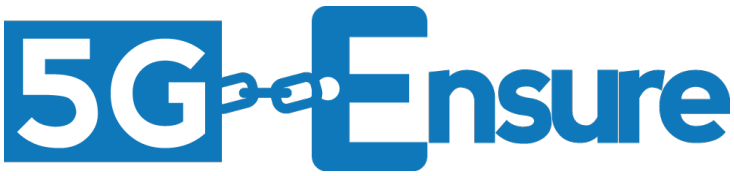


Deliverable D3.8
5G-PPP Security Enablers Documentation (v2.0)
Enabler System Security State Repository (SSSR)

Project name	5G Enablers for Network and System Security and Resilience	
Short name	5G-ENSURE	
Grant agreement	671562	
Call	H2020-ICT-2014-2	
Delivery date	31.08.2017	
Dissemination Level:	Public	
Lead beneficiary	NEC	Felix Klaedtke, felix.klaedtke@neclab.eu
Authors	IT Innovation Centre: Maxim Bashevoy, Toby Wilkinson	

Document Version	Date	Change(s)	Author(s)
0.1	02.06.2017	Created template	Felix Klaedtke
0.2	02.08.2017	First draft	Maxim Bashevoy
0.3	8.8.2017	Reviewed	Aleksi Dahl
0.4	25.08.2017	Final version	Maxim Bashevoy

Foreword

5G-ENSURE belongs to the first group of EU-funded projects which collaboratively develop 5G under the umbrella of the 5G Infrastructure Public Private Partnership (5G-PPP) in the Horizon 2020 Programme. The overall goal of 5G-ENSURE is to deliver strategic impact across technology and business enablement, standardization and vision for a secure, resilient and viable 5G network. The project covers research and innovation - from technical solutions (5G security architecture and testbed with 5G security enablers) to market validation and stakeholders engagement - spanning various application domains.

The System Security State Repository (SSSR) enabler consumes monitoring events from the Generic Collector Interface enabler to provide security information about a runtime system. It uses the same technologies and models as the Trust Builder enabler. This software release 2 is the first release of the SSSR.

Disclaimer

Copyright notice

Contents

1 Introduction

The Secure System State Repository captures the state of the instantiated network and issues events about assets not compliant with the Design-Time System Model based on monitoring data. The Repository also provides a visualisation of the current state of the system, specifically which run-time assets in each asset class specified in the Design-Time System Model are known to be present. Figure 1 represents an outline of the Repository architecture for R2:

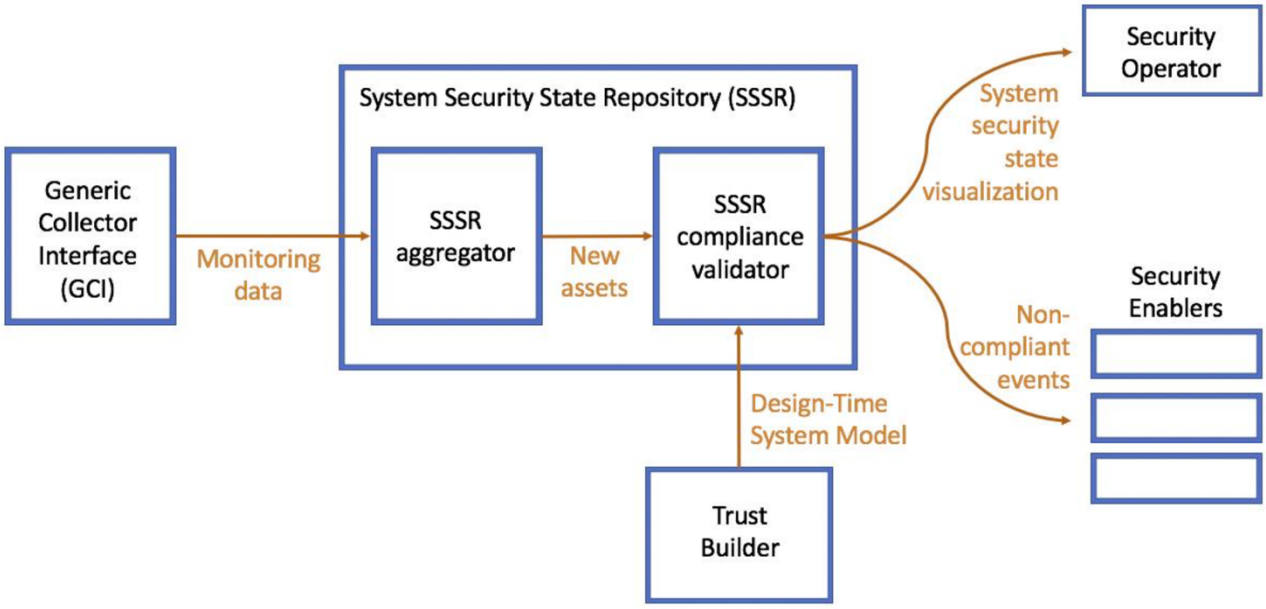


Figure 1. Basic concepts of the System Security State Repository

This manual is organised as follows. Quick installation and configuration instructions can be found in the next section. Section 3 contains detailed user guide for SSSR. Some basic tests and abbreviations constitute sections 4 and 5.

Current implementation of the system described above uses Distributed Data Protocol (DDP) [1] for the following communications:

1. Between the UI and the back end of GCI components simulator, GCI and SSSR
2. Between the back ends of GCI components simulator and the GCI simulator; GCI simulator and SSSR

Communication between the SSSR and the Trust Builder uses TB's REST API.

2 Installation and Administration Guide

2.1 System Requirements

This enabler runs on any host (preferably Linux) with Docker installed. Trust Builder enabler has to be running on the system with its endpoint accessible via a URL.

2.2 Enabler Configuration

The SSSR requires a valid URL to a Trust Builder endpoint.

2.3 Enabler Installation

SSSR enabler requires separate Trust Builder installation, which has to be installed first.

2.3.1 Trust Builder with SSSR model

Prerequisites:

- Docker

Unzip provided archive and change into the created folder:

```
unzip trust_builder_dist.zip
cd trust_builder_dist
```

Then create 5g-ensure docker network (only once):

```
docker network create -d bridge 5g-ensure-trust-builder
```

Start mongo container:

```
docker run \
--name 5g-ensure-mongo \
--network 5g-ensure-trust-builder \
--rm -ti -p 27017:27017 mongo
```

Confirm that Mongo database server started successfully by locating the following line in the output:

```
2017-08-22T08:46:23.998+0000 I NETWORK  [thread1] waiting for connections on port 27017
```

Start Trust Builder with the following command:

```
docker run \
--name 5g-trust-builder \
--network 5g-ensure-trust-builder \
--link 5g-ensure-mongo:mongo \
-v "$PWD"/trust-builder.war:/usr/local/tomcat/webapps/trust-builder.war \
--rm -ti \
-p 8080:8080 \
-e spring.data.mongodb.host=mongo \
-e server.externalUrl=http://localhost:8080 \
tomcat:7-jre8
```

Confirm that Trust Builder started successfully by locating the following line in the output:

```
08:46:51.432        INFO    u.a.s.i.s.s.SystemModellerApplication:57:    Started SystemModellerApplication in 10.075 seconds (JVM running for 14.595)
```

```
08:46:51.445 DEBUG u.a.s.i.s.s.a.JwtAuthenticationTokenFilter:177: Initializing filter 'authenticationTokenFilterBean'
08:46:51.446 DEBUG u.a.s.i.s.s.a.JwtAuthenticationTokenFilter:202: Filter 'authenticationTokenFilterBean' configured successfully
...
Aug 22, 2017 8:46:51 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-apr-8080"]
Aug 22, 2017 8:46:51 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-apr-8009"]
Aug 22, 2017 8:46:51 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 14397 ms
```

You should be able to access the following endpoint in your browser at this stage:

1. Trust Builder: http://localhost:8080/trust-builder/dashboard

Please follow Load model unit test in Section 4.1 to get read-only model ID needed for the next section. In the dashboard locate **Read** button and click on it:

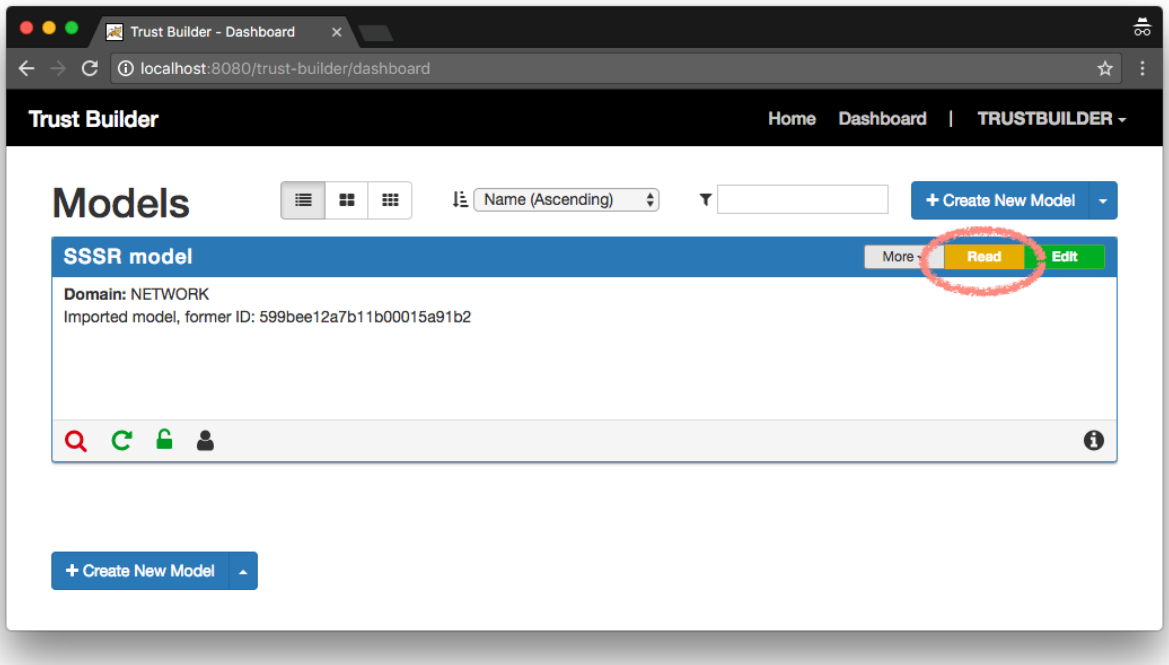


Figure 2. SSSR sample model loaded in Trust Builder with a link to read-only view highlighted

Copy the long alpha-numeric string between “model/” and “/read” pars of the URL:

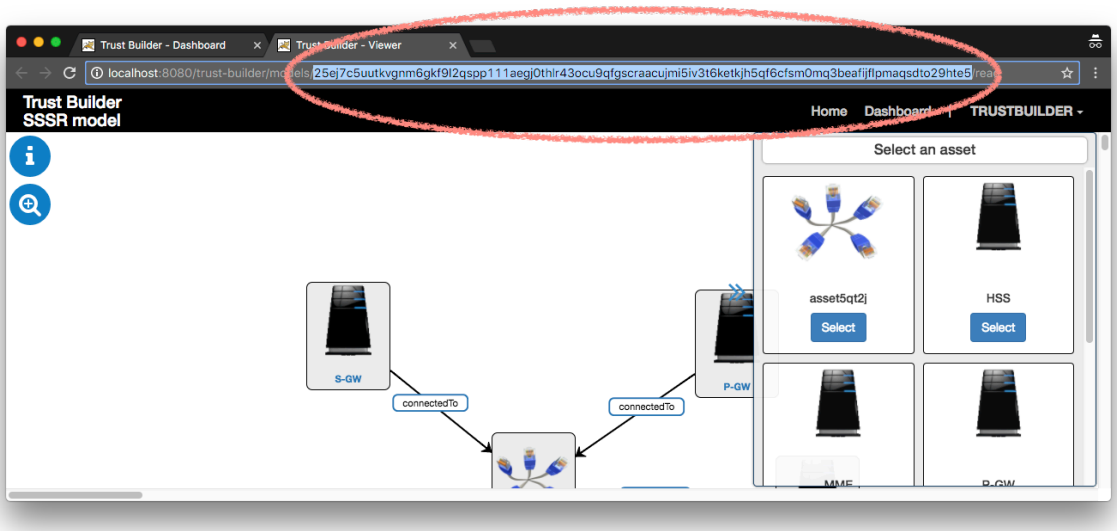


Figure 3. Read-only view of the sample SSSR model open in Trust Builder with the read-only model ID highlighted in browser URL

That string will be needed during SSSR setup, which is detailed in the following section.

2.3.2 SSSR

Prerequisites:

- Docker

Unzip provided archive and change into the created folder:

```
unzip sssr_dist.zip
cd sssr_dist
```

Go back to **sssr_dist/** and build **5g-ensure/sssr:v1** docker image (only once):

```
docker build -t 5g-ensure/sssr:v1 .
```

Replace read-only model ID in **sssr/settings.json** with the one obtained in the previous section:

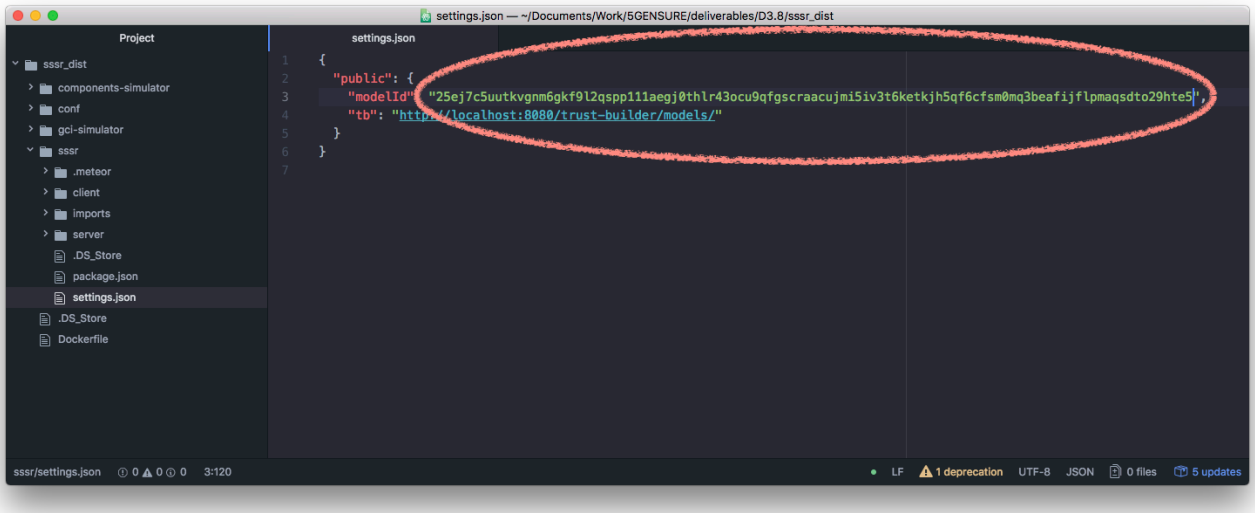


Figure 4. SSSR settings file with Trust Builder's read-only model ID highlighted

Run SSSR with the following command:

```
docker run --rm -it --name 5g-sssr \
-p 3001:3001 -p 3010:3010 -p 3020:3020 \
-v "$PWD"/sssr/settings.json:/app/sssr/settings.json \
5g-ensure/sssr:v1
```

The output should read:

```
2017-08-24 15:08:04,715 INFO success: sssr entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2017-08-24 15:08:04,715 INFO success: components-simulator entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2017-08-24 15:08:04,715 INFO success: gci-simulator entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
```

Wait (it will take over a minute) for the SSSR to start, output of the following command (in a new terminal window):

```
docker exec -it 5g-sssr bash -c 'tail -f /app/logs/sssr'
```

should have the following lines:

```
=> Started your app.
=> App running at: http://localhost:3020/
```

when the SSSR is up.

You should be able to access the following new endpoints in your browser at this stage:

1. GCI components simulator: http://localhost:3001
2. GCI simulator: http://localhost:3010
3. SSSR: http://localhost:3020

See section 3 for detailed user guide.

2.4 Troubleshooting

The start up sequence should be the following:

- Trust Builder
- SSSR and components

Browser console can be used to identify connection and timeout issues. Server logs can ensure correct communication between all parts of the system.

