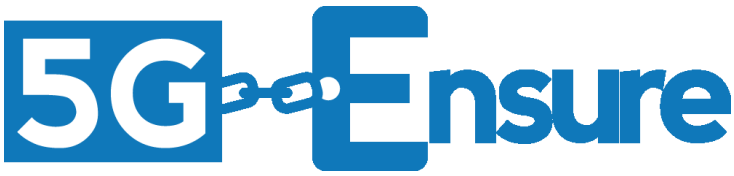


Deliverable D3.4
5G-PPP Security Enablers Documentation (v1.0)
Trust Builder

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	30.09.2016	
Dissemination Level:	Public	
Lead beneficiary	NEC	Felix Klaedtke, felix.klaedtke@neclab.eu
Authors	IT Innovation: Stephen Phillips, Juri Papay, Mike Surridge	

Document Version	Date	Change(s)	Author(s)
0.1	28.06.2016	Created template	Felix Klaedtke
0.2	25.08.2016	Document outline	Stephen Phillips, Juri Papay
0.3	20.09.2016	Adding screenshots	Juri Papay
0.4	23.09.2016	Revision	Mike Surridge
0.9	28.09.2016	Final screenshots	Juri Papay
1.0	28.09.2016	QA	Stephen C. Phillips

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, standardisation and vision for a secure, resilient and viable 5G network. The project covers research & innovation - from technical solutions (5G security architecture and testbed with 5G security enablers) to market validation and stakeholders engagement - spanning various application domains.

The presented deliverable describes the Trust Builder enabler developed at IT Innovation. This deliverable forms a part of D3.4 that describes the WP3 Security Enablers for 5G. This deliverable is a draft version some sections are not yet complete and will be updated soon.

Disclaimer

The information in this document is provided 'as is', and no guarantee or warranty is given that the information is fit for any particular purpose.

The EC flag in this deliverable is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-ENSURE receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

Copyright notice

Contents

1 Introduction

The notion of trust in 5G mobile networks if one of the hot topics of research. In this respect we may consider trust between the end users and network operators or trust between the network operators only. Over the years numerous trust models have been suggested, however they must be updated so that these models reflect the requirements of 5G. One of the motivations of our work was to deliver a tool that enables to construct trust networks, to reason about these networks and perform "*what if studies*". The presented Trust Builder uses ontology-based reasoning for analyzing dynamic complex systems. This tool enables the user to identify security threats and take mitigation actions.

Trust Builder is a graphical tool that enables to construct trust networks for representing 5G networks, generate potential threats and validate the model. The output of validation is a modified trust network enriched by features that were not captured by the initial design.

For describing the Trust Builder we use a set of terms for explaining various features if the tool:

- *Core Model* – the core ontology, defining common vocabulary and relationships used in all higher level models.
- *Generic Model* – an ontology defining the typology of Assets, Threats and Controls (security measures) for a given domain (e.g. 5G networks).
- *Design-Time System Model* – an abstract model of a particular system, described in terms of relationships between system specific Asset classes. The design time model can be enriched by specifying which Security Controls. These controls allow to protect the assets, and generate a set of system-specific Threat Classes for describing potential threats to the system.
- *Runtime System Model* – a model using instances of Assets, Threats and Controls for describing what is known about the current state of the system.
- *Domain Modeler* - a software tool for defining a generic domain model.
- *System Modeler* - a software tool for defining a design-time system model in terms of assets and other elements from a suitable generic model.

The presented document is structured as follows:

- Section 2 is the Installation and Administration Guide that describes the system requirements, configuration, installation and troubleshooting.
- Section 3 is the User and the Programmer Guide for Trust Builder. The User Guide represents the bulk of this document and provides a detailed account of the system's functionality.
- Section 4 describes a case study that illustrates a typical usage of Trust Builder.

The Trust Builder is released as a confidential project output to partners under the terms of the 5G-ENSURE consortium agreement.

2 Installation and Administration Guide

This section describes the system requirements, configuration, installation and administration of "*Trust Builder*" software. Trust Builder is released as a confidential project output to partners under the terms of the 5G-ENSURE consortium agreement.

2.1 System Requirements

For running the software requires a Java 8 installation, Tomcat server and a MongoDB server. The dependencies are Java libraries that are managed by Gradle tool. The software dependencies of Trust Builder are summarised in Table 1.

Group	Artefact	Version
org.springframework.boot	spring-boot-starter-data-mongodb	1.4.0
	spring-boot-starter-data-rest	1.4.0
	spring-boot-starter-web	1.4.0
	spring-boot-starter-security	1.4.0
	spring-boot-starter-thymeleaf	1.4.0
	spring-boot-starter-tomcat	1.4.0
org.springframework.mobile	spring-mobile-device	1.1.5
com.googlecode.json-simple	json-simple	1.1
io.jsonwebtoken	jjwt	0.2

Table 1 - System requirements

2.2 Enabler Configuration

The default configuration provides a deployment onto a single machine. The pre-condition is that both Tomcat server and MongoDB are running with the default settings on the same machine. In case the MongoDB is deployed on a remote host then we need to update the *application.properties* file so that it refers to MongoDB's address. The configurable settings are as follows:

```
#tomcat server settings
server.port=8080
server.contextPath=/system-modeller

#MongoDB connection URL
spring.data.mongodb.uri=mongodb://localhost:27017/system-modeller

#MongoDB Users collection names
mongo.user.collection=Users

#Logging levels
logging.level.org.springframework.web=DEBUG

# default users path
# note: users file contains test users (all of them will be stored in the database in the future)
load.users.from.file=true
user.path=src/main/resources/users.json

#Json serialization settings
spring.jackson.serialization.INDENT_OUTPUT=true

#Java Web Token authorization settings
jwt.header=Authorization
```

```
jwt.secret=secret
jwt.expiration=604800
jwt.route.authentication.path=auth
jwt.route.authentication.refresh=refresh

# Default admin user
create.default.admin=true
admin.email=admin
admin.password=admin@server.org

# Default guest user
create.default.guest=true
guest.email=guest@server.org
guest.password=guest

# cookies settings
trust.builder.cookies=JSESSIONID,Authorization
```

2.3 Enabler Installation

The installation procedure consists in copying the war file into the Tomcat *webapps* root folder and restarting the Tomcat server. The Tomcat manager app can also be used for deploying the war file and check if the installation was successful. The software was installed on Ubuntu 14.04 Operating System, the installation steps is represented by a sequence of “apt-get” commands.

```
> apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927
> echo "deb http://repo.mongodb.org/apt/ubuntu "$(lsb_release -sc)"/mongodb-org/3.2
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb.list
> apt-get update
> apt-get install mongodb-org
> service mongod start
> add-apt-repository ppa:webupd8team/java
> apt-get update
> apt-get install oracle-java8-installer
> apt-get install tomcat7
> service tomcat7 start
> mv system-modeller.war /var/lib/tomcat7/webapps
> service tomcat7 restart
```

2.4 Troubleshooting

The installation procedure declares explicitly the dependencies of the software as standard packages and libraries. The application is composed of a standard MongoDB installation and a web application running on a Tomcat server. Any errors are captured in the standard log files (e.g. /var/log/tomcat7/catalina.out) after the application war file has been deployed.

