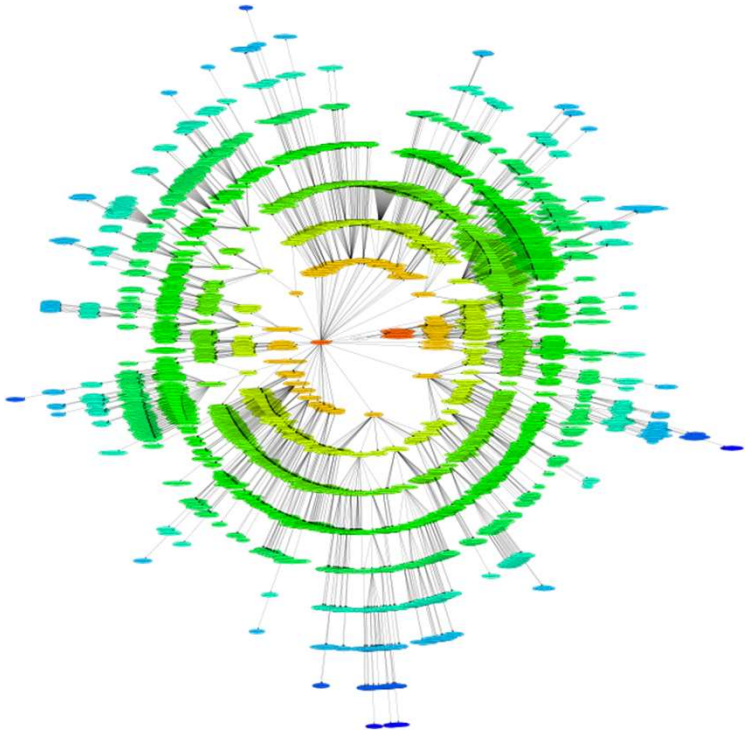
Hackers could crack email systems, security firewalls and possibly mobile phones through the "Heartbleed" computer bug, according to security experts who warned on Thursday that the risks extended beyond just Internet Web servers.

"Analysis on a random sample of 521 advisories from across our six ecosystems finds that **17% of the advisories are related to explicitly malicious behavior** such as backdoor attempts." – GitHub 2020 Report

Bugdoors and Backdoors - Backdoors are software vulnerabilities that are intentionally planted in software to facilitate exploitation. Bugdoors are a specific type of backdoor that disguise themselves as conveniently exploitable yet hard-to-spot bugs, as opposed to introducing explicitly malicious behavior.

