Mojaloop Secure Network & Application Access Standard and Recommendations

As part of the PCI DSS section 1 - Build A Secure Network baselines findings, we have made recommendations on how implementors should address the basic network security requirements of the PCI DSS Standard. We have provided a clear demarcation of responsibility and ownership for all requirements in the section and provided a line-by-line recommendations on addressing those.

To further help improve the network perimeter posture of the Mojaloop Hub, the following additional measures are recommended for considerations at both hub operator and Mojaloop OSS level.

Some of the Network level threats facing any Mojaloop implementation include:

1. DOS and DDOS Attacks – these attacks involve sending multiple requests to the network infrastructure targeting web applications that exceed the infrastructure's capacity, resulting in service unavailability.
2. MITM - an attacker intercepts communication between two parties either to eavesdrop or modify the information being exchanged secretly.
3. DNS Hijacking – Domain Name Service (DNS) queries are incorrectly resolved to redirect users to malicious sites unexpectedly. Usually, a part of a MITM attack.
4. TCP Session Hijacking – The attacker's goal is to create a state where the client and server cannot exchange data; enabling him/her to forge acceptable packets for both ends, which mimic the real packets. Thus, the attacker can gain control of the session.
5. Host Address spoofing - a malicious user transmits data packets with an IP address, indicating that the packets originate from another trusted machine within the network, avoiding detection.
6. Application and infrastructure misconfiguration implies failing to implement all the security controls for an infrastructure component, a server or web application, or implementing the security controls, but with errors. This could expose services to attackers and grant opportunities for attack. Complex networks with many running applications also run the risk of management overheads leading to some components being neglected from a security configuration standpoint.

Stride analysis of the network threats

	STRIDE (See Appendix)					
VULNERABILITY	**S**	**T**	**R**	**I**	**D**	**E**
DOS and DDOS attacks					X	
Man in the Middle (MITM) traffic eavesdropping		X	X	X		X
Monitoring and Intrusion detection	X	X	X	X	X	X
DNS Hijacking				X	X	

TCP Session hijacking				X	X	
Host Address Spoofing				X	X	
Network Outage and Downtime					X	
Application and Infrastructure misconfiguration	X	X	X	X	X	X

To address the risks above, we recommend the following requirements as proposed from a PCI-DSS perspective.

Requirement 1: Secure Network Perimeter

1. First and foremost, we emphasize the need for a perimeter network-level firewall (layer 3-4) on both the egress and ingress points within the Mojaloop hub as first-line protection which will also help fulfil much of the required controls of the PCI DSS Section 1. This requires Mojaloop implementers to enforce in-depth defence design for their production setup.

- We also recommend setting up a DMZ where all publicly exposed services will be hosted. These will include the gateways and externally facing web servers.

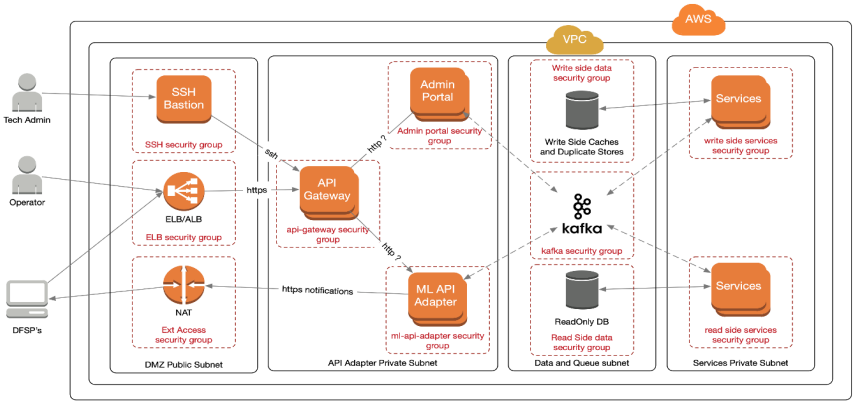
2. Link Encryption – where possible, the P2P connection between scheme participants and the Hub can enforce link encryption. However, in cases where costs are prohibitive, we will insist on ending data encryption in transit.

3. Load Balancing - After the firewall, we recommend deploying a load-balanced to help route traffic between n+ one web/API gateway nodes.

4. Web and API Gateway - The purpose is to establish a secure perimeter for all web-based traffic entering and leaving the Hub. The Web gateway can also act as the SSL offloading endpoint for incoming traffic to the Hub.

    - This will enable the addition of a fine-grained web access policy at the perimeter without impacting on the transaction flow and paths. This targets the envisaged Portal architecture and will act a gatekeeper to protect the switch OWASP Top 10 related vulnerabilities that might slip through our static code analysis checks.