3 User and Programmer Guide

3.1 User Guide

3.1.1 Overview and setup

The complete running system consists of:

- Trust builder (running in the background)

- GCI components simulator
- GCI simulator
- SSSR

For a sample layout displaying three main parts of the system, see figure below:

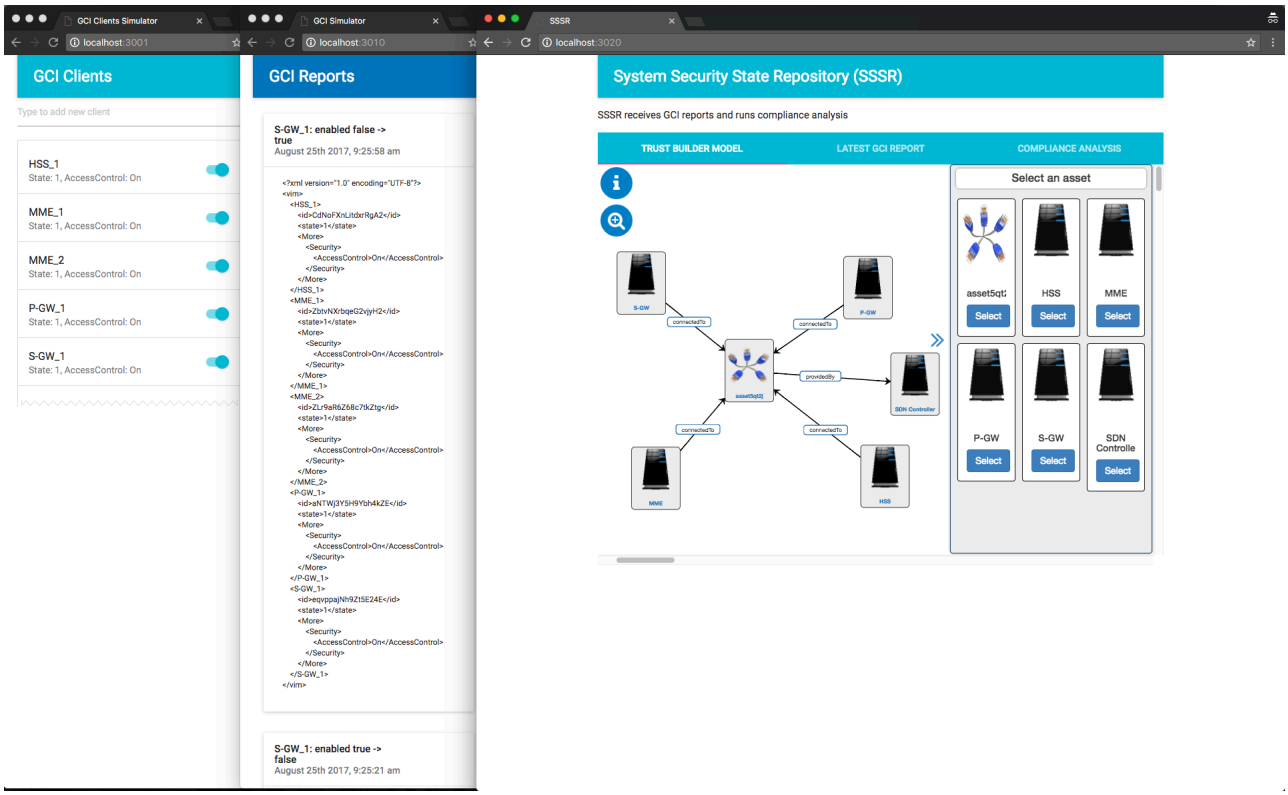


Figure 5. Basic system layout (from left to right): GCI components simulator, GCI simulator, SSSR

Three browser windows in the figure above represent (from left to right): GCI components simulator with five sample components from a sample Mobile Network; GCI simulator with two latest generated GCI reports; main view of the SSSR with a sample Mobile Network loaded from the Trust Builder.

Start by logging into the Trust Builder and importing [2] a “Model_for_SSSR_validated.nq” model provided as described in Section 4.1:

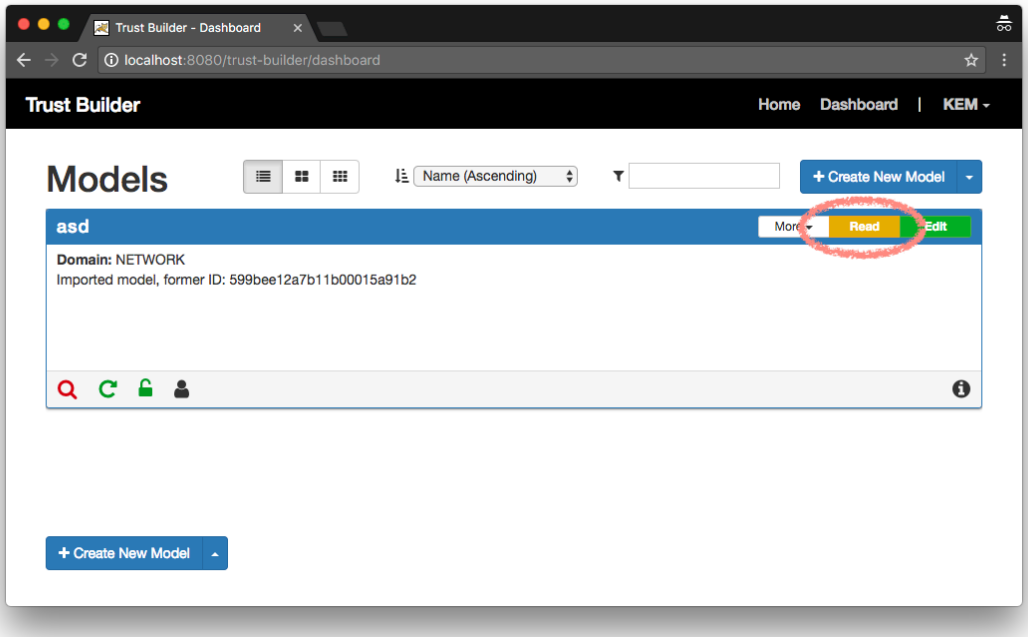


Figure 6. View of the imported sample model in the Trust Builder

Click on “Read” button and configure SSSR with the URL of the page that opens.

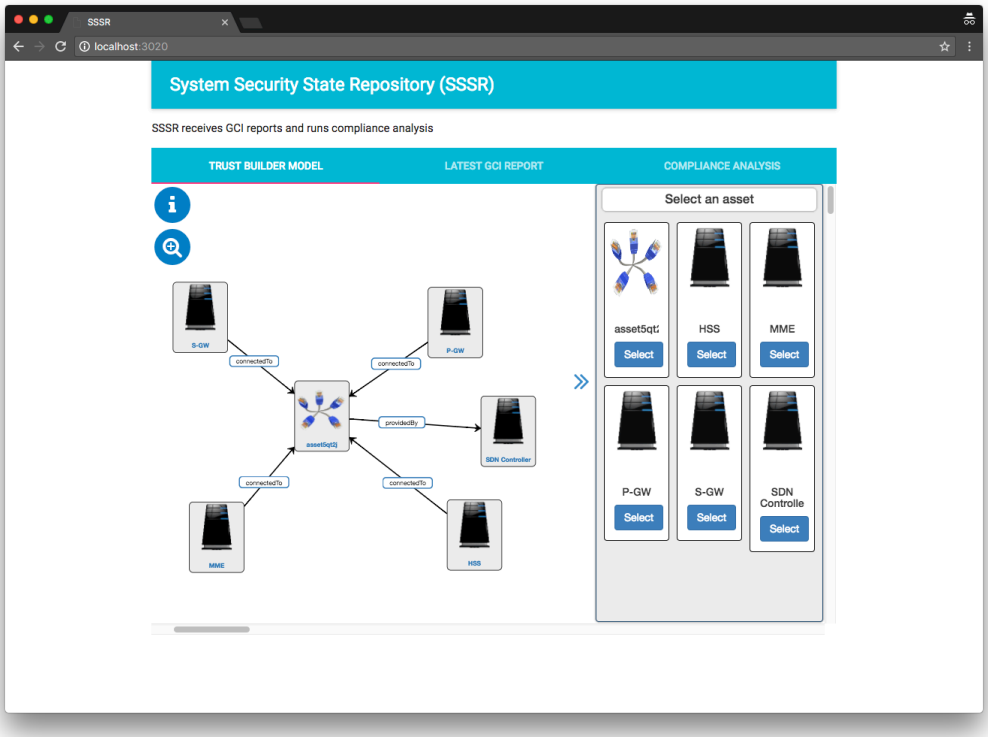


Figure 7. Sample imported model in the SSSR user interface

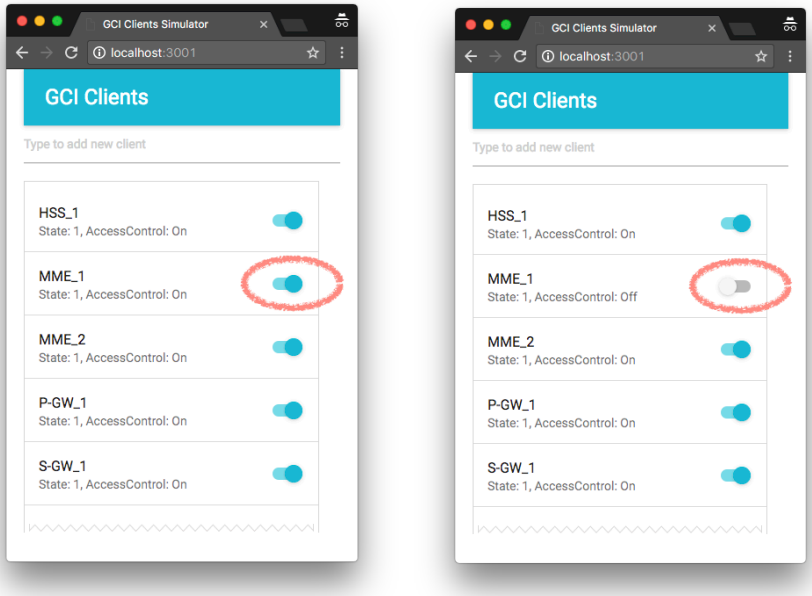
The same model should now be displayed in the SSSR main window.