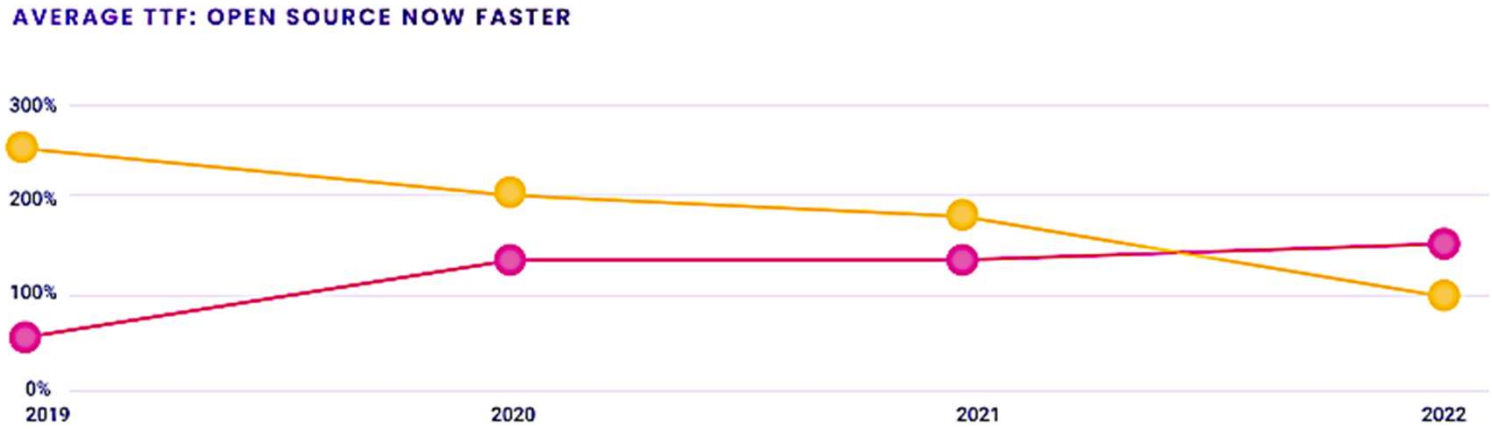
Open Source Security Risks

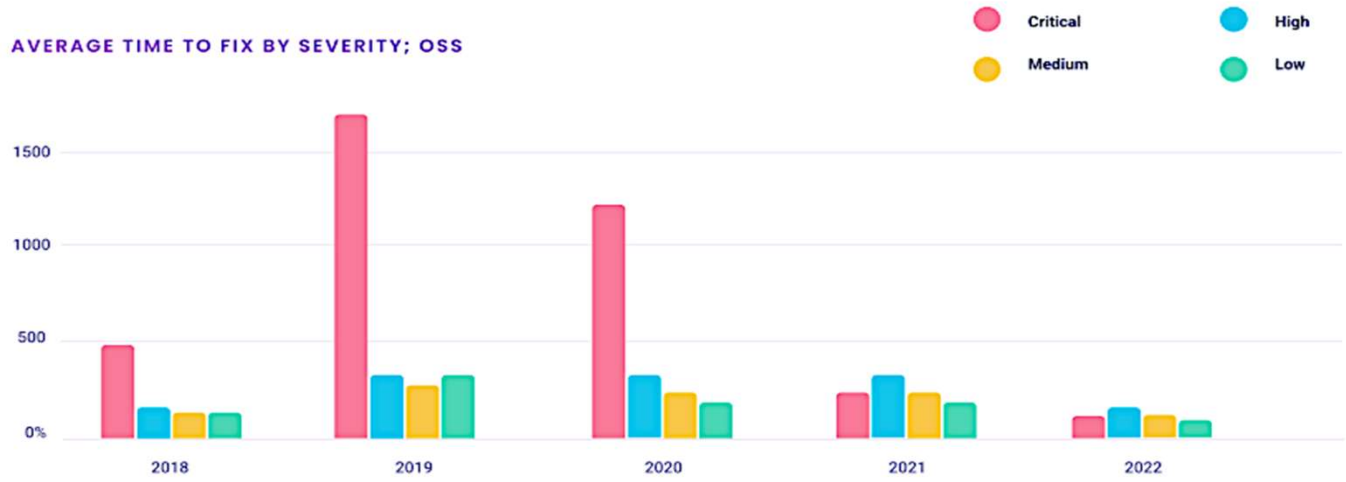
- Known Vulnerabilities
- According to Synopsis security report, 84% of codebases have at least one open-source vulnerability, and 48% have high-risk vulnerabilities
- According to the Mend database, the number of published open source software vulnerabilities in 2020 rose by over 50%.
- Vulnerabilities of Transitive Dependency
- Untracked Dependencies



React App Dependency Graph
Image Source – www.bytesafe.dev

Time to Fix (TTF) the Vulnerabilities





Open Source Security and Software Supply Chain Attacks

- Open Source Software Risk and Supply Chain Security are related
- A software supply chain attack occurs when malicious code is purposefully added to a component and distribute the same to their targets using the supply chain.
- Methods of attack in supply chain
  - Directly Inserting Malicious Code as a committer
  - Compromised committer accounts
  - Compromised certificate for package signing and distribution
- Most of the supply chain attacks are exploiting non patched vulnerabilities
- Event-Stream Backdoor in NodeJS is an example of supply chain attack using open source software