3 User and Programmer Guide

3.1 User Guide

In this section we describe the functionality of the Trust Builder. For a better understanding we provide definitions of the main concepts used in this document. An Asset is an element of the network that can be the following: Stakeholder, Logical asset or Physical asset. Stakeholder is a person or organisation, i.e. an entity that can carry out actions. Logical asset is a process, usually represented by a software. A physical asset is an element of the infrastructure or environment. For example it can be a Host, a Network or an Interface (a point of control). A Misbehaviour represents different ways in which assets may be compromised as a consequence of an active threat. An Involved asset is an asset whose presence is necessary for a threat to occur or to be managed (where a control can be located). A Control Strategy is a set of controls located at different assets that block or mitigate a threat.

The following sections provide details of the system's functionality covering:

- User management
- Model management
- Model editing
    a. Stage 1: defining assets and relationships which provide the initial model of a network
    b. Stage 2: validation and auto-generation of threats
    c. Stage 3: defining threat management strategy (selecting controls for assets or control strategies for threats)
- Model outputs

Trust modelling in essence is a three stage process, with the stages used repeatedly in an iterative fashion. In the first stage the user constructs the trust model by putting assets into the modelling panel and establishing links between the assets. An "asserted asset" is one defined by the user in this way during Stage 1. There are Physical, Logical, Stakeholder and Connection assets that can be used for constructing the Trust model. An "inferred asset" is one generated automatically in the model validation at Stage 2, whose presence can be inferred from the asserted assets and their relationships.

The validation process in stage 2 automatically generates inferred assets and also threats and possible security controls to counter act threats. The validation process also determines whether the information provided about assets and their relationships is consistent and complete. If this is not the case, the validation still runs but the model is marked as 'invalid'. In this case the uses should go back to Stage 1 and updates the model so that it contains sufficient information for a successful validation. Once the validation step is successful, the model is 'valid', but it becomes invalid again if any Stage 1 changes are made to the assets or their relationships.

In Stage 3 the user addresses threats by selecting or modifying the set of security controls that should protect the assets in the system. The aim is to eliminate threats by addressing them, or at least reduce the list of threats to an acceptable level.

3.1.1 User Management

3.1.1.1 Main page

On the main page of Trust Builder (Figure 1) there are several links in the upper right hand corner of this page, these are: *Home*, *View Models* and a dropdown menu under the *Person* icon. The *Home* link takes the

user to the main page of "*Trust Builder*". The *View Models* presents a list of models previously defined by the user. The dropdown menu under the person icon provides the following functionality:

a) Sign in
b) Register
c) Forgot Password
d) Manage Account
e) Sign out

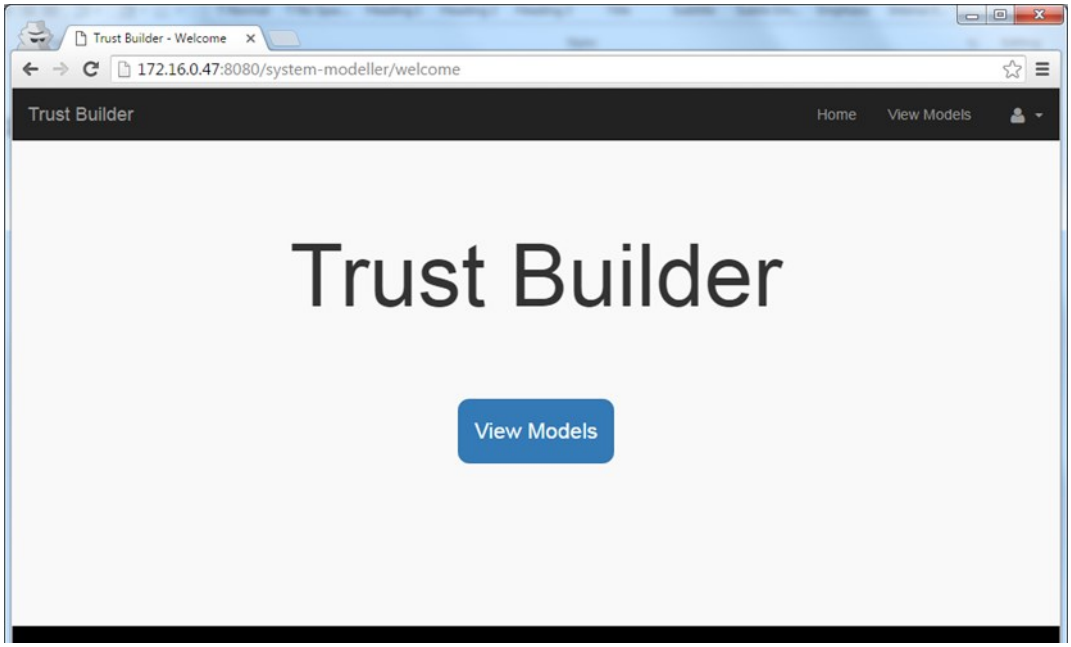


Figure 1- Trust Builder main page

3.1.1.2 User login

The login page of Trust Builder is activated by clicking the *Sign In* link in the dropdown menu under the person icon. If there is no login and password the user needs to register for the service by clicking on a link *Sign up here* (see Figure 2).

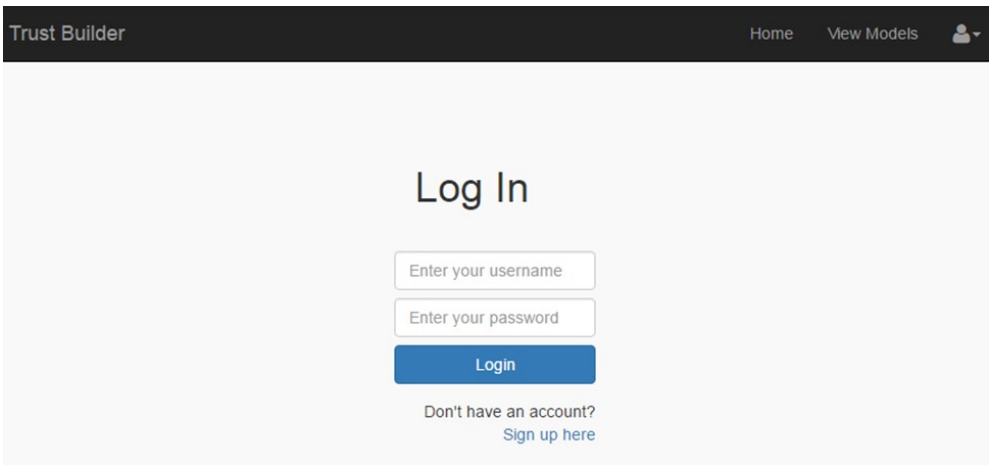


Figure 2 – User login page

3.1.1.3 User registration

The user can register by providing email address and password (see Figure 3). On registration the user's account is still inactive. The system administrator needs to activate the account before the user is able to login. The user is notified via email about activating the account.

