

***CMI 4.x Security Audit***

***Neuron/Cortex +***

***Open API***

***Test description***

*Version 1.0.1 from 29.06.2022*

**Table of Contents**

1 General 3

2 Technical Overview 4

2.1 Server architecture 4

2.2 Servers / IP’s / endpoints 4

2.3 Hermes/ESB/Merlin Architecture 6

2.4 Status workflow (Test details in Chapter 5) 6

2.5 Statement workflow (Test details in Chapter 5) 6

2.6 Open API: Statement workflow 7

3 Audit topic #1: Vulnerability scans 7

4 Audit topic #2: Open API Tests 7

4.1 Swagger definition 8

4.2 Testdata 8

4.2.1 Client credentials and accounts 8

4.2.2 Positive Testscenarios 8

4.2.3 Negative Testscenarios 10

4.2.4 Signing and encryption mechanism 10

5 Audit topic #3: NWU API Tests 12

5.1 Swagger definition 12

5.1.1 Status API 12

5.1.2 Statement API 12

5.2 Testdata 13

5.2.1 Client credentials and accounts 13

5.2.2 Positive Testscenarios 13

5.2.3 Negative Testscenarios 13

6 Test preparation 13

6.1.1 Required access rights and users 13

7 Document history 14

# General

RBI has implemented two critical changes to the infrastructure of the CMI landscape:

* Neuron/Cortex: Based on an architectural decision we have migrated our API integration layer from “Hermes” to a new solution based on Neuron/Cortex. As you might remember, Hermes was based on webMethods technology deployed on AWS (Merlin) and mainly used for the API based communication between the CMI application and the Network Banks. Our new layer is also deployed on AWS but its underlying technology is based on Neuron/Cortex. The new platform itself was already audited, but we need to test all CMI applications built on the platform.
* We have introduced a new “Open API” Channel which will provide banking APIs to TPPs or directly to clients. The APIs will be exposed via the AWS Gateway and the microservices are running on the Neuron/Cortex layer.



# Technical Overview

## Server architecture

For a detailed plan of server architecture please see the following diagram:



The AWS API gateway is deployed in a Merlin account. All the rest is running in a Kubernetes/Openshift platform called Cortex combined with a Pulsar event broker called Neuron.

## Servers / IP’s / endpoints

Merlin account

* HERMES-TEST / SIT050
* Endpoint: <https://vfth80l859.execute-api.eu-central-1.amazonaws.com/SIT050/api/v1>
* Cognito IDP: <https://rbi-hermes-test-userpool.auth.eu-central-1.amazoncognito.com/oauth2/token>
* Installed components
  + AWS API Gateway
  + Cognito IDP
  + haproxy

Neuron/Cortex

* AWS account adc-test (866255434500)
* Kubernetes project/namespace: prod01-fcmi
* OpenAPI endpoint: <https://vfth80l859.execute-api.eu-central-1.amazonaws.com/SIT050/api/v1>
* Endpoint for NWU services: <https://cmi-routing.mesh.cortex-test01.adc-test.internal.rbigroup.cloud>
* IP ranges for endpoints
  + 10.223.254.192/27
  + 10.223.254.224/27
  + 10.223.52.192/27
* IDP endpoint: <https://idp-d.rbinternational.com/as/token.oauth2>
* Components (all of them instantiated as Kubernetes PODs)
  + Pulsar functions
    - addOmikronHeader2Statement
    - extractMinimumMetadata
    - validateAndMapPayment
    - validateAndMapStatus
    - addOmikronHeader2Status
    - mapPaymentResponse
    - validateAndMapStatement
  + Pulsar sinks and sources
    - MQCreditRequest
    - MQDebitRequest
    - MQStatementRequest
    - MQStatusRequest
    - MQCreditResponse
    - MQDebitResponse
    - MQStatementResponse
    - MQStatusResponse
  + Microservices
    - prod01-cmi-openapi: OpenAPI implementation
    - prod01-cmi-routing: NWU API implementation
  + Databases
    - cmi-cmi: Postgresql db instance for openapi
* IP subnets for PODs
  + 10.223.254.0/26
  + 10.223.254.64/26
  + 10.223.254.128/26

The vulnerability scans should cover TEST environments.

