Notarization API Threat Model

Owner: Neil Crossley
Reviewer: Tobias Wich
Contributors: Mike Prechtl, Dr. Detlef Hühnlein
Date Generated: Mon Sep 19 2022

OWASP Threat Dragon

Executive Summary

High level system description

The document discusses security concepts around GAIA-X Notarization Service. The document assumes a basic knowledge of security methodologies and practices in the audience reading the document and does not explain these topics in detail.

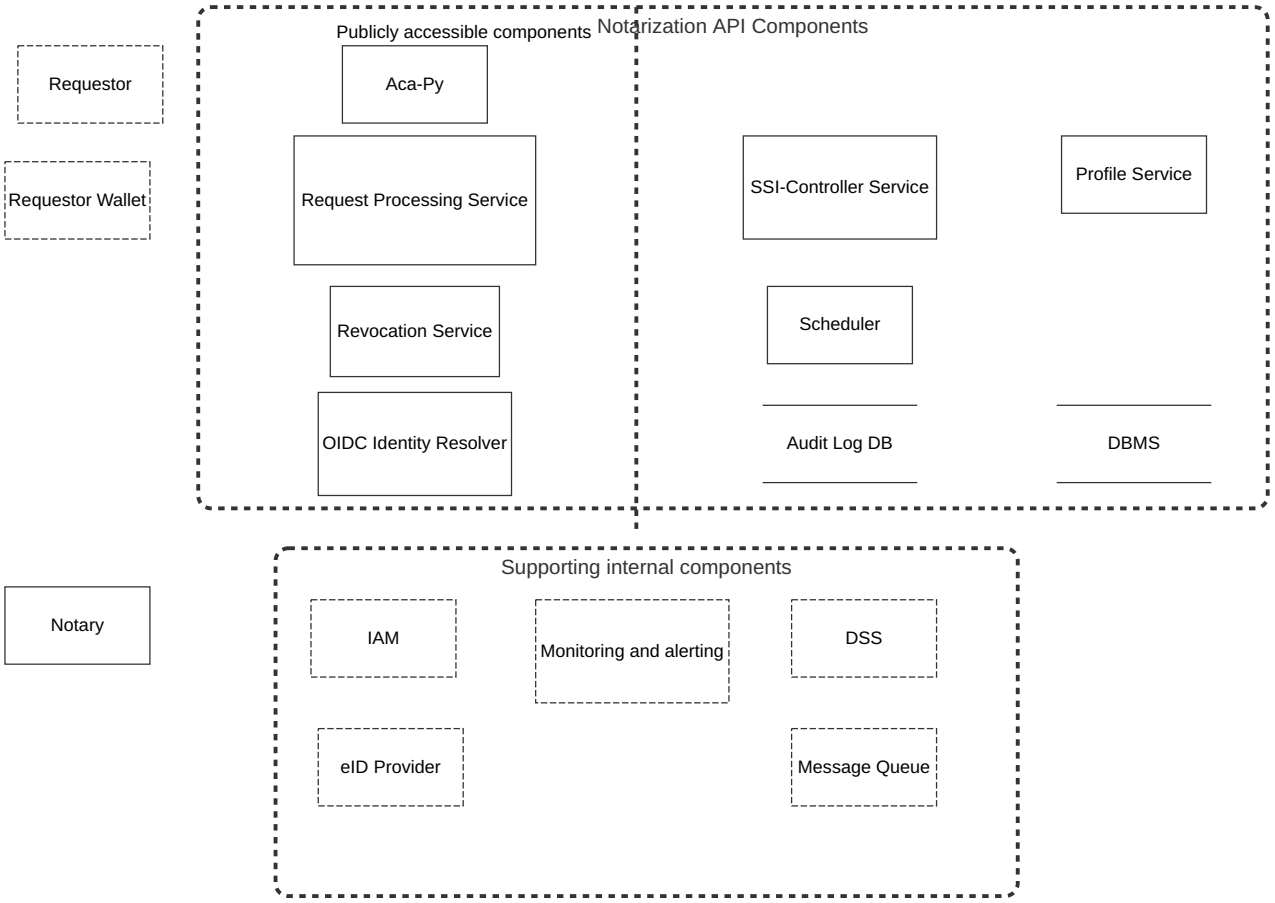
The goal of GAIA-X Notarization Service project is to implement a service which will issue verifiable credentials to respective holders.