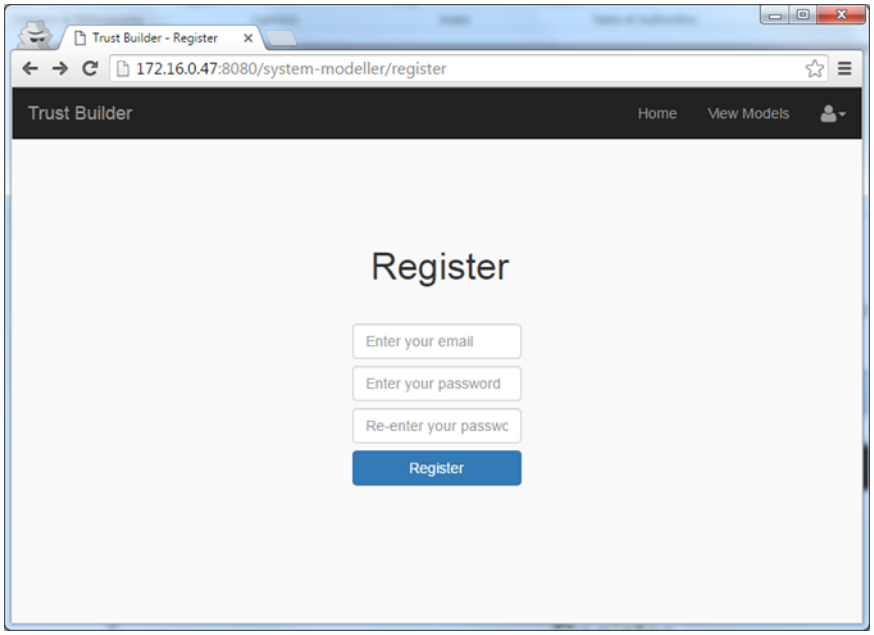


Figure 3- User registration

3.1.1.4 Password management

The user can reset the password by clicking on the "Forgot Password" link in the dropdown menu (see Figure 4).

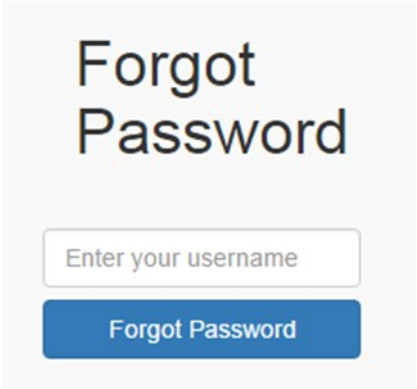


Figure 4 – Resetting the password

After typing in the user id and clinking on the "Forgot Password" button the user is presented with the "Reset Password" page (see Figure 5).

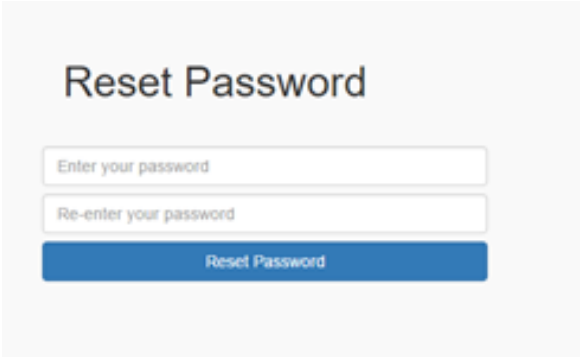


Figure 5 - Password reset

3.1.1.5 Logout

Logout is activated by clicking on the *Sign Out* link in the dropdown menu under the person icon.

3.1.2 Model management

The term Model management incorporates several functions such as:

a) Listing models created by the user or shared by others
b) Creating models
c) Importing/Exporting Models
d) Managing access to models
e) Deleting models
f) Checking in/out models

In the first release of the Trust Builder only a subset of the above functions has been implemented.

3.1.2.1 List models

By clicking on the "*View Models*" button on the main page the user can list the models that the user either owns or has read/write access to (see Figure 6).

Figure 6- List models

In Figure 6 we can see one model called "caseStudy1". At the bottom left corner of the model window there are five icons. These are indicators reflecting the status of the model. The colour of these indicators reflects whether the corresponding action has taken place or not. For example if the "magnifying glass" indicator is read it means that for the given model no report was generated. These indicators are described in Table 2.

Icon	Description

	indicates if a report was generated (green colour) on not (red colour)
	indicates if the model was validated (green colour) on not (red colour)
	indicates if the model was shared (not yet implemented)
	indicates if the model is not locked (green) or locked (red)
	last modification of the model
	when the model was created

Table 2 - Model indicators

The dropdown menu in the top right corner of the model window offers several functionalities, these are: Edit, Share, Rename, Copy and Lock (see Figure 7).

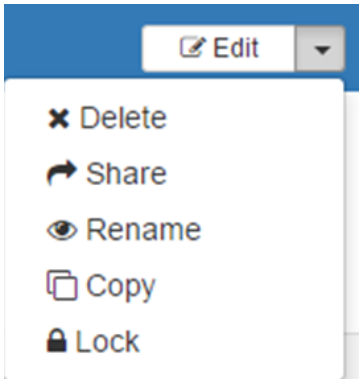


Figure 7- Model dropdown menu

3.1.2.2 Create model

By clicking on the "*Create New Model*" the user can create an empty model (see Figure 8). The drop-down selection allows the user to choose which generic model (version) to use to construct the model. The user automatically becomes an owner of the newly created model with read/write access and with the ability to of granting/revoking access rights for other users. The new model is added to the model list and it can be edited.

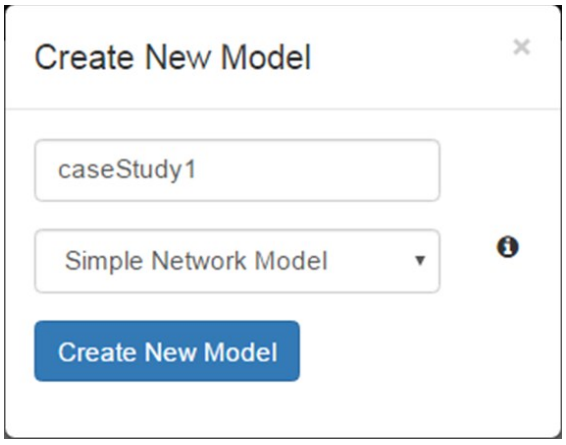


Figure 8 - Creating a new model

The "*Import Existing Model*" and "*Create Model from Template*" options are not implemented for the first release.

3.1.2.3 Delete model

The delete action removes the model from the list along with the contained resources (e.g. assets, relationships) and all information about the access rights to the model for other users. The precondition for the delete operation is that the model should not be checked out by another user.

3.1.3 Asset definition

Clicking the *Edit* button opens the editing panel that consists of three parts. On the left side there is the "*Select Assets*" panel. In the middle there is the Model Construction Canvas. The right side panel contains various categories related to an item selected on the canvas, such as *Incoming/Outgoing Relations*, *Control Sets*, *Roles*, *Inferred Relations and Threats* (see Figure 9).