    - This will also enable the addition of security policy/rules on the fly to all API calls traversing the API gateway level, thereby allowing architect/developers to focus on writing functionality. The gateway will handle heavy security lifting inclusive of integration to ICAP enabled system like ATP and DCS. They were generally having an API Gateway will only additions non-functional services.



The diagram above is the security view of an enhanced secure network implementation - other implementations should follow similar boundaries.

- Subnets are firewalled network segments (logically or physically separated);
- Security groups are essentially firewalled zones, or groups of components, depending on where the traffic comes from that is where the firewall component will be on a physical installation - for example, there is a zone for the write side data components, only the write side services should be able to access it.
- The API gateway is making decisions about authentication and Authorization (using JWTs). TLS the ELB should be configured as a simple TCP load balancer so that the API Gateway does tls termination.
- API Gateway makes all HTTP routing decisions (layer 7 - HTTP/DNS requests);
- Firewall/Security groups do all TCP/routing decisions (layer 4 - TCP/UDP connections)

Commented [Ma1]: This might create a bottleneck as API GW is processing API calls and callbacks plus SSL offloading

This is a base recommended model, not everything that implementation will need to have, but it shows which component does what in terms of network security controls.

Requirement 2: User Access Management

To further enhance the IAM posture of Mojaloop, we recommend that implementors should also address the following over and above the password requirement of the PCI DSS Section 1:

1. Design and Implement a Privileged Identity Management - Secure Management of superuser credentials within the switch or Hub may reside within a hub or core switch. At the OSS level, default passwords will be used to get up and running on development and testing environments quickly. However, there is a need to manage all privileged access on all system components used in Mojaloop for their pre-production and production instance.

2. Design and Implement an Access Management Module - The purpose of this module is to facilitate the management access by admin users to various components within the switch core following a well-defined role-based access control system. This will reduce the risks of attackers traversing across the switch system components and network zones without any need for Authorization.

3. Design and implement federated access policies within the identity management services. These will govern how least privilege access can be issued to hub administrators who in turn will be able to create their internal users on the Mojaloop platform as permitted based on functional needs.

4. Design and Implement Centralized credential storage to facilitate secure storage of authentication artefacts and keys that will be used by people and applications within Mojaloop hub.

5. Design and enforce least privilege control across all types of actors within Mojaloop ecosystem.

Requirement 3: Application and Infrastructure Level Security

To further enhance the Application security of Mojaloop, we recommend that implementers should also address the following:

1. Transport Level Security – ensure all API and application communication is encrypted. This may also require TLS implementation between internal Mojaloop services that interact via a service mesh architecture where applications authenticate with each other.
2. Monitoring and auditing controls should be embedded within the running applications and network infrastructure to ensure that any anomalies can be quickly detected and responded promptly. This will include a SIEM with security dashboards and alerts. Automated network responses are an added advantage.
3. Deploy centralized configuration management for applications and infrastructure. This will allow tracking the inventory of running services, tracking uptime, and enforcing centralized configuration management for all network components. In this way, the security and infrastructure team can ensure regular patching, standardized configuration across segments, and monitor for any anomalies to documented policy and rectify.

Appendix

STRIDE Model and Controls

Type	Examples	Security Controls
Spoofing (S)	Threat action aimed to illegally access and use another user's credentials, such as username and password.	Strong authentication, e.g. passwords, MFA, digital signatures
Tampering (T)	Threat action aimed to maliciously change/modify persistent data, such as continuous data in a database, and alter data in transit between two computers over an open network, such as the Internet.	Integrity controls, e.g. digital signatures, permissions (ACLs)
Repudiation (R)	Threat action aimed to perform illegal operations in a system that cannot trace the prohibited operations.	Secure logging and auditing Digital signatures
Information disclosure (I)	Threat action to read a file that one was not granted access to or read data in transit.	Confidentiality controls e.g. permissions (ACLs), encryption
Denial of service (D)	Threat aimed to deny access to valid users by making a web server temporarily unavailable or unusable.	Resilience and business continuity, e.g., permissions (ACLs), Quotas, Filtering
Elevation of privilege (E)	Threat aimed to gain privileged access to resources to gain unauthorized access to information or compromise a system.	Filtering and Authorization controls, e.g., permissions (ACLs), input validation, patch management, intrusion detection, Anti-exploit and antimalware tools, firewalling

Mojaloop PCI DSS Standard Baseline Report

Mojaloop PCI DSS Responsibility Matrix,