3.1.2 GCI Components simulator

GCI components simulator is preloaded with the same assets as modelled in "Model_for_SSSR_validated.nq" model in the Trust Builder. Each component has two parameters:

- State of the component (set to "1")
- AccessControl status of the component (can be toggled on or off)

The AccessControl of each component can be changed in the user interface:



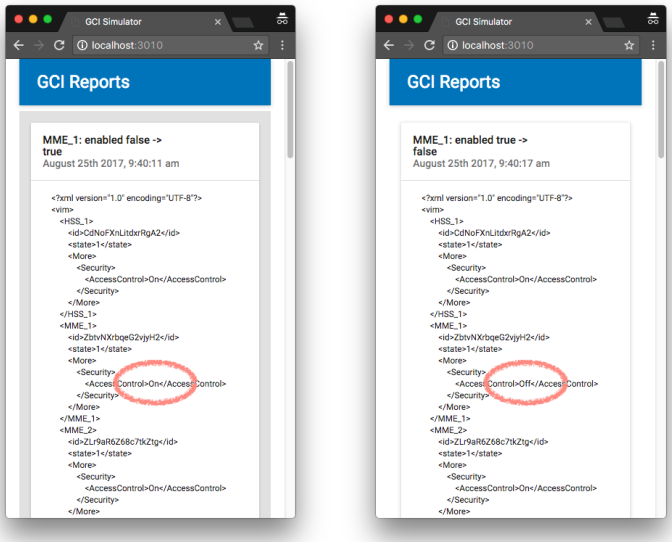
Each change is reported to the GCI simulator, which is described below.

3.1.3 GCI simulator

The Generic Collector Interface simulator has the following functionality:

1. Receive data from simulated GCI components
2. Create GCI system status reports (See D3.6 for details [3])

Changes in GCI components' state trigger new GCI report generation:



The reports are received by the SSSR as described in the following section.

3.1.4 SSSR

The System Security State Repository user interface has the following tabs:

- Read-only view of the Trust Builder model
- Latest GCI report
- Test output of the compliance analysis

3.1.4.1 SSSR compliance analysis

A system component is compliant when it's AccessControl is "On" (as per model validation results in Trust Builder). Compliance reports are generated for each GCI report received. The SSSR "Compliance analysis" tab displays five most recent compliance reports:

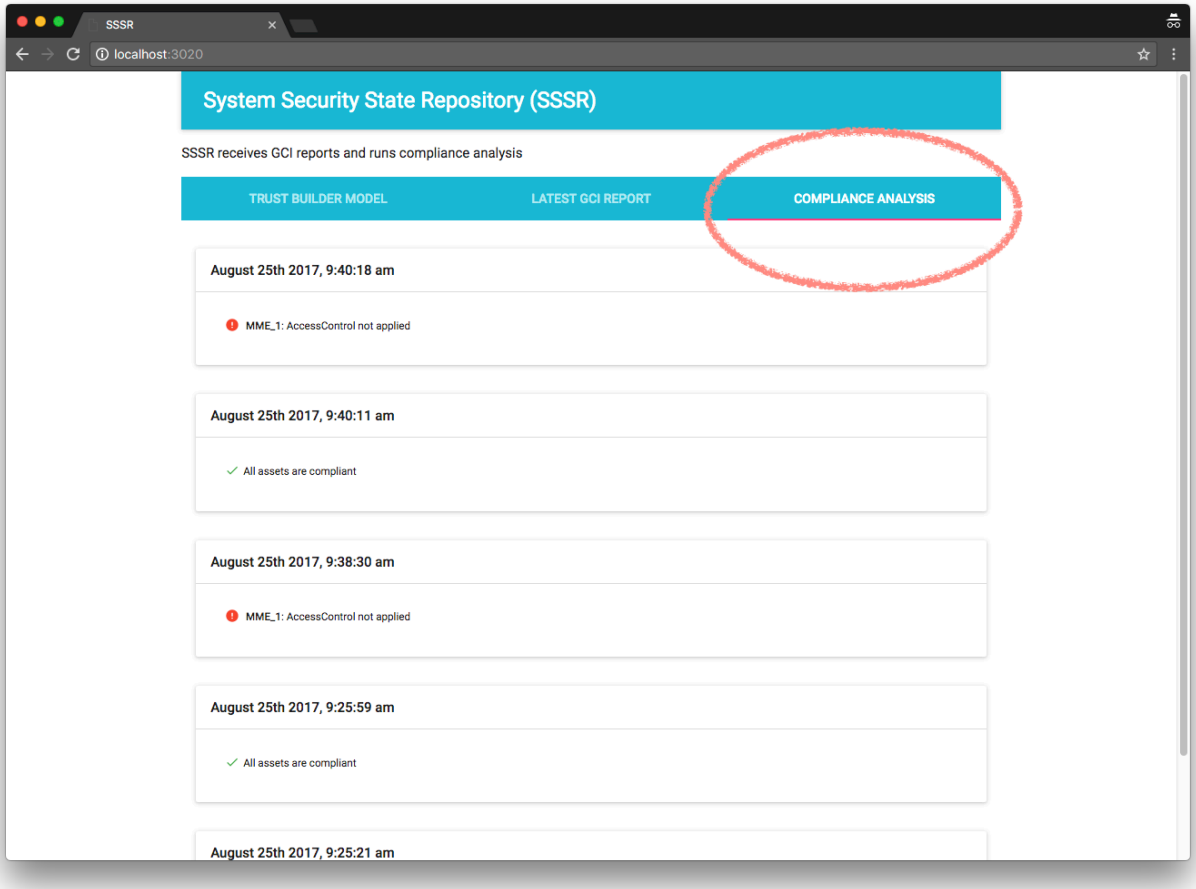


Figure 8. SSSR user interface: compliance analysis tab with the latest five compliance reports

3.1.4.2 SSSR latest GCI report

The second tab in the SSSR UI shows the latest report received from the GCI:

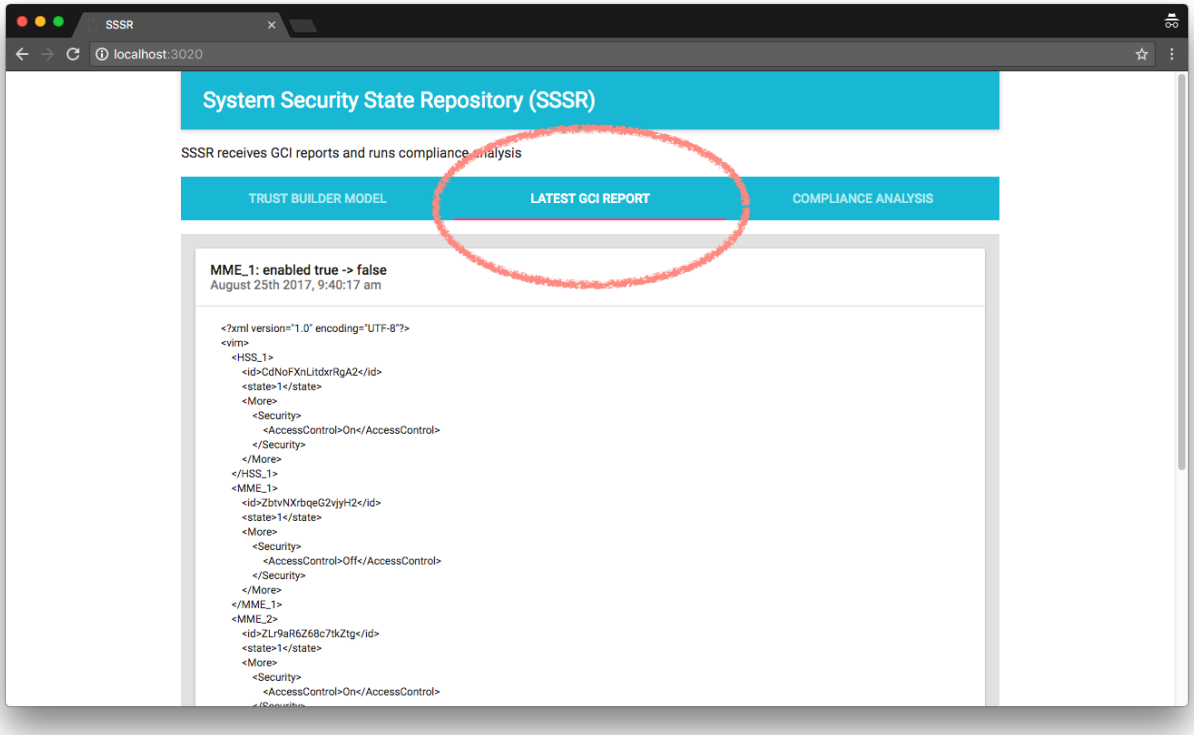


Figure 9. SSSR user interface: latest GCI report tab displays the latest received report from the GCI simulator

The data from that report is used to generate the current security state of the system and to create compliance reports.