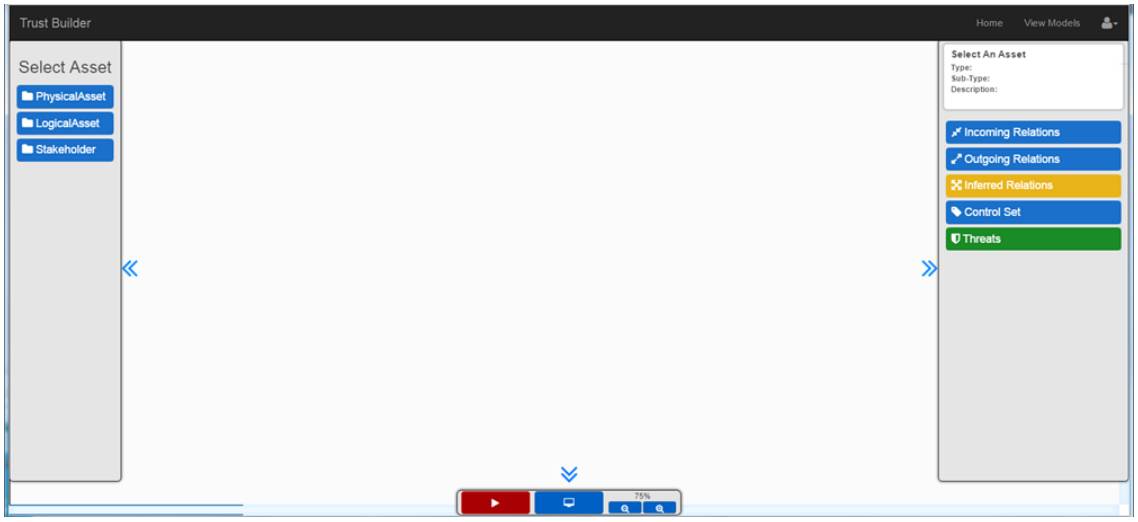


Figure 9 - Model editing

The main panel also contains four buttons, these are in Table 3.

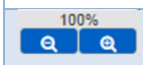
Icon	Description
	Process (validate) the model
	Upload a screenshot of the model
	Zoom controls

Table 3 – Model editing controls

3.1.3.1 Select and add asserted asset

The left panel of the model editor contains various assets, these fall into three categories (see Figure 10):

a) *Physical assets*
b) *Logical assets*
c) *Stakeholders*

The choices available depend on the generic model (version) chosen to construct the model. Here we give examples from the built-in Simple Network Model.

The Simple Network Model includes the following entities: Cellular Network, Cellular Router, Cloud Host, Host, Server, Smart Phone, Wifi Router, Wired Router, Wireless Network, Work Station.

The Logical Assets in the Simple Network Model are: DB Server, File Server, Mail Agent, Mail Store, Remote Login, Web Server.

In this model there are two entities in the Stakeholder category, these are Human and Organisation.

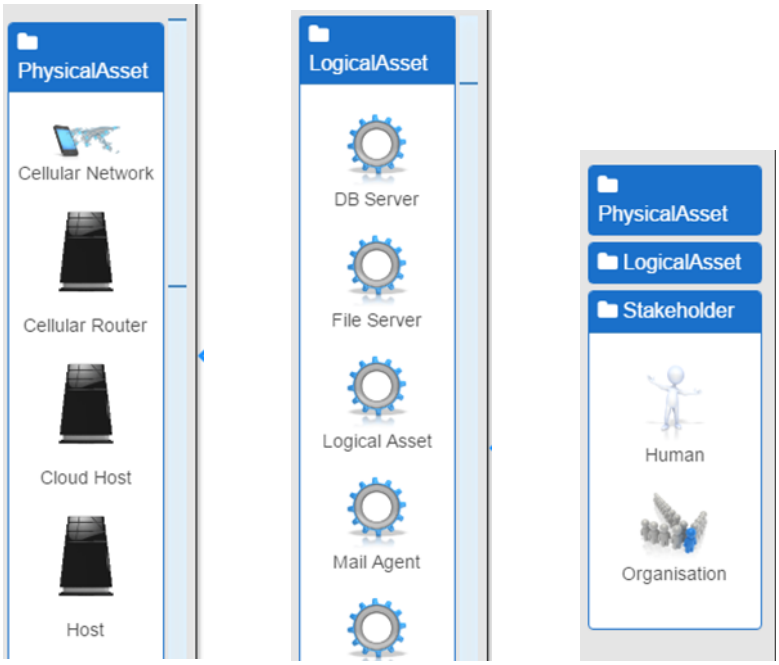


Figure 10 – Selecting assets

For inserting an asset in the model the user selects an asset by clicking on the asset's icon in left panel and dragging the asset into the model canvas (see Figure 11). By clicking on the asset we can view/edit the properties, these are:

a) name of the asset
b) incoming relations
c) outgoing relations
d) inferred relations
e) control set
f) threats (once the model has been processed)

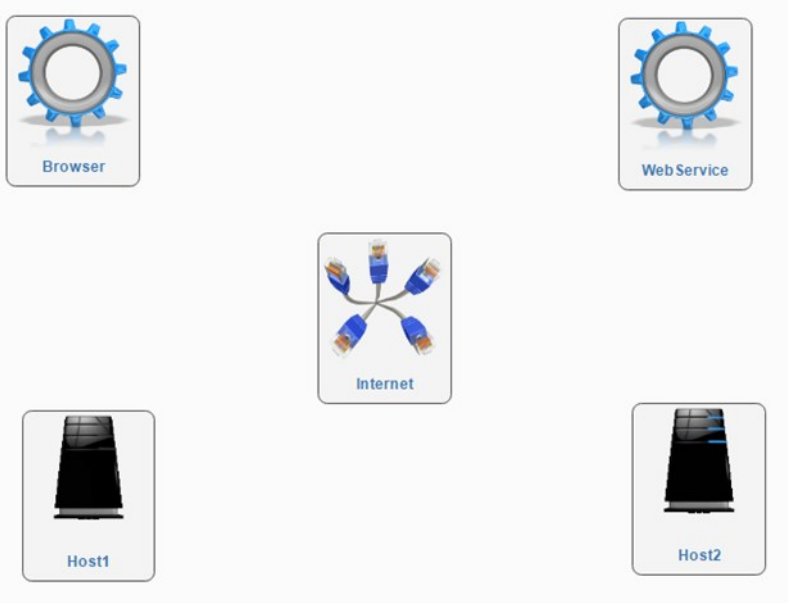


Figure 11 - Inserting assets into model canvas

3.1.3.2 Add relationship between assets

Once the assets have been put on the modelling canvas the user can connect two assets by establishing a relationship between them. By clicking on the asset a green cross appears in the left corner, this indicates that the asset can be connected to other assets (Figure 12).

Figure 12 - Connecting assets

By clicking on the green + sign of the asset (in our example Host1) appropriate assets on the canvas are indicated with a blue tick, showing that a connection can be made between the assets (see Figure 13).

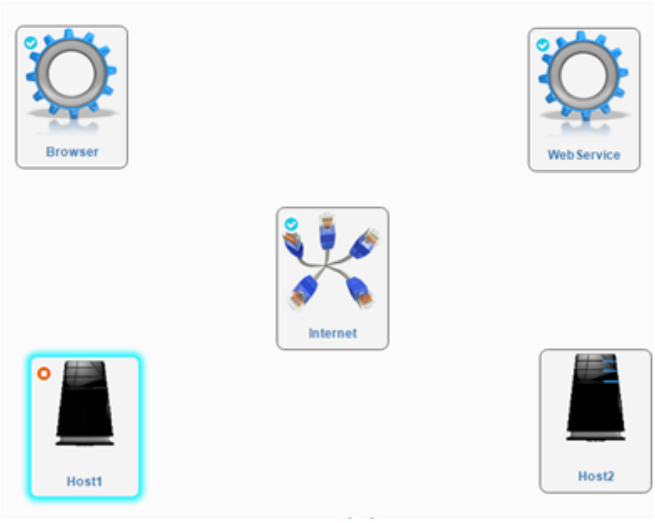


Figure 13 – Target assets for making connections

By clicking on the blue tick icons we can establish connections between assets (Figure 14).

