Code Quality & Security Workstream Update

PI-18 Workstream Feedback Session – 27 April 2022

Presenters

Godfrey Kutumela

Support Members : Aime Bukasa, Kim Walters, Victor Akidiva, Pedro Barreto, Michael Richards, Simeon Oriko, Miguel de Barros, Sam Kummary, Lewis & Tom Daly

Workstream Overview

Objective:

- *Continuously improve the Trust (reliability, transparency, privacy, compliance, quality and security)* of the Mojaloop Platform and transform our approach to quality and security in line with L1P principle on data privacy and emerging technological trends.

Delivery Model:

- Supports both *functional and non-functional* requirements of the project, working alongside with other *workstreams & various governance committees o*n a *shared responsibility Model.*

Approach:

- Standards and Control Centric – Define and maintain Mojaloop software quality and security standards and guidelines – In certain areas we provide reference implementation whereas for other areas we require certain policies or standards to be adhered to.

- Risk Centric – Perform risk assessments and threat modelling to identify, validate, classify & prioritize security requirements.

Milestones:

- PI 1 – 8 : Foundation Phase - Built-in confidentiality and Integrity as part of the Core Mojaloop Architecture.
  - ✓ Developed and Implemented Signatures, MTLS, PKI, encryption standards
  - ✓ Established a code quality and security framework - DevOps & CI/CD Tools automation, workflows & policies

- PI 9 – Current: Phase 5 (One Loop for all) – Consolidate, optimize & improve.
  - ✓ Baselining Mojaloop against best practice standards – PCI DSS, ISO27001 and GDPR
  - ✓ Embed Security into the reference architecture, new functionality additions and DevOps processes
  - ✓ Maintain and enhance DevSecOps processes

PI 17 Objectives

1. Fraud and risk management system (FRMS) security review and validation

2. Vulnerability management and DevSecOps process/tool enhancements

3. Perform a quarterly open-source software (OSS) scan

   - Open-Source License Assessment - To ensure compliance with Mojaloop open-source license policy
   - Open-Source Security Assessment - To ensure that libraries used are not vulnerable and outdated to minimize security risks

FRMS security implementation review and validation

Objective - Review and validate security implementation on all new major functionality additions to ensure alignment with mojaloop security standards and policies.

Completed to date – PI 18:

1. Reviewed and validated FRMS security designs and implementation
2. Performed an OSS scan and remediated all high findings

FRMS security implementation review and validation

Core security controls built into the current Beta version

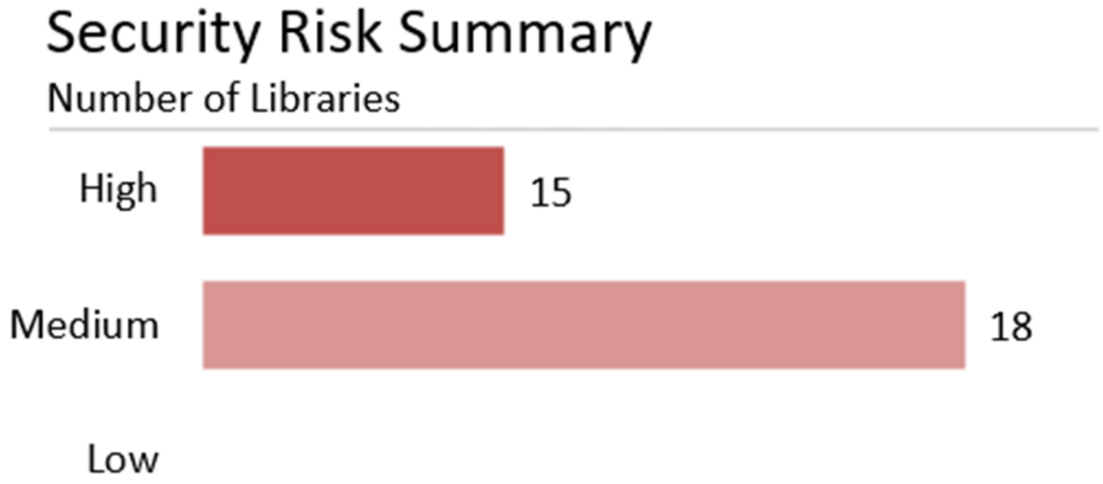
- *The focus was on what needs to be natively supported by the platform out of the box*