This includes Neuron/Cortex, Hermes

The API scans should cover only TEST environments.

We agreed with RBI security to focus on the TEST environment only.

## Hermes/ESB/Merlin Architecture



## Status workflow (Test details in Chapter 5)

1. Status message is generated in the NWU core system (9)
2. Status message is forwarded via Status API to the Internal CMI RT Layer (7)
3. The CMI RT Layer (7) is mapping the JSON to the XML
4. Status message is forwarded from CMI RT Layer (7) via MQ to CMI Bankserver (3)

## Statement workflow (Test details in Chapter 5)

1. Account statement is generated in the NWU core system (9)
2. Account statement is forwarded via Statement API to the Internal CMI RT Layer (7)
3. The CMI RT Layer (7) is mapping the JSON to the XML
4. Account statement is forwarded from CMI RT Layer (7)
   1. via MQ to CMI Bankserver (3)
   2. to the CM Open API Database (8) – only if the account from the account statement is configured in the Open API DB

## Open API: Statement workflow

1. Client application (2) is calling the IDP (5) with his ClientID/Secret to receive the token
2. Client application (2) is calling the RBI CM API Gateway (4) with the token and request
3. The API Gateway (4) is verifying the token and request
4. The Open API CMI Layer (6) is validating if the User ID has the rights to view statements for this account
5. Then the response is sent back to the client application (2)

# Audit topic #1: Vulnerability scans

The following components are in scope:

* Neuron/Cortex with namespace prod01-fcmi
* hermes-test
  + cmi-openapi-gateway
  + HA Proxy
  + Cognito user pool Hermes-test\_userpool with app integration users cmi-openapi-user000[1-8]
  + Out of Scope: IDMS API Gateway

The vulnerability scans should cover TEST environment.

Scanning should take place from audit clients within VPN / RBI LAN against.

Temporary firewall rules have been created to ensure access to all ports and servers.

For the architecture diagrams please see chapter 2.1.

For the servers / IP’s / endpoints please see chapter 2.2.

# Audit topic #2: Open API Tests

The following tests are in scope:

* Open API Endpoint scans for statement API at HERMES
* Open API Endpoint scans for status API at HERMES

The vulnerability scans should cover TEST environment only.

For the architecture diagrams please see chapter 2.1.

For the servers / IP’s / endpoints please see chapter 2.2.

Furthermore, the API specification & concept should be checked and also technical API tests should take place. Those tests should cover typical API scans for HERMES endpoints

Example for typical API tests

* Test authentication method (e.g. Try to bypass the authentication method)
* Test for SSRF vulnerability when API request is sending URL parameters.
* Test for XML attacks
* Test IDs in bodies and headers
* Test for misconfigured HTTP headers, unnecessary HTTP methods, permissive Cross-Origin resource sharing (CORS), and verbose error messages containing sensitive information.
* Test for injection attacks
* Validate API responses (Check if endpoints might leak excessive data that should not be accessible by the user)
* Code review of the API
* Test for Missing Object Level Access Control
* Test for Missing Function/Resource Level Access Control

## Swagger definition

 

## Testdata

### Client credentials and accounts

* ClientID: 1qt04fv8k6tb8mrvabqo974ol2
* ClientSecret: 8df06gd67jertkf5lfius3qutl2evb86s39bnrq6qln635jbqrv
* Accounts connected to the user:
  + - RS35265110031000049131
* Certificate to validate signed and encrypted API request. These secrets are used on RBI side to sign payloads sent to customers and to decrypt payloads received from customers.

|  | **Test env** |
| --- | --- |
| CN | cmisign.mesh.cortex-test01.adc-test.internal.rbigroup.cloud |
| Issuer | CN=R3,O=Let's Encrypt,C=US |
| crt |  |