The intent of this document is to provide an overview of implemented functionality with respect to information security principles and concepts taken into account in implementation of the Notarization Service project.

Summary

Total Threats	29
Total Mitigated	29
Mitigated by Admin	25
Not Mitigated	0
Mitigated by Admin / High	7
Mitigated by Admin / Medium	17
Mitigated by Admin / Low	1
Mitigated by Admin / Unknown	0
Open / High Priority	0
Open / Medium Priority	0
Open / Low Priority	0
Open / Unknown Priority	0

System components



System components

Notarization API Components (Trust Boundary)

The core software components of the Notarization API system.

Title	Priority	Status	Description	Mitigations
Data flow should use HTTP/S	High	Mitigated by Admin	All requests are made over the public internet and could be intercepted by an attacker.	Mitigated by the administrator: 1. The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.
Credential theft	Medium	Mitigated by Admin	An external attacker could use stolen credentials to make unauthorised queries against internal components.	Mitigated by the administrator: 1. Use ingress firewalls to restrict access to the internal services. 2. Use network mesh to restrict access between components such that only required service-to-service communication is permitted. 3. Use network mesh to encrypt all communication in transport to prevent leakage.
Brute force request information	Medium	Mitigated by Admin	An external attacker could use brute force to discover the endpoints and thus information of the backend services.	Mitigated by the administrator: 1. Use ingress firewalls to restrict access to the internal services. 2. Use network mesh to restrict access between components such that only authenticated service-to-service communication is permitted. 3. Use network mesh to encrypt all communication in transport.
Log files leak	Medium	Mitigated by Admin	Log files could contain sensitive information.	Mitigated by the application: 1. Log files only contain strictly necessary information. Mitigated by the administrator: 1. Access to log files is restricted. 2. Log level is appropriate.
Distributed Denial of Service Attack	Medium	Mitigated by Admin	A huge number of requests may lead to denial of service (DoS).	Mitigated by the administrator: 1. Implement additional web application firewalls (WAF). 2. Implement rate limiting and quota on the API gateway to mitigate DDOS attacks. 3. Improve availability with redundancy measures.
Outdated components enable elevated privileges	Medium	Mitigated by Admin	Outdated software components with vulnerabilities give rise to elevation of privilege.	Mitigated by the administrator: 1. Regularly check for vulnerabilities and patch accordingly.
Run operating system commands to elevate privileges	High	Mitigated by Admin	A malicious user could run operating system commands through injection attacks to elevate privileges and perform remote code execution	Mitigated by the administrator: 1. Ensure the server process is running with the principle of least privilege 2. Use jailing and sandboxing mechanisms whereever applicable 3. Ensure input validation routines are in place to allow only known data (whitelisting)

Audit Log DB (Store)

Title	Priority	Status	Description	Mitigations

Title	Priority	Status	Description	Mitigations
Unauthorized data access through database files	Medium	Mitigated By Admin	An attacker could steal audit data by stealing database files.	Mitigated by administrator: 1. Set proper privileges on the database files. 2. Run the database server with the principle of least privilege. 3. Encrypt the database at rest. 4. Store the decryption keys in a Hardware Security Module (HSM).

DBMS (Store)