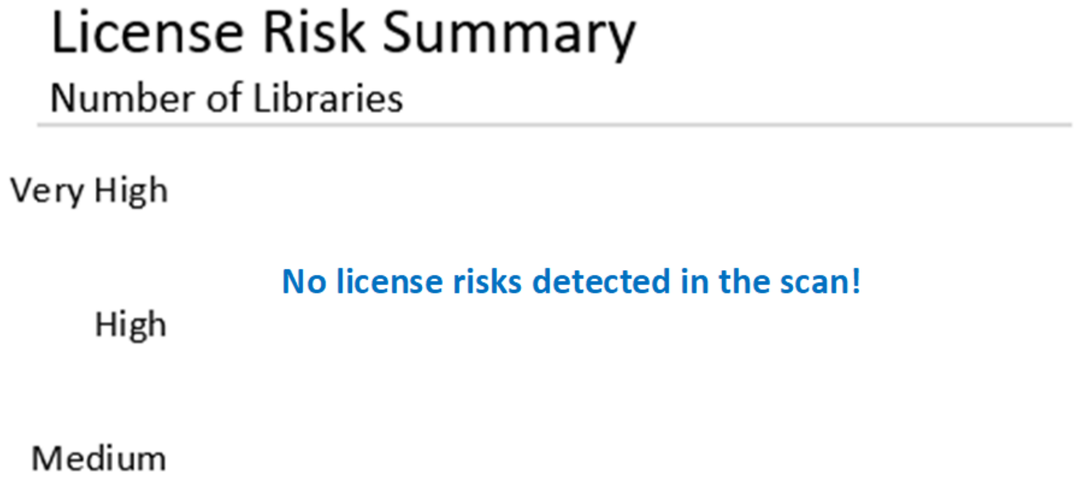
Design Requirement	Description	Reference Implementation
1. **Password setting must be policy based. Hashed and never in the clear in the DB.**	Length, composition (letters, digit, special characters, number of trial and duration). Minimum age of 24 hours.	Regarding RBAC all passwords follow the Microsoft password standard. See "Password Restrictions" in the provided Documentation.
2. **Adhere to best practice in - application authorization model - The ROLE needs to be obtained from an AIM subsystem.**	Role and permission definitions should be segregated and decouple from functionality.	Roles are fetched and managed in Azure portal and then bound to the cluster via RBAC. Keyclock and Azure AD are used for this.
3. **2FA needs to be supported**	2FA should leverage FIDO 2 or Webauthn where feasible for extended support	Keycloak as seen in this article officially supports 2FA. However, the Actio platform does not yet have an implementation thereof
4. **All traffic in transit needs to be adequately protected**	Leverage only TLS 1.2 and TLS 1.3 and TLS.1.1 should not be allowed.	As seen in proof all cluster ingress is HTTPS, no exceptions.
5. **Certificates needs to be signed by an established a trusted certification authority.**	Certificates need to be distributed to counterparts for signature verification.	Certificates are manually managed within Kubernetes as seen in the proof.
6. **Ensure cloud security**	Harden the cloud platform in line with best practice standards	Hardening plans for Azure include the following: - Ensuring all components and tools are up to date. - Any dependencies added are scanned and vetted - Strict lockdown of system access and level of access - Defender security alerts is enable for kubernates

FRMS OSS scan results overview

Codebase size

- 42 repositories with 593 unique libraries





Upon Analysis:

- No library was flagged with a very-high or high-risk license
- All libraries with high security vulnerable have a fix available via an upgrade.

Version Management Status:

Out of 593 libraries

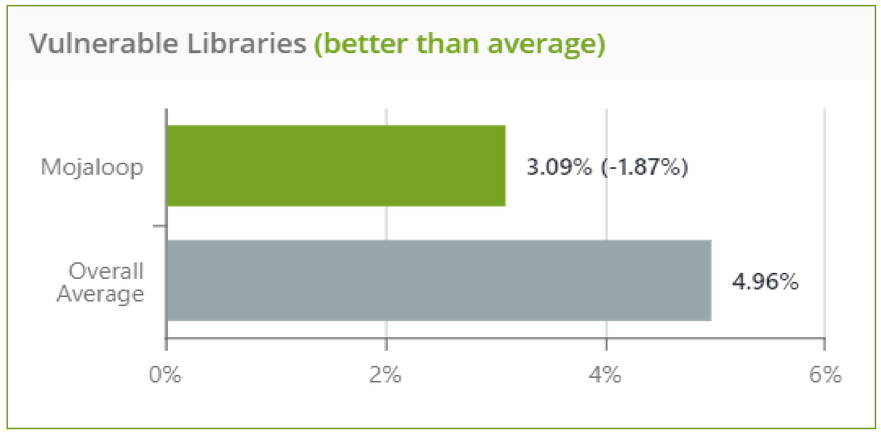
- 568 are up to date
- 25 libraries are outdated