3.1.4.3 SSSR read only view of the system model

The first tab of the SSSR UI displays the current security state of the overall system using read-only view of the Trust Builder's model:

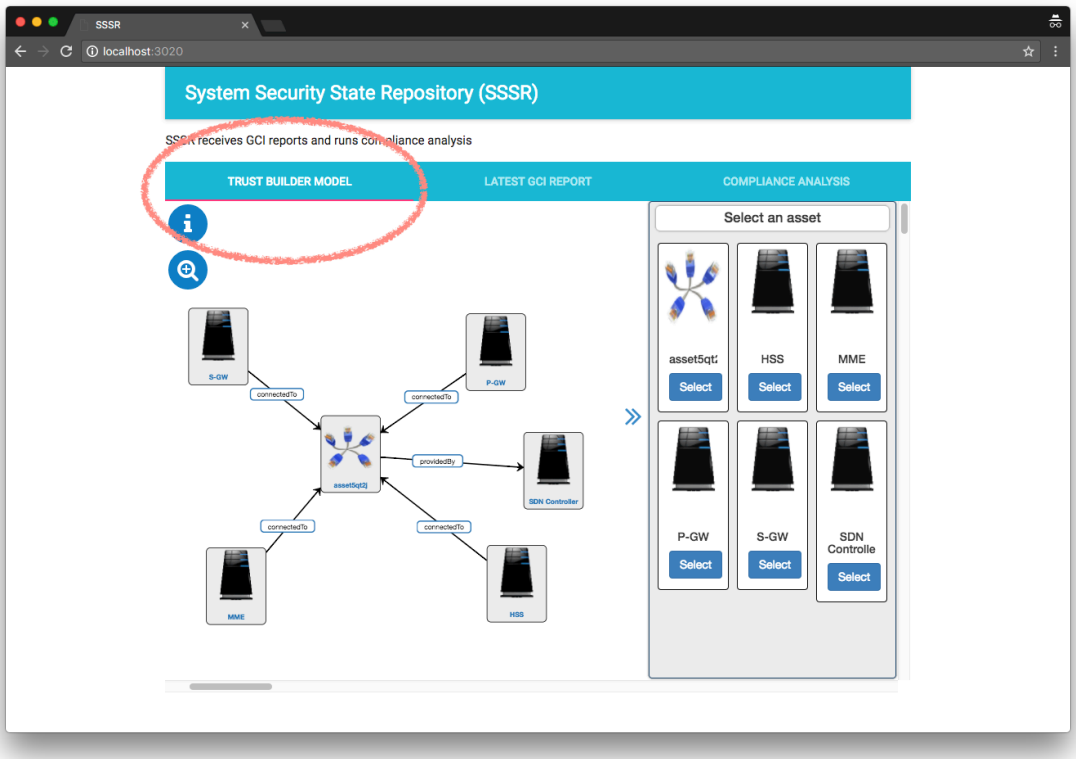


Figure 10. SSSR user interface: view of the read only model of the system from the Trust Builder

When non-compliant assets are identified, the corresponding asset type is highlighted red in that view:

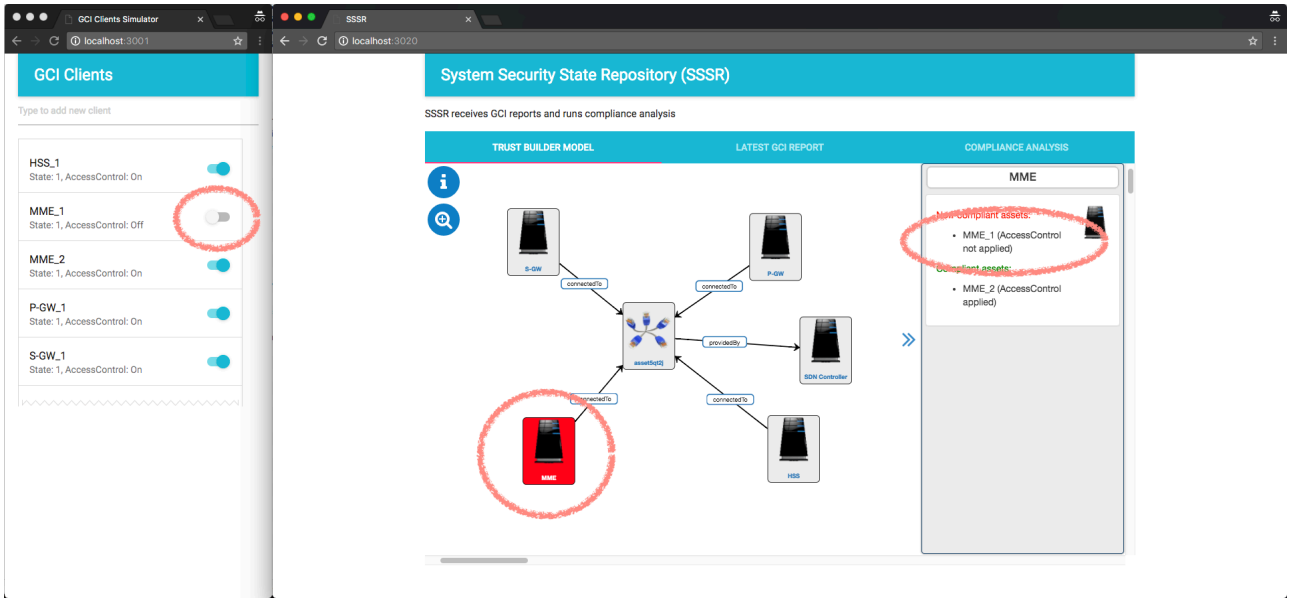


Figure 11. Display of the non-compliant assets in the model view. Asset type tile with non-compliant asset is highlighted. Panel on the right displays compliance details about assets of that type

Clicking on the highlighted asset type displays information about non-compliant assets in the right-hand panel of the model view, in the example below asset “MME_1” is non-compliant as it is expected to have AccessControl “On”. Display of multiple non-compliant assets is supported:

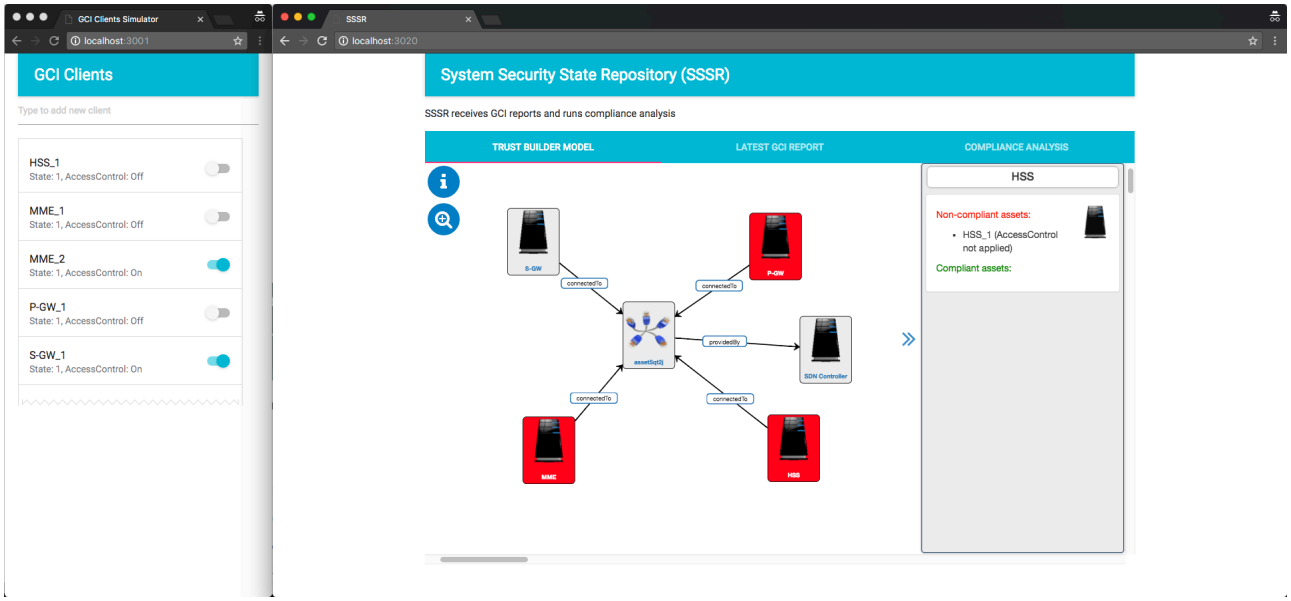


Figure 12. Sample mobile network with multiple non-compliant assets visualized with corresponding asset types highlighted red in the Trust Builder's system model (MME asset type is selected)

3.2 Programmer Guide

N/A.