### Positive Testscenarios

*The requests to the Statement api require a valid token and it is recommended a unique X-Request\_ID to facilitate the debugging.*

**Get token**

curl -L -X POST "<https://rbi-hermes-test-userpool.auth.eu-central-1.amazoncognito.com/oauth2/token>" -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Basic TO\_REPLACE" -H "Cookie: XSRF-TOKEN=c80dbec2-1601-4419-9f57-0f0c8da496ec" --data-urlencode "grant\_type=client\_credentials"

**Retrieve customer's accounts statements**

curl -L -X GET "<https://vfth80l859.execute-api.eu-central-1.amazonaws.com/SIT050/api/v1/statements?DateFrom=2022-05-31T00:00:00Z&DateTo=2022-06-04T23:59:59Z>" -H "X-Request-ID: 4a55c16f-b809-427d-81ec-ae8d3eb06ec9" -H "Authorization: Bearer TO\_REPLACE\_WITH\_TOKEN"

Expected response:

* + Http 200
  + Json format
  + 12 statements

**Retrieve statements of a given account**

curl -L -X GET "https://vfth80l859.execute-api.eu-central-1.amazonaws.com/SIT050/api/v1/accounts/RS35265110031000049131/statements?DateFrom=2022-05-31T00:00:00Z&DateTo=2022-06-04T23:59:59Z&Size=100" -H "X-Request-ID: 9699e5a1-0a80-4cd6-b8c0-50fb68b624b5" -H "Authorization: Bearer TO\_REPLACE\_WITH\_TOKEN"

Expected response (same than the previous request):

* + Http 200
  + Json format
  + 12 statements

**Retrieve statement content**

curl -L -X GET "<https://vfth80l859.execute-api.eu-central-1.amazonaws.com/SIT050/api/v1/statements/09cdea2ffba0415f9d7e1c42422f2275>" -H "X-Request-ID: eca74462-64fc-44c0-a103-4b310635f2f9" -H "Authorization: Bearer TO\_REPLACE\_WITH\_TOKEN"

Expected response (same as the previous request):

* + Http 200
  + Json response
  + StatementId: 09cdea2ffba0415f9d7e1c42422f2275
  + LegalSequenceNumber:155
  + PageNumber: 1

**Retrieve latest statement**

curl -L -X GET "<https://vfth80l859.execute-api.eu-central-1.amazonaws.com/SIT050/api/v1/statements/latest?StatementFormat=CAMT52>" -H "X-Request-ID: dc6149d5-fdc6-4810-9b48-257ba02a3eb1" -H "Authorization: Bearer TO\_REPLACE\_WITH\_TOKEN"

Expected response (same as the previous request):

* + Http 200
  + Json response

### Negative Testscenarios

The above provided user is only allowed to access the accounts which are connected to him, please verify that it is not possible to retrieve any other data for the API.

Other accounts:

* RS35265110031000028470
* RS35265100000000023137
* RS35265111031000135120
* RS35265401031000331498
* RS35265104031000094924
* RS35265621031000467162
* RS35265361031000061122
* RS35265111031000108833
* RS35265110031000010719
* XK051501001015360624
* XK051501001005492038
* XK051501001002060469
* XK051501001000060038
* XK051501001002639559
* XK051501001004322218

Other valid ClientIDs (the ClientSecret is not provided):

* 5qitfdb2aaoan3dgji6c8hflup
* 23j62iarp4albi1oulvn901cl5
* 3vq0j0j1jf27e68j8l1l7jdfuq
* 1evpjuq46rib4nvaakgo0ogdag
* 4pfkoaafbb2gfaf17ilkpjjqrc
* 52rhu4l0thqjm9guf6r8vnblh2
* 7lv038e5u7k3qcf38s6aha2b51

### Signing and encryption mechanism

Additionally to authentication via JWT tokens, it has been implemented a payload signing and encryption mechanism.

The certificates itself are header parameters, we only have to exchange CN names of our private keys and the trust chain. Our details can be found in the Test data section.

From you, we will need the same (private key + trust chain).