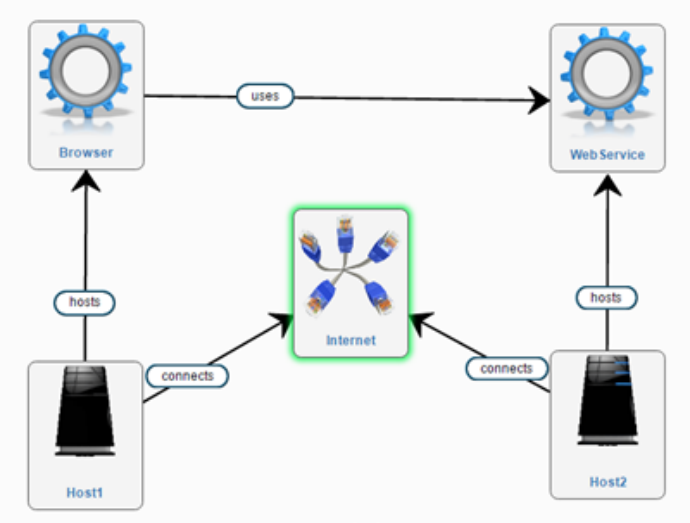


Figure 14 – Connecting assets

3.1.3.3 Remove asset

Assets can be removed by clicking on the red trash icon of the asset in the top right corner (see Figure 12). The delete operation removes all relationships between the selected asset class and other assets.

3.1.3.4 Remove relationship

By right clicking on the connection between two assets an option with a delete button comes up (see Figure 15). The delete applies only on the relationship the assets remain on the canvas.

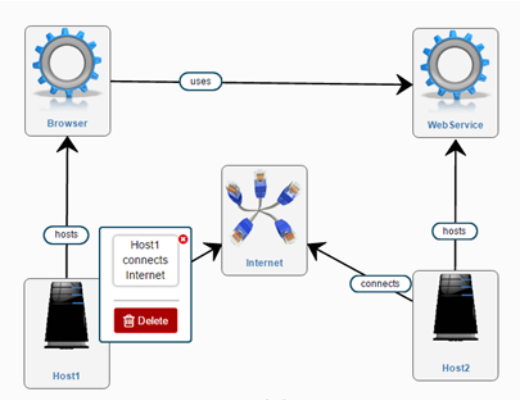


Figure 15 - Remove relationship between assets

3.1.3.5 Rename asset

The user can rename an existing asset by editing the asset's name under the corresponding icon. NB by changing the name the asset's connections will stay unaffected. All asset names must be unique.

3.1.4 Validation

Once the model is constructed it can be processed. This operation is activated by clicking on the red "play" button (see Table 3). The validation operation activates semantic reasoning that generates inferred assets that are added to the model and also a List of Threats that can be associated with the given model. This operation guarantees that the inferred assets are consistent with the asserted assets and relationships. On completion of the validation operation the updated model is presented to the user. Figure 16 illustrates the model after the validation operation.

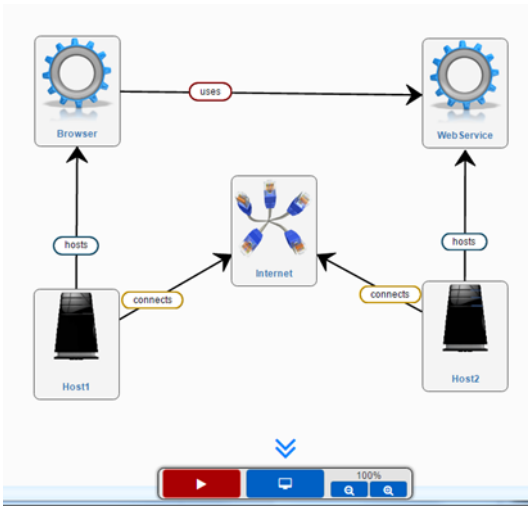


Figure 16 – Model after validation

The validation in essence generates entities that were missing from the initial model. Then the user needs to configure the new entities (e.g. inferred assets and inferred relationships). In our case there are some issues with the "*uses*" connection between the *Browser* and *Web Service*. The "play" button being red at the bottom in Figure 16 indicates that the model is not yet correct and needs to be updated and validated again. When the validation succeeds, the play button will turn green.

3.1.5 Threat management

3.1.5.1 Selecting a threat

The user can view the threats associated with the given asset. First the user needs to select an asset then click on the "*Threats*" button on the right panel (see Figure 17). This panel also provides options for editing the incoming/outgoing connections, control strategies to address a specific threat and control sets that make up the control strategy. Clicking the Edit button brings up the *Threat Editor* that allows to configure various parameters for the give threat.

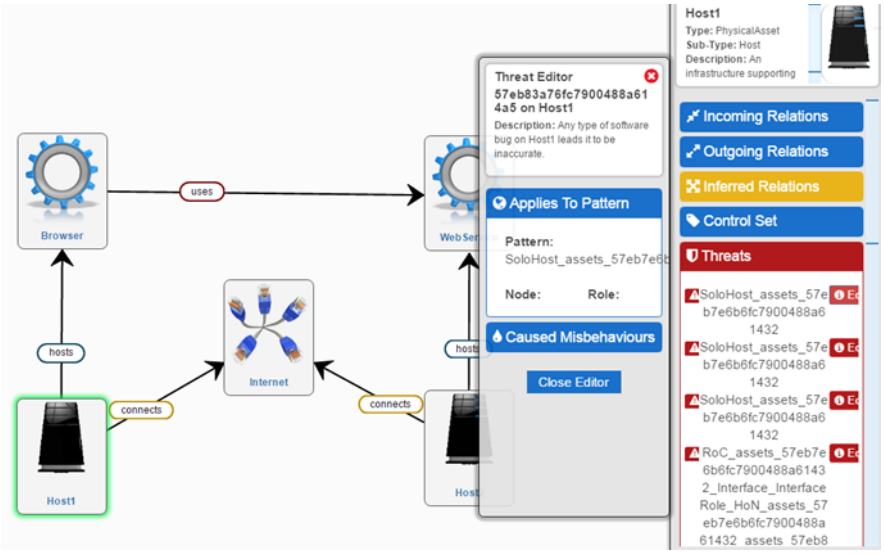


Figure 17 – List of threats associated with Host1

3.1.5.2 Set/unset control set

The user can select a control set within the asset properties panel marking it for implementation or removing the marker if already present. The control sets that are marked determine which control strategies will be in place, and hence the status of threats. The example on Figure 18 illustrates the Control Set for Host1.