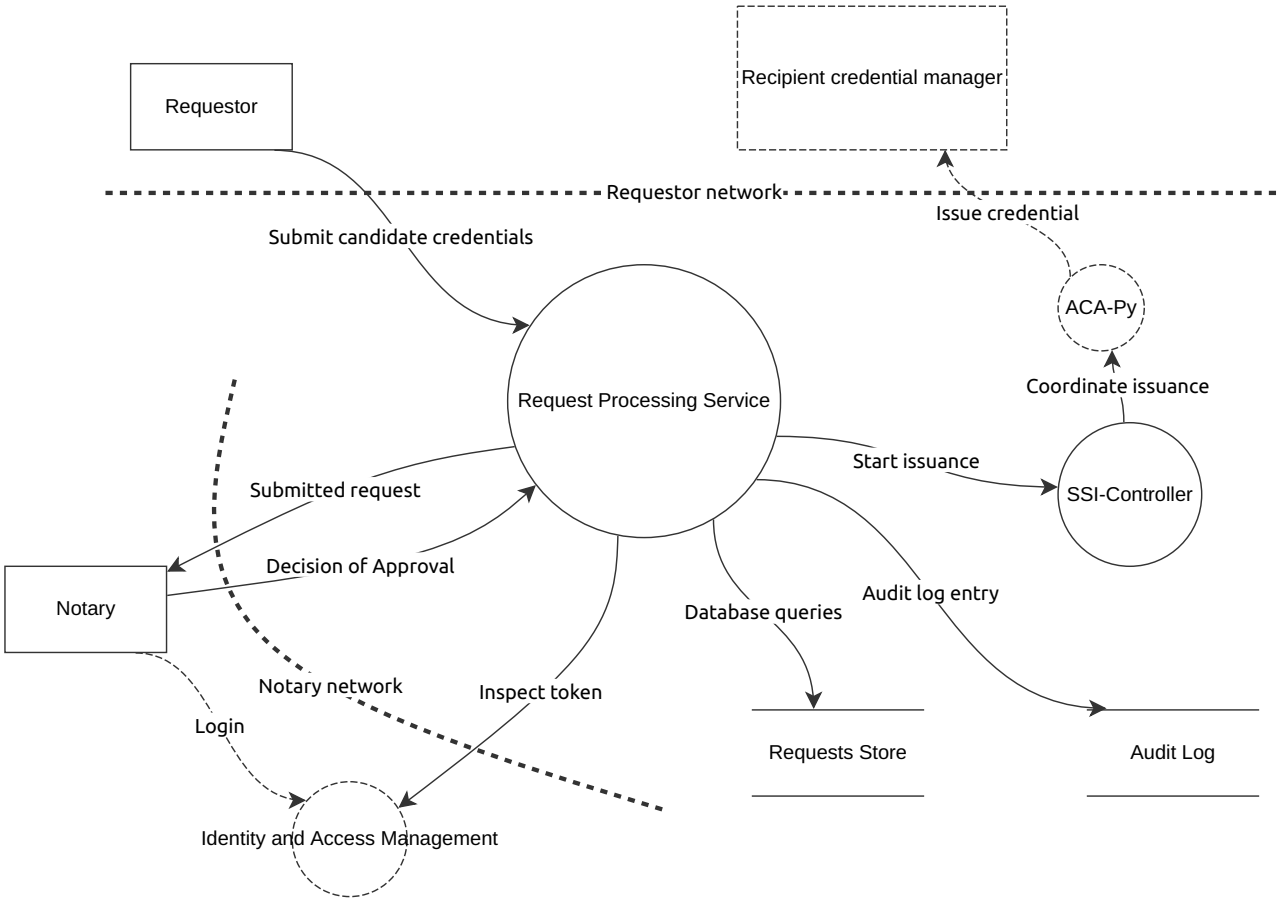
The database management service and the individual databases it provides.

Title	Priority	Status	Description	Mitigations
Unauthorized data access through database files	Medium	Mitigated By Admin	An attacker could steal request data or issuing keys by stealing database files.	Mitigated by administrator: 1. Set proper privileges on the database files. 2. Run the database server with the principle of least privilege. 3. Encrypt the database at rest. 4. Store the decryption keys in a Hardware Security Module (HSM).

Scheduler (Actor)

Title	Priority	Status	Description	Mitigations
Triggers not called	Medium	Mitigated	An attacker might disrupt this service, preventing this service from triggering required operations in other services (such as clean-up or renewal).	Mitigated by administrator: 1. Configure the monitoring and alerting system to the notify the administrator when the scheduled operations fail unexpectedly often, or do not execute at all.

Minimal request submission and VC issuance.



Minimal request submission and VC issuance.

Database queries (Data Flow)

Queries to persist and retrieve the request information.