The encryption follows the JOSE JWS standard, as overview, the process:

* Converts the body of a rest api call in 3 base64 encoded sections:
  + Header with basic information about used encryption algorithms
  + body with encrypted and base64 encoded payload signature

**Header fields in response**

* Signature

keyId=<key-id>,algorithm="rsa-sha256",headers="Content-Type digest",signature=<signature-string>

<key-id> should be the CN of the certificate used to sign the message

* Digest

Hashcode of the message payload using the algorithm provided in Signature

* Certificate

Public key used to sign the message

* Encryption (only used for payment requests)

keyId=<key-id>,algorithm="rsa-sha256"

<key-id>: CN of certifcate used by RBI

**Header fields in response**

* Signature

keyId=<key-id>,algorithm="rsa-sha256",headers="Content-Type digest",signature=<signature-string>

<key-id> is the CN of the certificate used to sign the message

* Digest

Hashcode of the message payload using the algorithm provided in Signature

* Certificate

Public key used to sign the message

* Encryption

keyId=<key-id>,algorithm="rsa-sha256"

<key-id>: CN of certifcate received in the Signature header field

**Encrypted parameters**

StatementEncoding:

type: string

description: 'Defines encoding of xml message.

\* BASE64: XML message is base64 encoded (default value)

\* ENCRYPTED: XML message is encrypted with encryption key and algoritm as defined in header attribute Encryption and base64 encoded'

enum:

- BASE64

- ENCRYPTED

EncodedStatement:

type: string

description: Base64 encoded xml message of the requested statement

example: 'PD94bWwgdmVyc2lvbj0iMS4wIiBlbm...'

# Audit topic #3: NWU API Tests

The following tests are in scope:

* NWU API Endpoint scans for status & statement API at Neuron/Cortex

The vulnerability scans should cover TEST environment only.

For the architecture diagrams please see chapter 2.1.

For the servers / IP’s / endpoints please see chapter 2.2.

Furthermore, the API specification & concept should be checked and also technical API tests should take place. Those tests should cover typical API scans for API endpoints

Precondition for testing:

* AD user in Group AD to get a valid token

Example for typical API tests

* Test authentication method (e.g. Try to bypass the authentication method)
* Test for SSRF vulnerability when API request is sending URL parameters.
* Test for XML attacks
* Test IDs in bodies and headers
* Test for misconfigured HTTP headers, unnecessary HTTP methods, permissive Cross-Origin resource sharing (CORS), and verbose error messages containing sensitive information.
* Test for injection attacks
* Validate API responses (Check if endpoints might leak excessive data that should not be accessible by the user)
* Code review of the API
* Test for Missing Object Level Access Control
* Test for Missing Function/Resource Level Access Control

## Swagger definition

### Status API



### Statement API



## Testdata

### Client credentials and accounts

* Client crendential

### Positive Testscenarios

#### Status API

* Requests from Networkunit Serbia:



* Requests from Networkunit Kosovo:



#### Statement API

* Requests from Networkunit Serbia:



* Requests from Networkunit Kosovo:



### Negative Testscenarios

API calls should only be possible with the AD User and proper authentication, please verify that.

# Test preparation

### Required access rights and users

* AD user for Group AD
* Cognito User for API Gateway
* Access to Hermes (TBC)
* Access to OpenShift Console (TBC)

# Document history

| Date | Creator | Version | Comment |
| --- | --- | --- | --- |
| 01.05.2022 | Philipp Navratil | 0.1 | Initial creation of document  + Input from Mario Vogelsberger / Andreas Halwein |
| 01.06.2022 | Philipp Navratil | 0.9 | Formatting and adding Testdata + Scenarios |
| 09.06.2022 | Alberto Diaz | 0.91 | Added data for positive and negative test scenarios |
| 27.06.2022 | Alberto Diaz | 1.0.0 | Updated OpenAPi swaggers |
| 29.06.2022 | Alberto Diaz | 1.0.1 | Added “Signing and encryption mechanism” section |