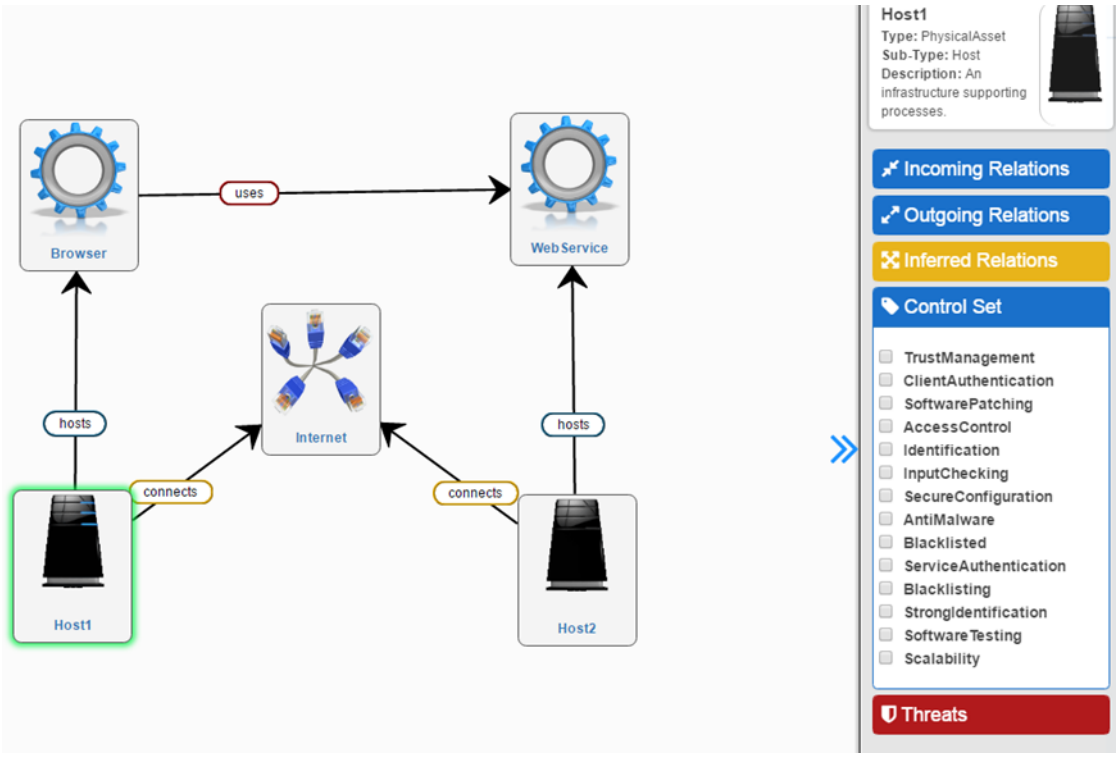


Figure 18 - Selecting Control Set properties for Host1

3.1.6 Model outputs

3.1.6.1 Generate report

The *Generate Report* function is activated by clicking on the "blue report" button (see Table 3).The report contains information about the current system setup. This is feature is not implemented in the current release.

3.1.6.2 Obtain network diagram

Allows retrieval of the network diagram as it would appear in the editor canvas. This feature is not implemented in the current release.

3.2 Programmer Guide

N.B. The presented *Trust Builder* is not programmable: there is no API that can drive the system programmatically.

4 Unit Tests

4.1 Information about Tests

In this section we describe a case study represent simple scenario demonstrating the use of Trust Builder tool. The main tasks that we need to complete for each case study are:

a) Constructing the model
b) Validating the model
c) Addressing the threats
d) Interpreting the outcome of threat elimination

The interpretation of the output is beyond the scope of this test.

4.2 Case Study

4.2.1 Constructing a model

In this section we analyse a typical case for ICT systems representing the threats associated with accessing a web page hosted on a remote server. Constructing a security model involves placing assets on the canvas and establishing connections between them. First step we open a new model by clicking on the “Create New Model” link (see Figure 6). Then we drag the assets in the modelling panel and make connections (see Figure 14). The model itself consist of two hosts connected to the internet. The user on Host1 accesses a Web Service deployed on Host2. This is a simple model, in essence downloading a web page from a remote web server. The reasons for selecting this model are as follows:

a) Frequently occurring case, typical for all web applications
b) Simplicity
c) All types of inference can be well demonstrated, these are:
    - Inferred assets
    - Inferred relationships
    - Mandatory relationships
d) The threats inherent to the model are well understood
e) The effect of controls and threats can be easily interpreted

The model is shown below in Figure 19. The model consists of

- two physical asset “Host” assets, re-labelled to be “Host1” and “Host2”;
- the “Wired Network” physical asset, labelled “Internet”;
- the generic “Logical Asset” logical asset, labelled “Browser”;
- the “Web Server” logical asset, labelled “Web Service”.

The assets are linked together as shown.

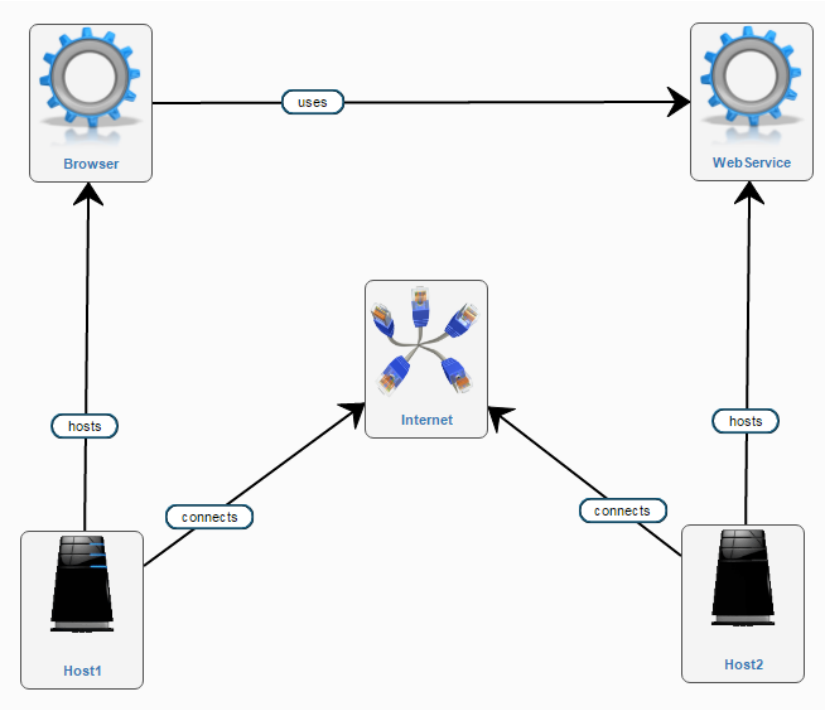


Figure 19 – Test model

4.2.2 Validating the model

The validation operation checks the model and generates inferred assets and relationships if they were missing from the initial model. For example if there are two assets connected and the user had missed out an asset between them, this inferred asset will be automatically added by the validator.

The red "play" button must be pressed. The validation operation can take some seconds. The outcome of the validation operation is indicated by changing the colour of the boundary around the assets. A green colour indicates the validation has succeeded, otherwise the colour is red. All issues identified by the validation operation need to be addressed and the model validated again.

In this case we need to pay attention to the connection between the Browser and Web service (the "uses" label is framed by red line). Clicking on the "uses" relation reveals an inferred "Service Pool" asset which must have its relations to the asserted assets (Browser and Web Service) defined. This is indicated by the red background of Incoming Relations in Figure 20.

