mojaloop

Mojaloop Application Security Summary

Core Quality\Security Core Team – Kim, Pedro, Lewis, Godfrey, Miguel, Sam & Victor

Community Contributors – Karim, Azeem & Aime

Application Security Initiative Overview

In order to address application security issues Mojaloop, we have a dedicated security stream consisting of the members the core team and community contributors with the key focus of building security into the development cycle of the Mojaloop platform as much as possible following the below principles.

Underpinning Principles:

- Deep integration into DevOps and CI\CD processes (All security must be automated gates to keep the DevOps workflow from slowing down)
- Developer centric and inclusive (Empower and delegate with trust security integration activities to the devops teams)
- Open source first approach to tooling (Commercial tools to be considered only if open source tools does not fulfil the requirements)
- Adapt Application Security to Cloud Native Technologies (Containers and Microservices)

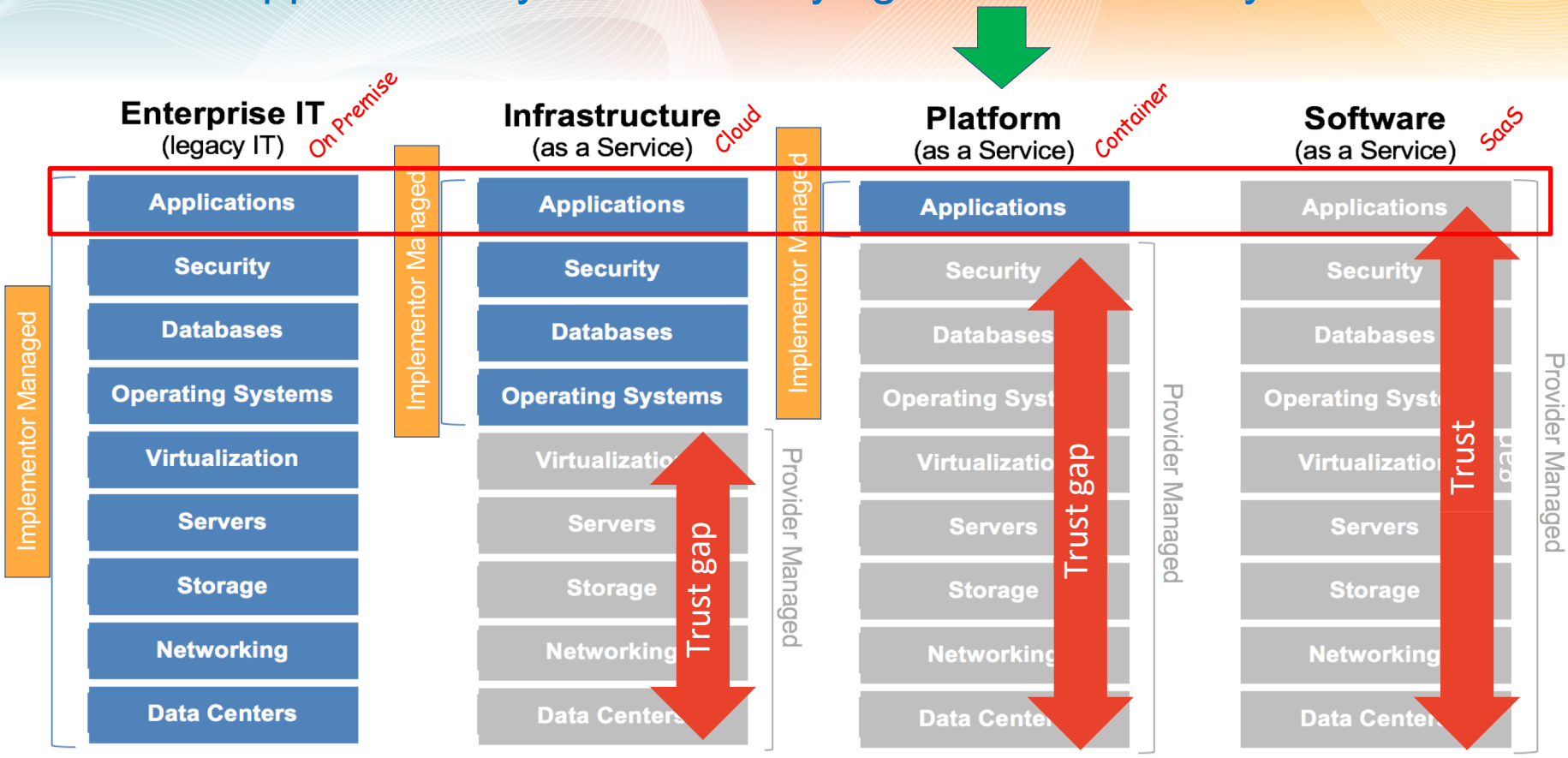
Our Prescriptive Security Approach

Prescriptive in the sense that we require the use of certain security practices and techniques (delivered as guidelines).

For some areas we will have reference technology implementations in place, like npm dependency, docker image vulnerability and license scanning, for other areas we require certain policies or standards to be adhered to and verifiable.

Mojaloop DevSecOps Approach

Focus is on the application layer not underlying infrastructure layer



Output:

- ***Provision of a security guidelines based on the Platform as a Service Model*** – Implementers to adapt to their own environment and policies
- Alignment to CIS benchmarks and CSA cloud security essentials for IaaS and PaaS

Application Security Domains

Our application security programme covers the four domains briefly descripted below.

1. Open Source Vulnerability Management – To enforce consistent use of non-vulnerable and non-obsolete open source components and dependencies.

2. Open Source License Compliance – To ensure compliance with Mojaloop open source license policy

1. Static Code Analysis – To ensure custom code is free of well known high and medium vulnerabilities before any code merging.

2. Container Security – To ensure secure image configuration and also find\fix in container based vulnerabilities and kerbunates applications

There are more solutions under investigation in this PI and will be added to these or new domains once they fully implemented, tested and signed.

Open-Source Vulnerability Management