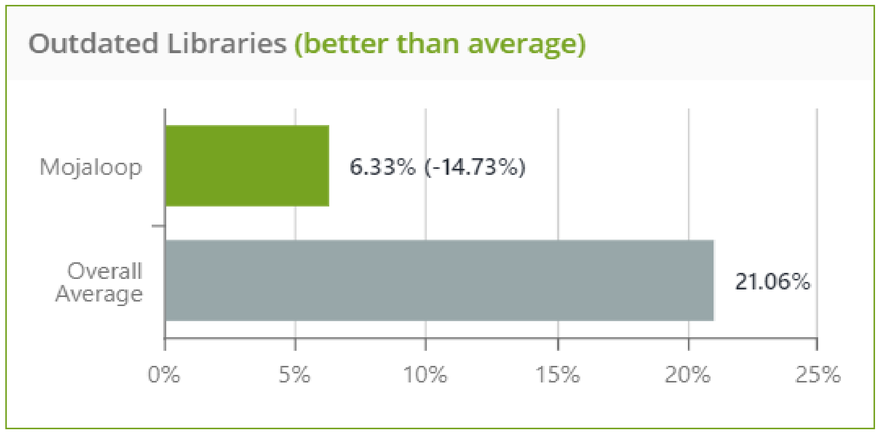
Quarterly OSS Overview

All Mojaloop repos were included in the scan

18 January 2022	18 April 2022
104 Codebases with 7055 libraries • **6625** libraries are up to date – Excellent version management • **430** libraries are outdated and should be reviewed for upgrades • **130** libraries with multiple version and should be upgraded to the most updated version.	**111 Codebases with 7729 libraries 7729** • **7221** libraries are up to date – Excellent version management • **489** libraries are outdated and should be reviewed for upgrades • **165** libraries with multiple version and should be upgraded to the most updated version.

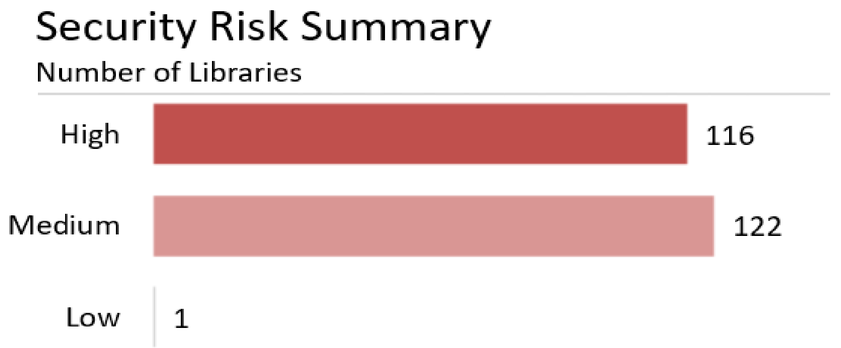
According to WhiteSource Benchmark database, Mojaloop is better than average in vulnerable libraries and in outdated libraries, potentially indicating that Mojaloop have an effective security vulnerability scanning capability and version management controls.