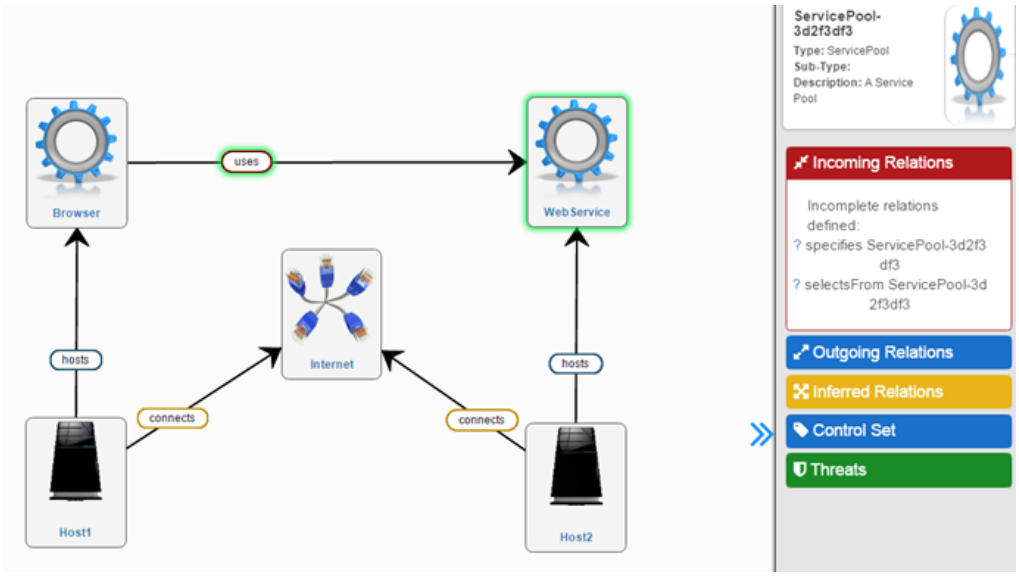


Figure 20 - Incoming relations must be defined on the inferred asset

The blue question marks indicate missing information. By clicking on the first blue question mark we can specify that it is the Browser which "specifies" the Service pool (inferred asset), see Figure 21.

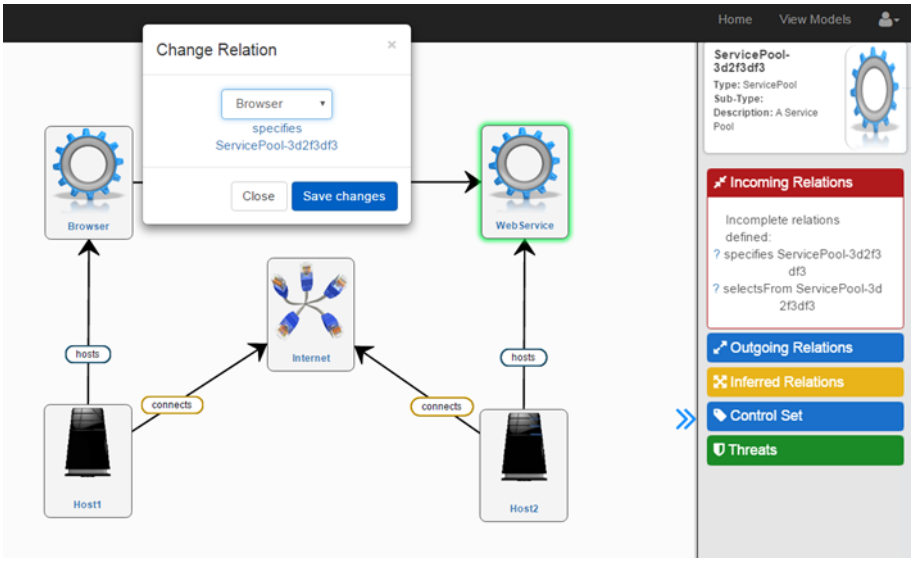


Figure 21 – Establish connection between Browser and Service Pool

By clicking on the second blue questions marks we can specify a relationship between the Web Server and Service pool (inferred asset), see Figure 22.

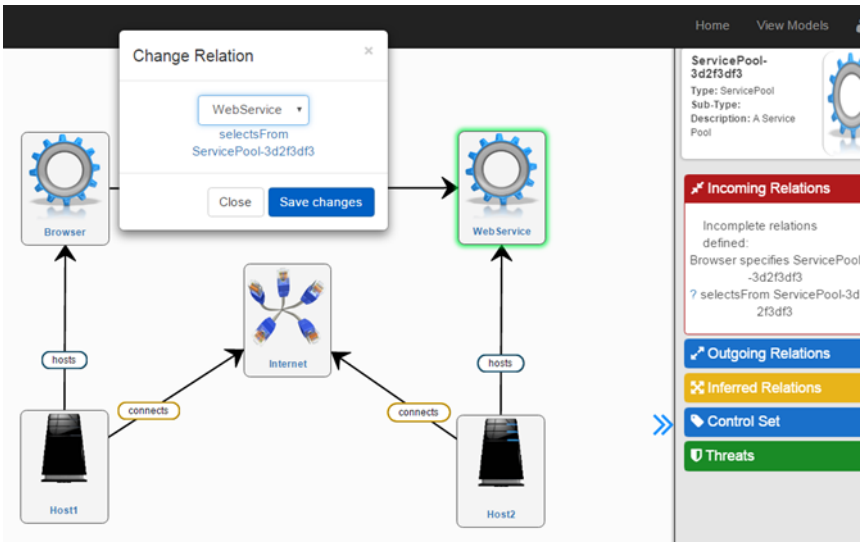


Figure 22 – Establish connection between Web Service and Service Pool

After establishing connections to the Service Pool (inferred asset) the background colour of Incoming Relations should turn blue and the "uses" relation label should change from red to yellow to indicate that an inferred asset can be found at that relationship but that the inferred asset is fully specified. In the following step we validate the model again by clicking on the red play button at the bottom of the screen (see Figure 23).

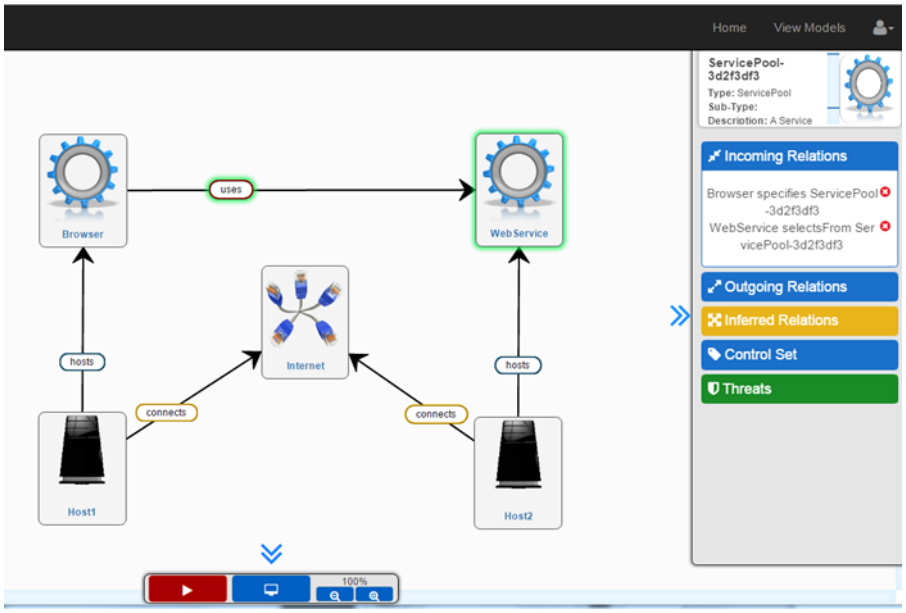


Figure 23 - Setting all Incoming Relations

4.2.3 Addressing the threats

The validation step also generates a list of threats that need to be resolved. Resolving threats is achieved by specifying the controls to use. The effect of specifying a control is that the threat becomes either eliminated or its likelihood of occurring significantly reduced. After validating the model the forward button turns green indicating the model is correct. In the following step we need to resolve the threats for each asset. For example by clicking on Host1 and the Threats tab we can see a list of threats associated with Host1 (see Figure 24).