4 Unit Tests

Any change in GCI client's state should propagate immediately through the whole system. Tests below can help identify potential communication issues between parts of the system.

4.1 Trust Builder with SSSR model

You should be able to login into Trust builder at the following URL: http://localhost:8080/trust-builder/dashboard

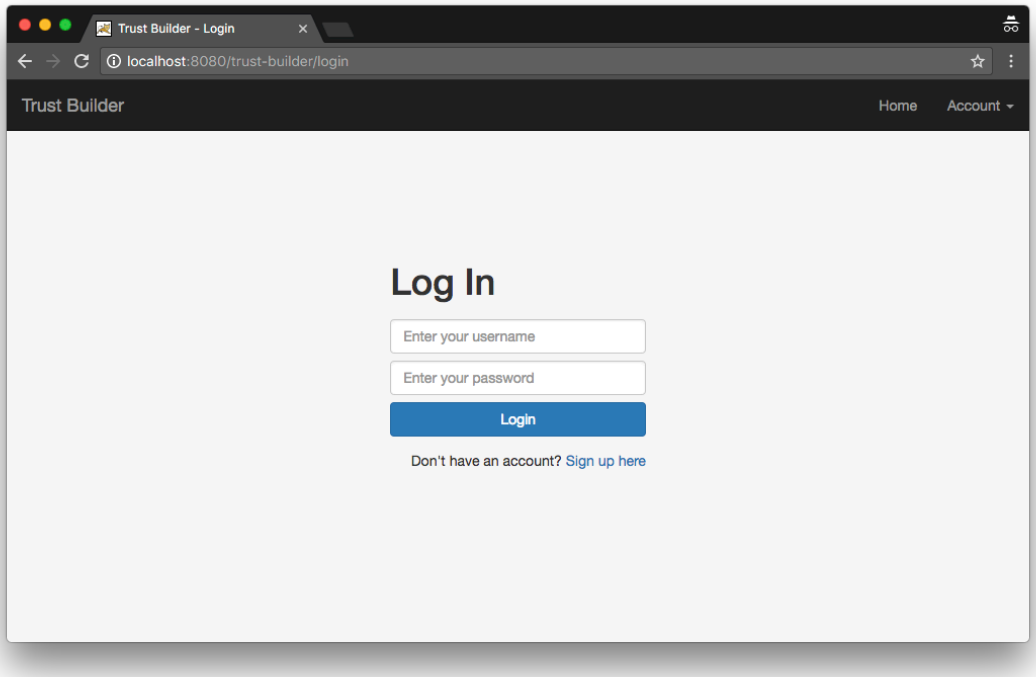


Figure 13. Trust Builder dashboard login page

Log in using trustbuilder:5fd4661f32ef9d2be4a3f794dff64cdd and you should see an empty Trust Builder dashboard:

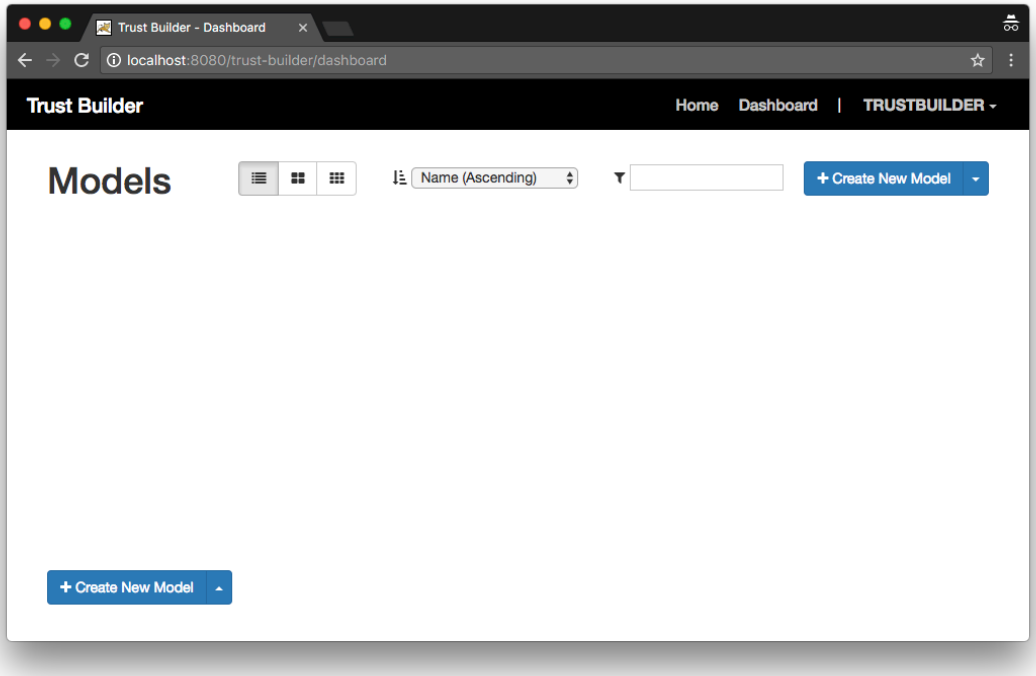


Figure 14. Empty dashboard page of the Trust Builder after initial setup

Expand **Create New Model** dropdown and select **Import Existing Model**:

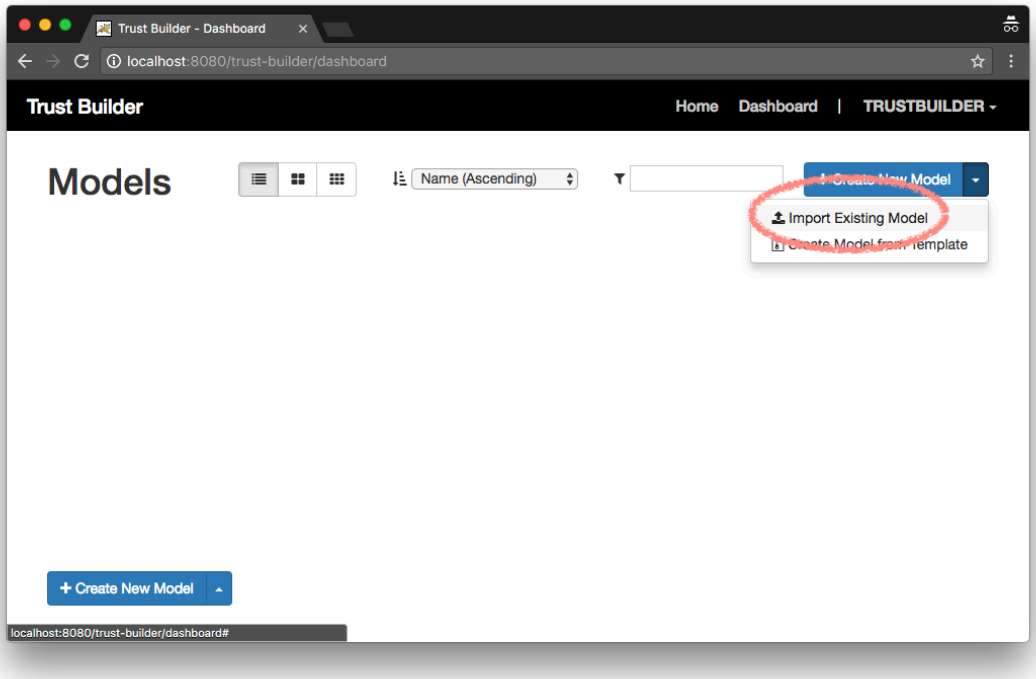


Figure 15. Link to importing functionality of existing models from a file into the Trust Builder dashboard