Title	Priority	Status	Description	Mitigations
Man in the middle attack	Low	Mitigated by Admin	An attacker could intercept the DB queries in transit and obtain sensitive information, such as DB credentials, query parameters or query results (is unlikely since the data flow is over a private network).	Mitigated by the administrator: 1. Enforce an encrypted connection to the DB server, such as via a network mesh.

Submitted request (Data Flow)

The submitted notarization request including the details of the candidate credential are presented to the notary for reviewing.

Title	Priority	Status	Description	Mitigations
Data flow should use HTTP/S	High	Mitigated by Admin	These responses are over the public internet and could be intercepted by an attacker.	Mitigated by the administrator: 1. The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.
Spoofing	High	Mitigated by Admin	An attacker could spoof the notary.	1. Enforce server side checks for privileges by actively validating the user. 2. The identity of the notary is established using a secure authentication protocol, OpenID-connect. Mitigated by the administrator: 1. Access to notary credentials are safely and securely stored.

Submit candidate credentials (Data Flow)

The request is submitted

Title	Priority	Status	Description	Mitigations
Tamper with HTTP request data	Medium	Mitigated	Tamper with HTTP request data to alter requested credentials.	1. Enforce server side checks for priveleges using an access token issued once per credential submission request.
Brute force request information	Medium	Mitigated	An external attacker could use brute force to discover the endpoints and thus information of the submitted credential requests.	1. Enforce server side checks for priveleges using a unique access token issued once per credential submission request. 2. The access token cannot be requested via any API. 3. Use random identifiers (UUID) instead of sequential values
Data flow should use HTTP/S	High	Mitigated by Admin	These requests are made over the public internet and could be intercepted by an attacker.	Mitigated by the administrator: 1. The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

Decision of Approval (Data Flow)

The notary submits their approval or rejection of the notarization request. The approval leads to the issuance of the requested credentials.

Title	Priority	Status	Description	Mitigations
Data flow should use HTTP/S	High	Mitigated by Admin	These responses are over the public internet and could be intercepted by an attacker.	Mitigated by the administrator: 1. The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

Requests Store (Store)

Title	Priority	Status	Description	Mitigations
Unauthorised access	High	Mitigated by Admin	An attacker could make an query call on the DB.	Mitigated by the administrator: 1. Require all queries to be authenticated.
Credential theft	Medium	Mitigated by Admin	An attacker could obtain the DB credentials and use them to make unauthorised queries.	Mitigated by the administrator: 1. Use network mesh to restrict access to the database only to the request submission service. 2. Use network mesh to encrypt all communication in transport.