Objectives	Perform vulnerability assessment on open source components (composition analysis) before every commit, pull request and release as per our CI\CD pipeline workflow and security gates
Tools	1. NPM - https://docs.npmjs.com/cli/audit 2. Archery (Under Evaluation) 3. Snyk (Under Evaluation)
Policies and Standards	1. Default tool policies has been applied and busy investing best practices polices as a basis to adopt a Mojaloop specific policy 2. No standard has been adopted yet for this domain – Still under investigation
Recommendation to Implementors	1. Custom-tuned policies to your use case using the outcome of your threat and risk assessment.

Static Code Analysis

Objectives	Perform code level vulnerability assessment and code quality analysis with every commit, pull request and release as per our CI\CD pipeline workflow and security gates.
Tools	1. NodeJsScan - https://github.com/ajinabraham/NodeJsScan 2. SonarQube - https://github.com/SonarSource/sonarqube 3. Owasp Dependency Check - https://owasp.org/www-project-dependency-check/
Policies and Standards	1. Default tool policies has been applied and busy investing best practices polices as a basis to adopt a Mojaloop specific policy 2. No standard has been adopted yet for this domain – Still under investigation
Recommendation to Implementors	1. We advise implementors to custom-tuned their policies to your use case using the outcome of your threat and risk assessment.

Container Security

Objectives	Ensure only secure container configuration is used in Mojaloop and find and fix container based vulnerabilities before any release.
Tools	1. Dockerfile Security 2. Anchore 3. AppArmor
Policies and Standards	1. Default tool policies has been applied and busy investing best practices polices as a basis to adopt a Mojaloop specific policy 2. CIS Cloud Security Benchmark (141 Checks)
Recommendation to Implementors	1. We advise implementors to custom-tuned their policies to your use case using the outcome of your threat and risk assessment.

Open Source License Compliance

Objectives	Ensure compliance with the Mojaloop open security license policy before every commit, pull request and release as per our CI\CD pipeline workflow and security gates
Tools	1. NPM License check 2. Snyk (under investigation)
Policies and Standards	1. Apache 2.0 and the ones compatible to it are allowed / conforms to the Apache 2.0, whereas GPL / GNU aren't?
Recommendation to Implementors	1. Custom-tuned policies to your use case using the outcome of your threat and risk assessment.

mojaloop

More coffee stains

Thank You

For Queries feel free to contact

sam@modusbox.com
lewisd@crosslaketech.com
victor.akidiva@modusbox.com
godfreyk@crosslaketech.com

Oh no! A spill!