Open **Model_for_SSSR_validated.nq** provided and click **Import**:

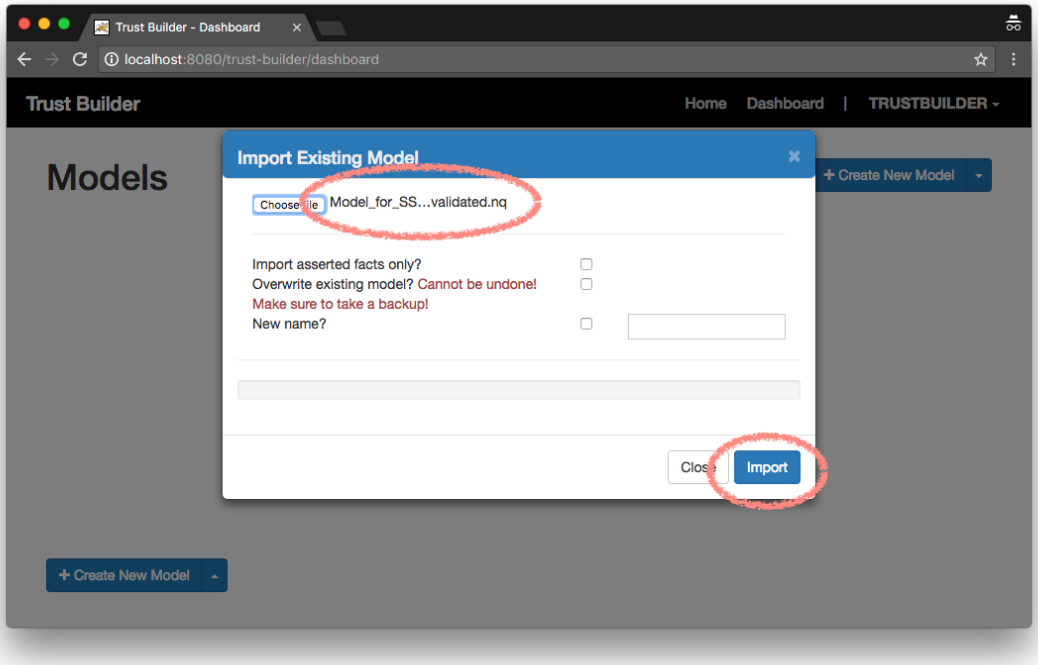


Figure 16. Importing selected model from a file into the Trust Builder

Click **Close** when the import is complete:

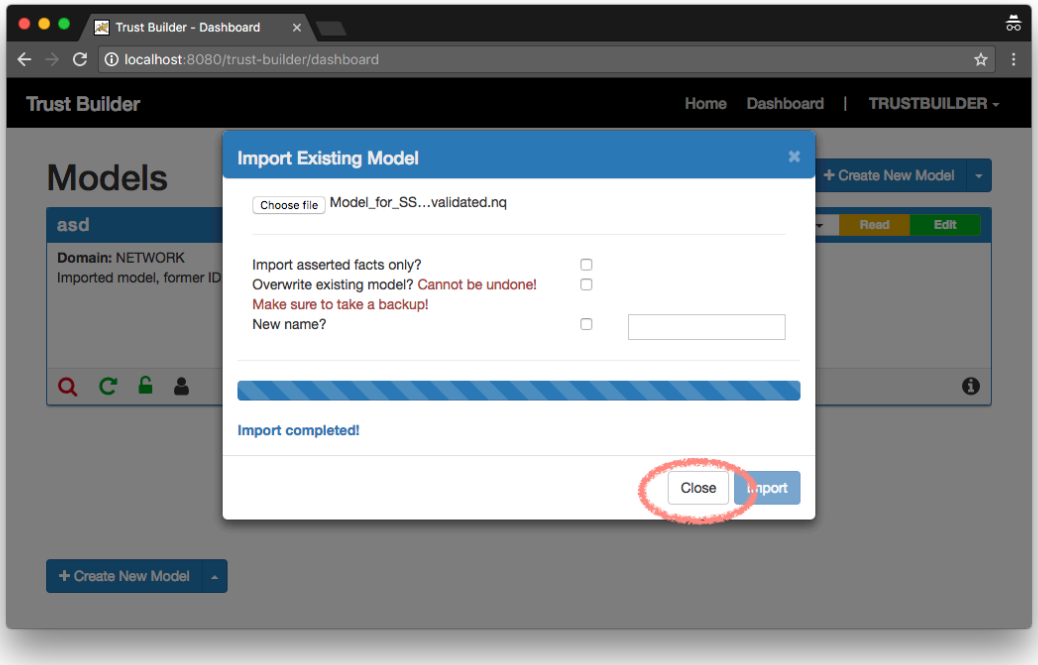


Figure 17. View of the import dialog on successful import of a model

You should now have one model loaded in the dashboard. Click **Edit** in the model tile:

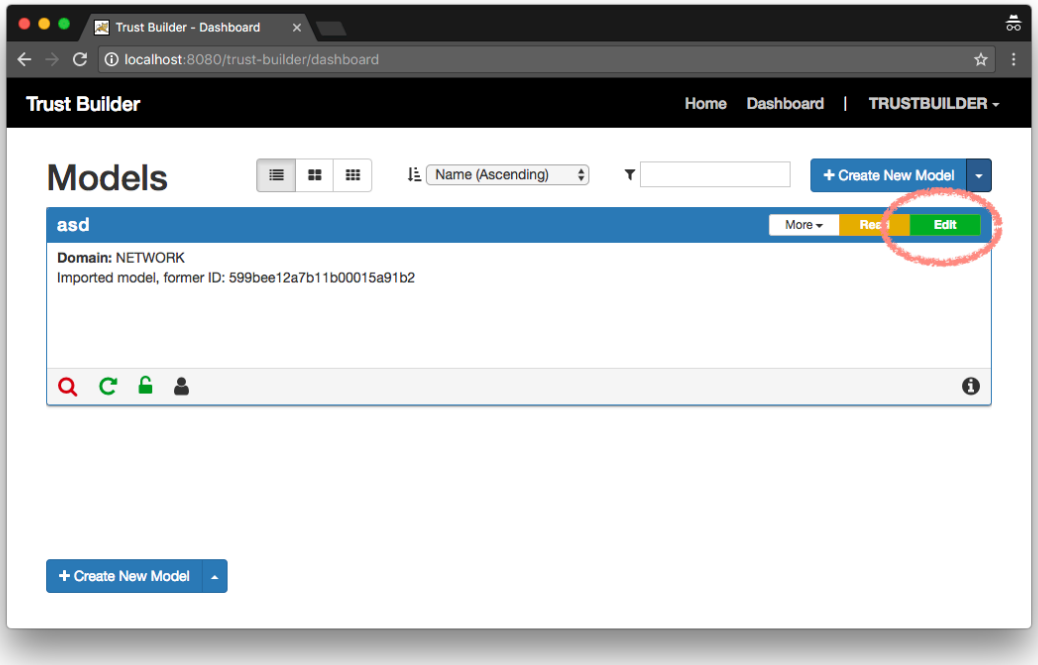


Figure 18. Trust Builder dashboard with one imported model. Link to model editing page is highlighted

The model should load:

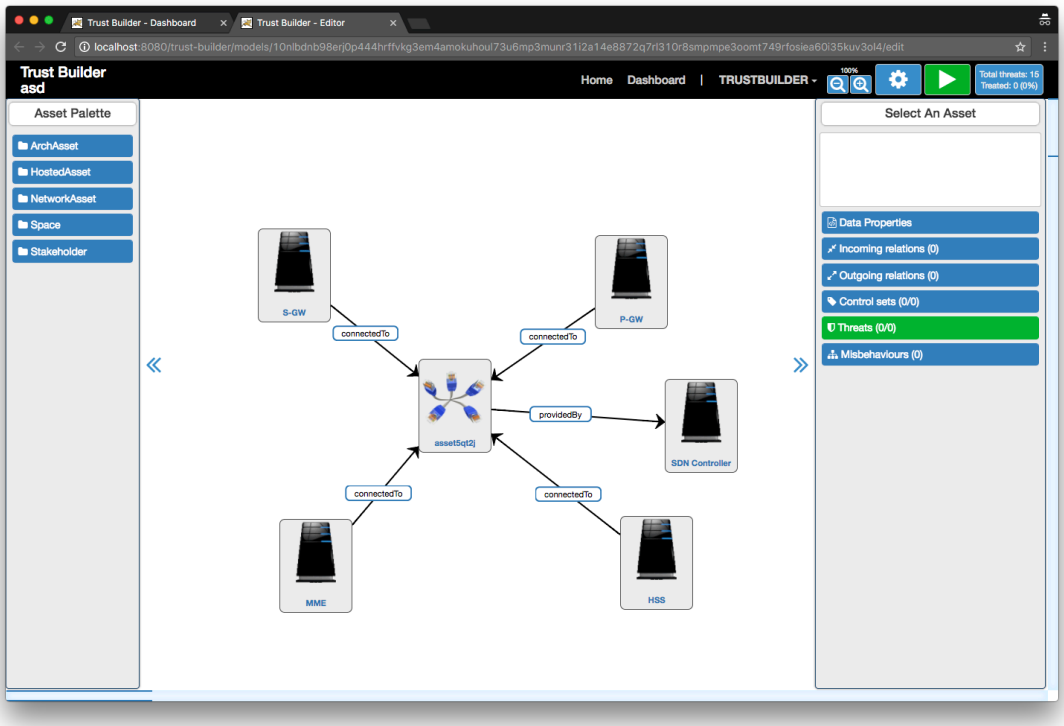


Figure 19. Sample SSSR model on Trust Builder editing page

Click on **MME** and expand **Control sets** in the right-hand panel:

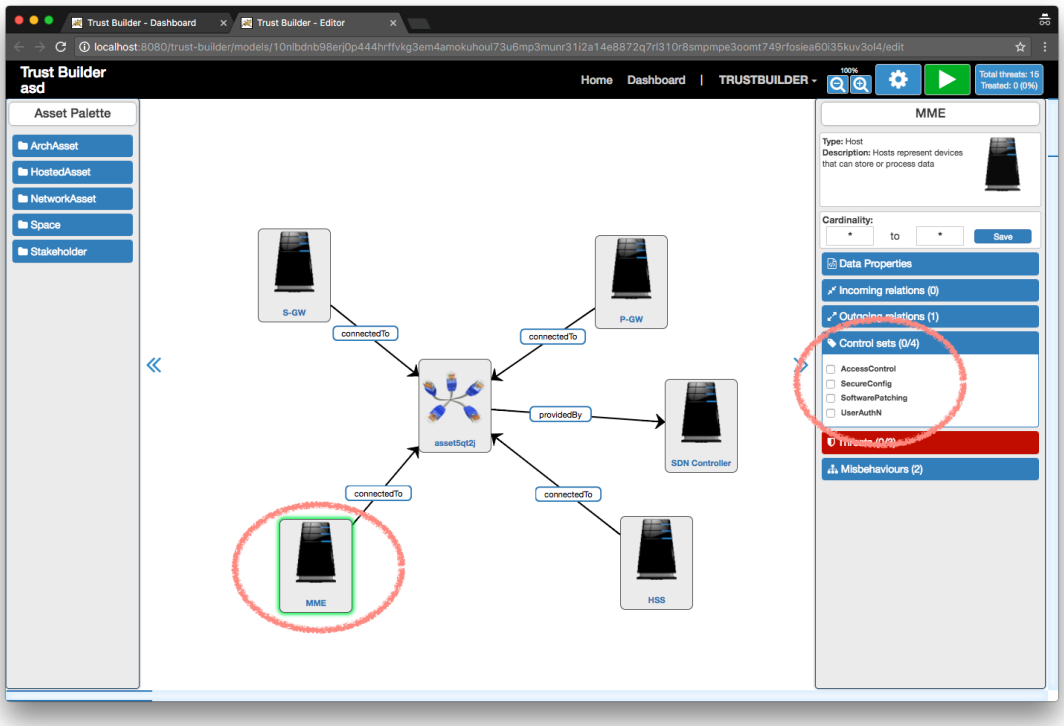


Figure 20. Highlighted model entity from a sample SSSR model in Trust Builder editing page

This concludes the test.

4.2 SSSR – Trust Builder

SSSR main page (http://localhost:3020) should load the exact same model in the first tab as the one loaded in the Trust Builder. If that doesn't happen, double-check sssr/settings.json, rebuild the image and restart SSSR.

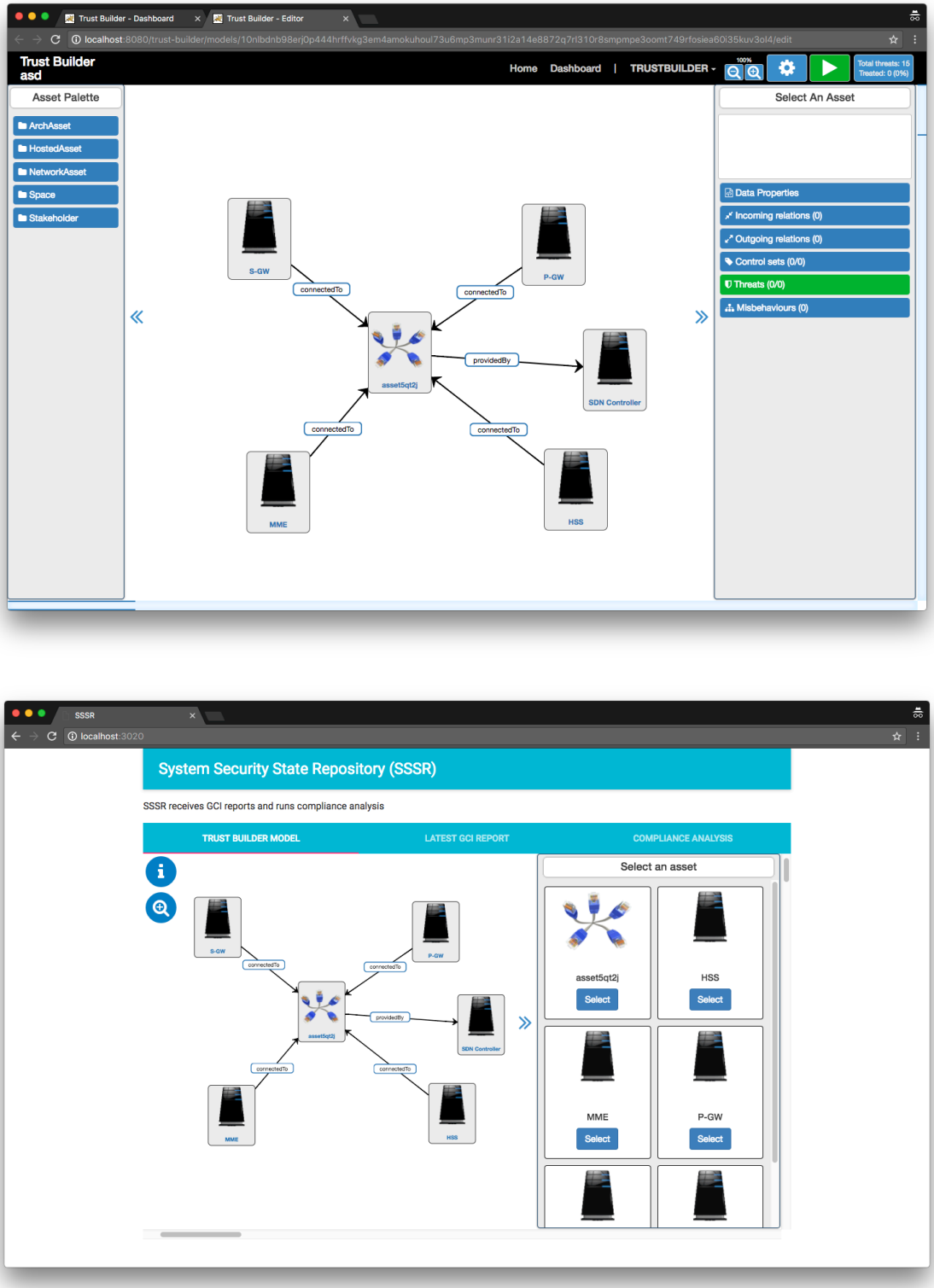


Figure 21. The same model viewer in the Trust Builder (top) and SSSR (bottom)

4.3 SSSR – Components

Any change in GCI Clients simulator should result in immediate update of both:

- GCI Simulator (new GCI report should appear)
- SSSR “Compliance Analysis” tab on the main page

5 Acknowledgements

The work described in this deliverable was sponsored by 5G-ENSURE Project (Grant Agreement number: 671562 — 5G-ENSURE — H2020-ICT-2014/H2020-ICT-2014-2).

6 Abbreviations

SSSR	System Security State Repository
TB	Trust Builder
GCI	Generic Collector Interface

7 References

[1] DDP Specification: https://github.com/meteor/meteor/blob/master/packages/ddp/DDP.md

[2] 5G-ENSURE_D3.8 5G-PPP Security Enablers Documentation (v2.0), Trust Builder, section 3.1.2.3

[3] 5G-ENSURE_D3.6 5G-PPP security enablers open specifications (v2.0), section 5.2