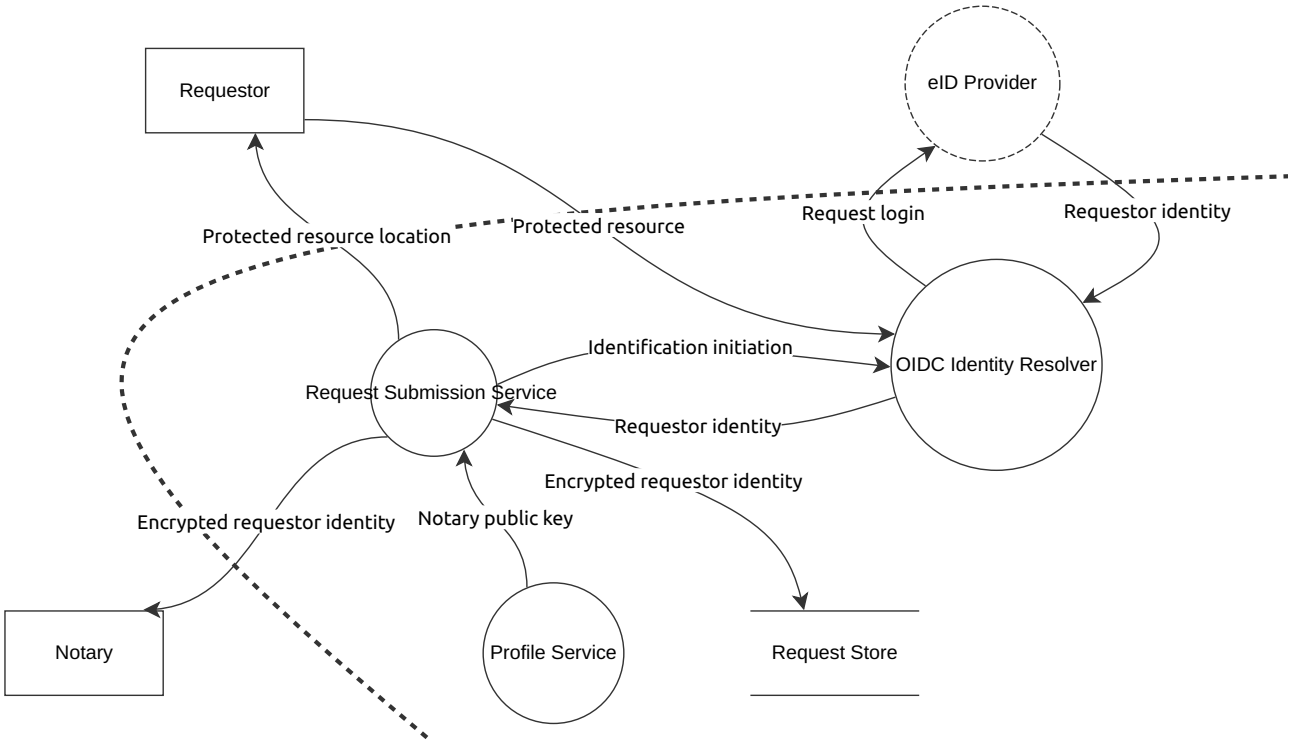
SSI-Controller (Process)

Title	Priority	Status	Description	Mitigations
Log files leak	Medium	Mitigated by Admin	Log files could contain sensitive information.	Mitigated by the application: 1. Log files only contain strictly necessary information. Mitigated by the administrator: 1. Access to log files is restricted. 2. Log level is appropriate.
Private key leak	Medium	Mitigated by Admin	Private keys used to issue credentials could be leaked.	Mitigated by the administrator: 1. Protect private keys with suitable passwords. 2. Restrict access to private keys.
Issued credentials may be repudiated.	Medium	Mitigated	Issued credentials may be repudiated.	The service issues verifiable credentials.
Outdated components enable elevated privileges	Medium	Mitigated by Admin	Outdated software components with vulnerabilities give rise to elevation of privilege.	Mitigated by the administrator: 1. Regularly check for vulnerabilities and patch accordingly.

Audit Log (Store)

Title	Priority	Status	Description	Mitigations
Unauthorised access	Medium	Mitigated By Admin	An attacker could make an query call on the DB.	Mitigated by the administrator: 1. Require all queries to be authenticated.

Requestor Identification Process



Requestor Identification Process

Identification initiation (Data Flow)

Title	Priority	Status	Description	Mitigations
	Medium	Mitigated By Admin	An attacker could modify the callback URLs	Mitigated by the application: 1. Whitelist the callback URLs before accepting an identification initiation request. Mitigated by the administrator: 1. Use network mesh to restrict access to the OIDC identity resolver only to the request submission service. 2. Use network mesh to encrypt all communication in transport.

Requestor identity (Data Flow)

The attributes of the electronic identity of the requestor.

Title	Priority	Status	Description	Mitigations
Requestor identity leak	Medium	Mitigated By Admin	An attacker could intercept the requestor identity.	Mitigated by the administrator: 1. Use HTTPS to encrypt all communication in transport.

Requestor identity (Data Flow)

The attributes of the electronic identity of the requestor.

