LF NETWORKING

LFN Developer & Testing Forum

Quality and Assurance Pipeline

Muddasar Ahmed
msahmed@mitre.org

Who gives a damn!

Headline

XZ Utils Backdoor- CVE-2024-3094 “An example of how trust gets abused in open-source”

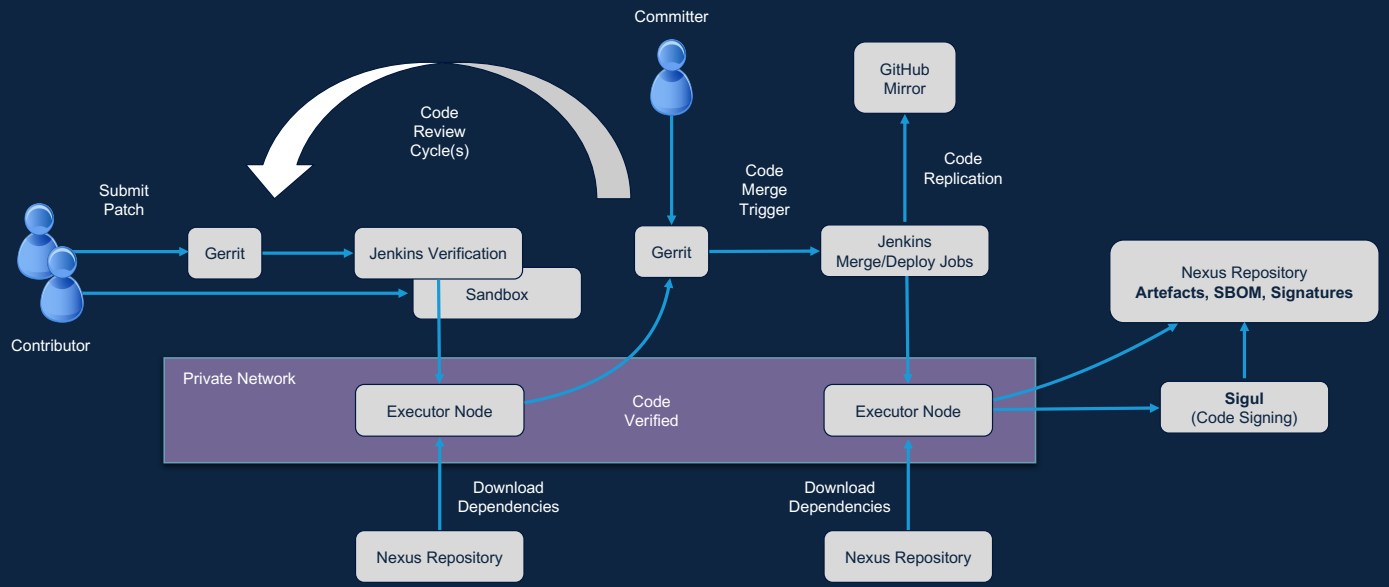
“Evolving from thinking of cybersecurity as a necessary evil to a **strategic advantage** sets you apart.”

ONAP LF-IT CI/CD

LF NETWORKING
LFN Developer & Testing Forum

- LF-IT/ONAP CI/CD is a development approach
- Emphasizes automation and frequent code changes.
- It involves a seamless pipeline from code integration to deployment, promoting agility and efficiency in software development.

CI/CD Workflow Diagram



CI Security

- Access roles (contributor, Committer, Tester)
- User management (credentials, permissions- Least privileged, MFA)
- Detective controls
- Infrastructure controls
- Hardening build servers/nodes

Contributor/Contribution Management

- Artifact validation
- Static code analysis
- Integrate security testing directly into your pipelines
- Example: Docker containers, scan them with tools such as Inspector or Twistlock,
- If your tool has an API, you can use GitHub action to trigger it in CodePipeline
- If any security test fails, the Pipeline stops, and the code does not move to production

Governance

- Structure, roles and rituals.
- create vulnerability reports to accompany each release: list vulnerabilities in project created code and known CVEs in 3rd party packages
- create Jira ticket per code vulnerability and package vulnerability
- static application security testing (SAST) and software composition analysis (SCA) – **code coverage**
- Security and Architectural reviews for features and changes
- Manage technical Debt by assigning minimum fixed story points for debt retirement

Integration of Quality Assurance- Automated Testing

LFN Developer & Testing Forum

Automated Code Review: sonarcloud.io

SonarCloud: https://sonarcloud.io/organizations/onap/projects
SAAS (Cloud hosted service)

Features

- Code auditing/review tool
- CI/CD integrated (automatic analysis of new code drops)
- Coverage reports
- Dependencies (e.g. Python modules and Java packages with known issues)
- Static/dynamic code analysis

Code Coverage

...l Review: Gerrit

...errit: https://gerrit.onap.org/

Features

- +2 and merge rights restricted and controlled by INFO.yaml
- Oversight of permissions managed by the project TSC meetings
- Approvals required for new top-level repository creation

Automated Security Scanning

- Security is integrated into the development process through automated security scanning. Static Application Security Testing (SAST) and
- Dynamic Application Security Testing (DAST) tools such as SonarQube and OWASP ZAP help identify and remediate vulnerabilities early in the development lifecycle. (Wish list)
- Dependency Management- for code reuse

Code Quality Metrics: Code Review and Static Analysis

- Regular code reviews and static analysis
- Tools like SonarQube helps maintain a high standard of code.
- Developers receive constructive feedback, promoting the creation of robust and maintainable software.

Traceability: Track code in CI/CD Pipeline

- CI/CD tools like Jenkins and GitHub enable visibility into the status of each code change,
- Facilitating effective collaboration and issue resolution.
- Use of Dashboards and Reports to assist quick review

Compliance and Auditing: Automated Compliance Checks

- Automated compliance checks with tools like InSpec help ensure that applications adhere to industry standards and regulatory requirements. This proactive approach mitigates the risk of non-compliance issues.
- Compliance may come from regional regulations or industry standards.
- MITRE Security Automation Framework (SAF) has published profiles for CIS Benchmarks several other best practices. Similarly commercial tools also offer profiles for specific industry.
- CNTI/Anuket etc..

Continuous Improvement: Regular Retrospectives and Feedback Loops

- Regular retrospectives and feedback loops are essential for continuous improvement. Learning from incidents and actively seeking feedback helps teams enhance their processes and adapt to evolving requirements.
- Improve community engagement to get feedback from End Users.

Secure Delivery

- Signed software and SBOMs/SW
- Documentation
  - User manuals/How To, Default/ranges in configs
- Release notes
  - Fixed, Outstanding
- Bug report (defects fixed and outstanding)
  - How to file a field found bug/feature request

LF NETWORKING

LFN Developer & Testing Forum

Anti-Trust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrustpolicy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.