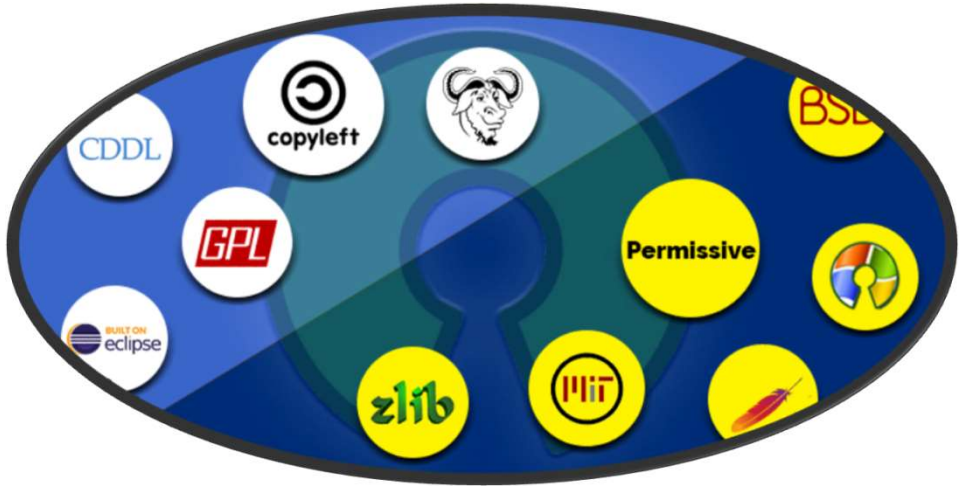
Risks related to Software Quality

- Unmaintained Software
- Outdated Software
- Performance
- Consistency and Stability
- Resource Leaks leading to sudden death
- Availability of Support
- Popularity and Active Community
- Immature development process
  - Version Control
  - Regression Test Suites
  - Review Guides and Documents
  - Security Audits
  - Pre Commit Checks

"The Open Source Security Foundation (OpenSSF) Best Practices badge is a way for Free/Libre and Open Source Software (FLOSS) projects to show that they follow best practices."

Risk associated with Open Source Licensing

- Open Source Licenses mainly classified into
  - Permissive
  - Copyleft
- Major Permissive License
  - MIT
  - Apache 2.0
  - BSD (2- clause, 3- clause and 4-clause)
- Copyleft Licenses
  - GPL
  - LGPL
  - Eclipse Public License (EPL)
- Terms for usage, source code modification and distribution are widely varying across these licenses.



Risk Mitigation - Security Risks

- Inventory the software components and its dependencies
- Maintain the Software Bill Of Material (SBOM)
- Track the Vulnerability of software components and its dependencies
- Implement Configuration and Change Management Process and its best practices
- Incorporate Software Component Analysis (SCA), Static and Dynamic Application Security Testing (SAST/DAST) as part of development process
- Validate the Integrity of the open source code
- Implement Security Standards for the development process like NIST SP 800 -160 ‘System Security Engineering’

Risk Mitigation – Software Supply Chain

- Implement Software Supply Chain Risk Management like NIST SP 800-161
- Sufficient evaluation of features and source code
- Verify the integrity and authenticity of the software package
- Verify Pedigree and Provenance of the software
- Evaluate the geopolitical risk that can raise due to the location of the software origin or distribution centre.
- Ensure the security of the build environment

Addressing the Quality Risks

- Ensure the popularity and maintenance status of the project through commit history, dependent projects, number of contributors and forks
- If there is no specific reason, avoid using outdated software component
- Continuously evaluate the performance and functionality of the software.
- Use wrapper APIs around open source component to support virtual patching and evaluate alternate sources
- Ensure the commercial support of the open source component
- Build expertise of the open source component in parallel to main development if the open source component is a critical one.
- Ensure the maturity of the open source development process.

Addressing the License Risks

- Use Software Component Analysis to identify the dependent components including transitive dependency and associated licenses
- Avoid using components that have conflicting license terms with that of your product.
- Maintain and update the SBOM with license information
- Obey the legal obligations for incorporating the required third party licenses as part of product distribution.
- Check and Comply with the regulatory requirements applicable in your jurisdiction where you operate and provide service.

Point of Discussion

- Any of these approaches are risk free?
  - Complete In house development
  - Third Party Software
  - COTS Software
  - Managed Services(SaaS, IaaS, PaaS)
  - Open Source Software
- There are specific risks associated with each development approach

Conclusion

- Identify the risks associated with the software component
- Measure the risk in terms of its impacts
- Determine the risk treatment approach
- Implement counter measures
- Incorporate the best practices throughout the software development life cycle processes
- Continuously Monitor and Measure the effectiveness of risk management
- Continuously Improve the risk management process

Thank You

www.cdac.in

धन्यवाद आभार આભાર ಧನ್ಯವಾದ ধন্যবাদ நன்றி നന്ദി ధన్యవాదాలు

One Vision. One Goal... Advanced Computing for Human Advancement...