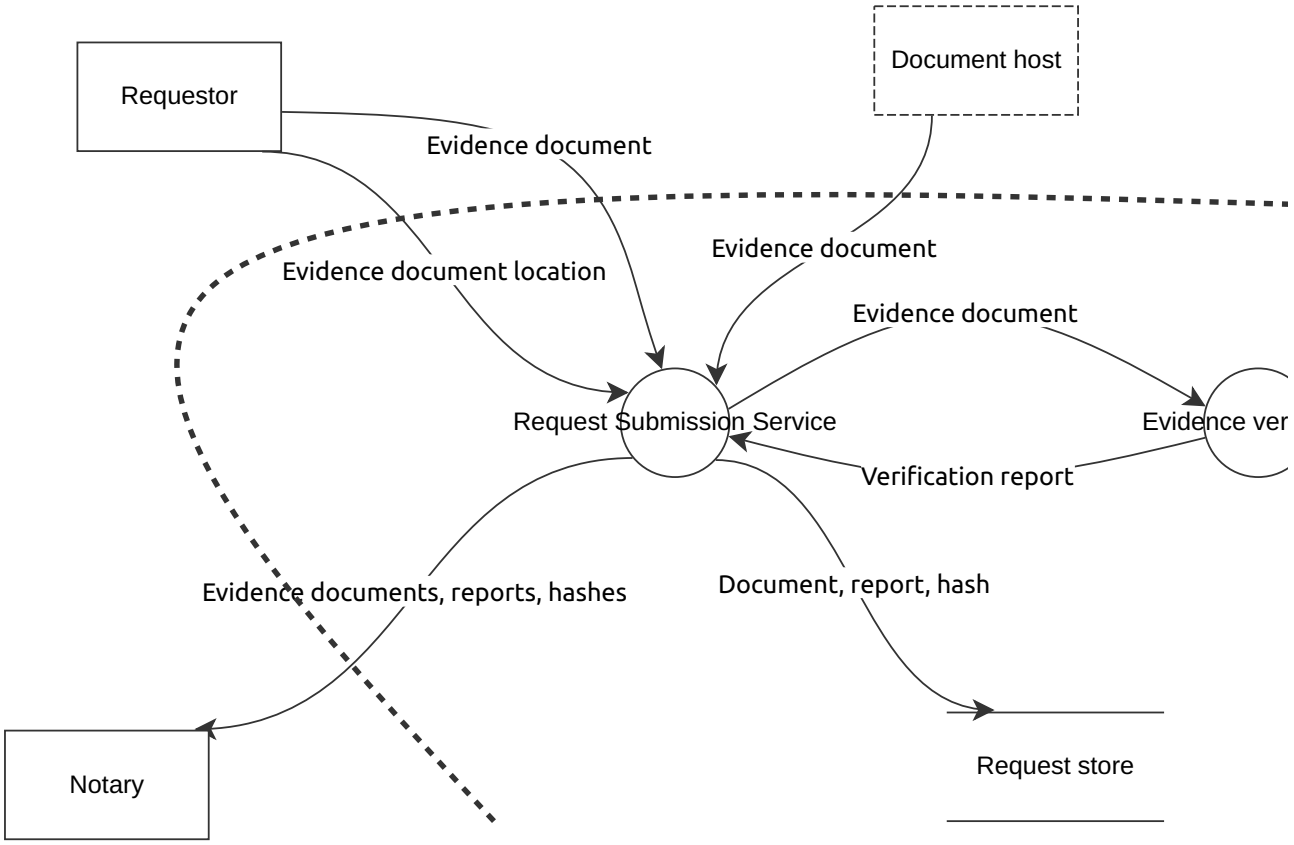
Title	Priority	Status	Description	Mitigations
Requestor identity leak	Medium	Mitigated By Admin	An attacker on the internal network could intercept the requestor identity.	Mitigated by the administrator: 1. Use network mesh to encrypt all communication in transport.

Notary public key (Data Flow)

The notary public key.

Title	Priority	Status	Description	Mitigations
Fake notary key insertion	Medium	Mitigated By Admin	An attacker could insert fake notary keys to ensure the identity can be encrypted with their own keys.	Mitigated by the administrator: 1. Use network mesh to authenticate the profile service by the request submission service. 2. Use network mesh to encrypt all communication in transport.

Evidence document processing



Evidence document processing

Verification report (Data Flow)

Title	Priority	Status	Description	Mitigations
Evidence report tampering	Medium	Mitigated By Admin	An attacker could tamper with the verification report.	Mitigated by the administrator: 1. Use network mesh to encrypt all communication in transport.