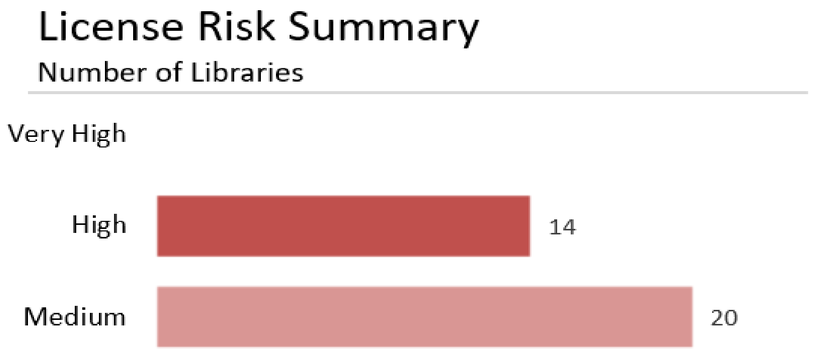




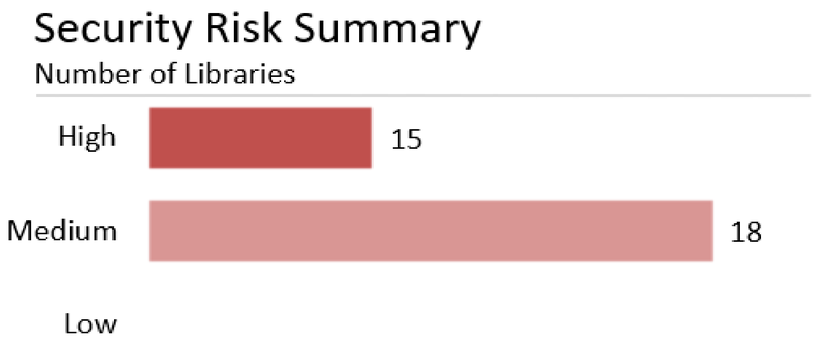
Findings Overview

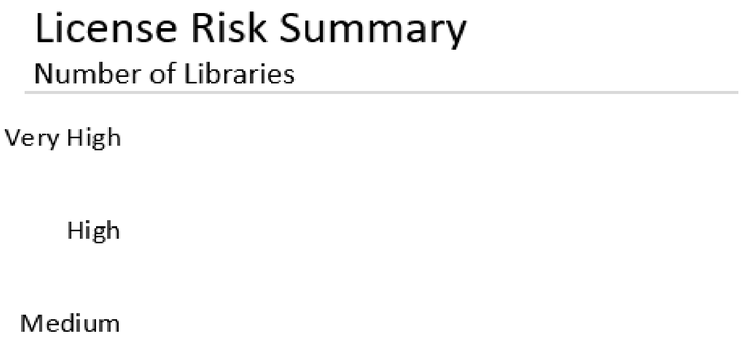
18 January 2022





18 April 2022



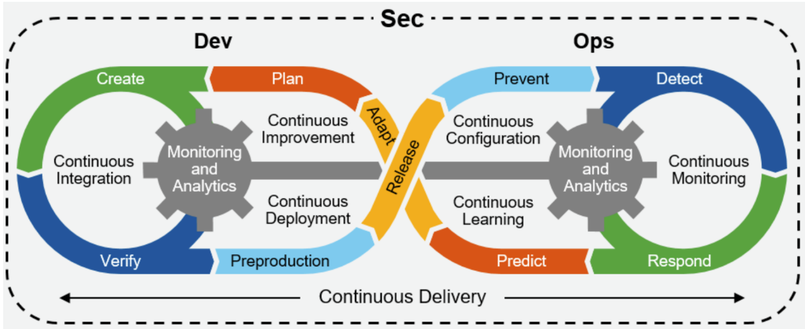


Vulnerability management and DevSecOps process/tool enhancements

On going maintenance and enhancements of the DevSecOps processes, policies and tools and policies.

On going support Activities:

1. Regular Security Patches + Updates
   - Addressing regular Dependabot and Snyk security alerts
   - Running `npm audit` on flagged repos
   - Improving CI/CD Workflows and adding new policies as needed
   - Partial implementation of automated releases
2. Performed an independent OSS scan once a quarter - #2750
3. Developed a Node version upgrade strategy – DA issue #78
4. Developed a draft framework for code security standard - #2722
5. Developed code integrity assurance solution – Helm charts signing - #2634
6. Exploring with Code Secret Scanning(#2737) and CodeQL (SAST) - #2738
7. Preventing/Mitigating Open Source Supply Chain Attacks – DA issue #88 – On going exploring node-built features for dependency security checks!



PI 18 Objectives

1. Quarterly OSS scan

2. DevSecOps maintenance and enhancements
   a. Regular security patches + updates
   b. Finalize the code security standard
   c. Prevention of dependency attacks
   d. Implement the code integrity assurance solution on helm charts
   e. Implement Github secret scanning and Code QL SAST - as per approved code security standard

3. Documentation
   - DevSecOps guideline for implementors
   - Contribute to the Mojaloop security whitepaper

muchas gracias

Thank you

Questions and comments