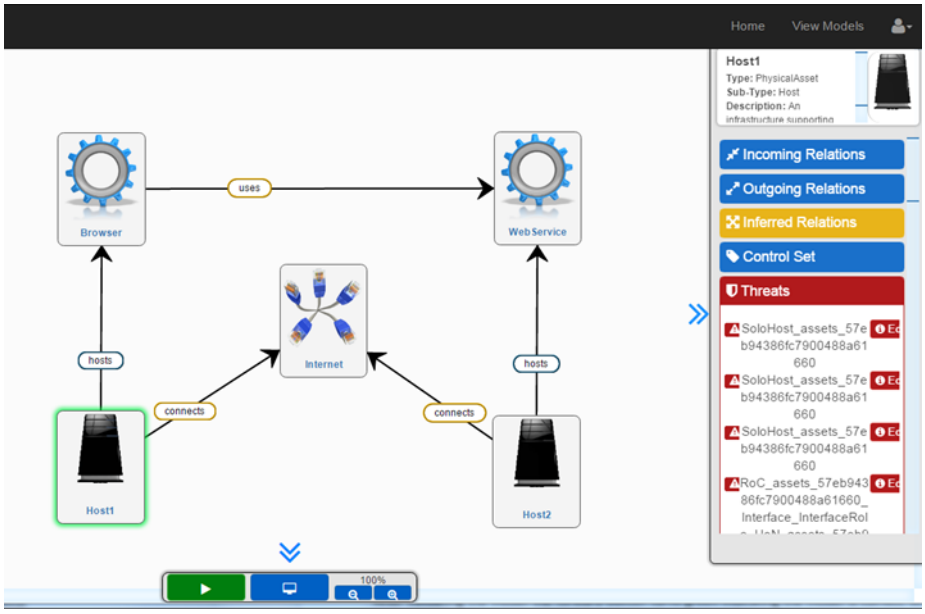


Figure 24 - Threats associated with Host1

By clicking on the red Edit button to the right of a threat, the user can see more details about the selected threat. The threats can be resolved by selecting options under the Control Set tab. A list of controls available for Host 1 is given in Figure 25.

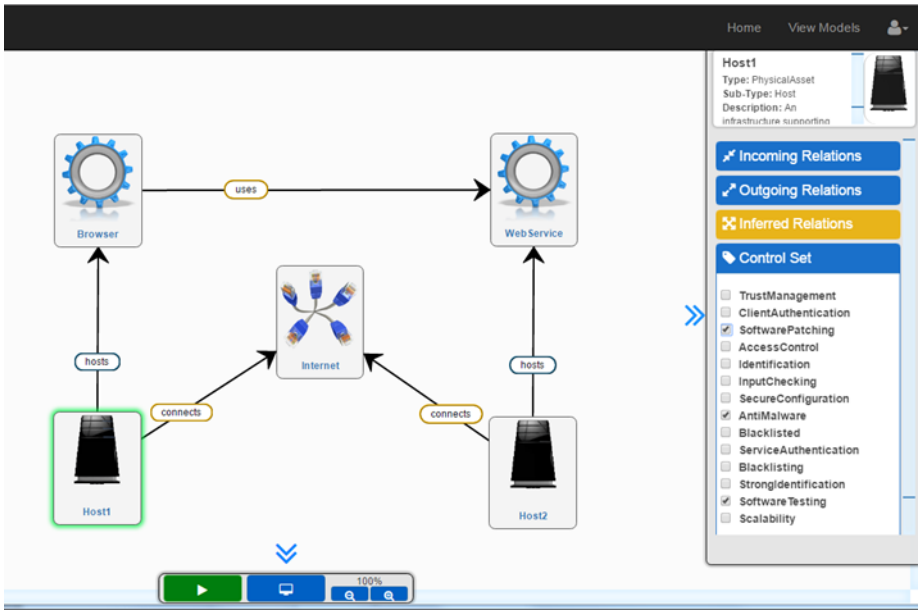


Figure 25 - Control set for Host1

For the purpose of demonstration we can select three options (Software Patching, Anti Malware and Software Testing) and see which threats will be resolved. These threats are indicated by green colour (see Figure 26).

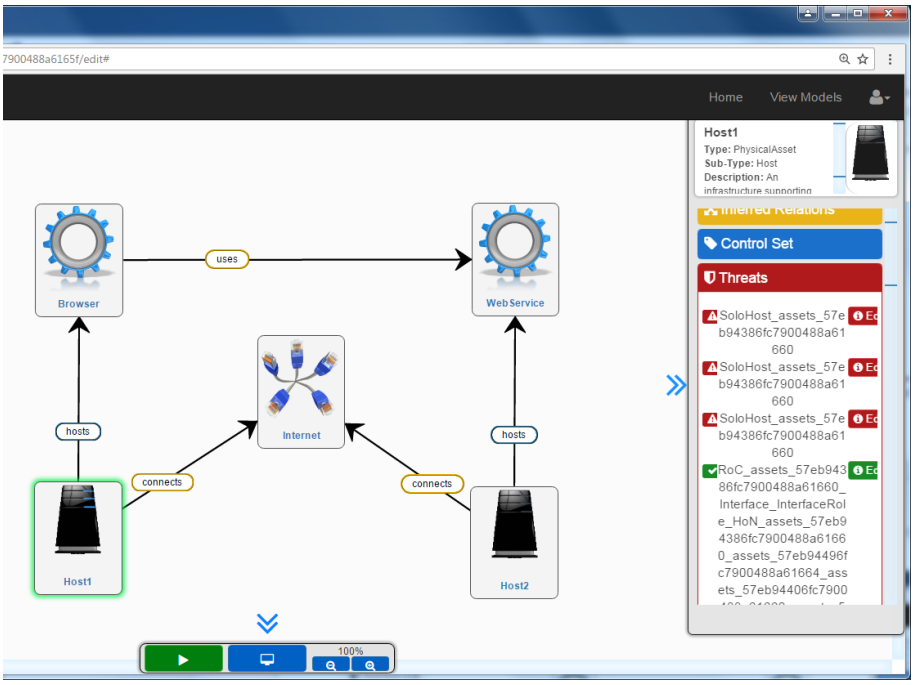


Figure 26 - Resolving threats

The screenshot in Figure 26 shows only one threat resolved. By scrolling down the list, one can see that there are four threats resolved as a result of selecting Software Patching, Anti Malware and Software Testing from the Control set.

Resolving the threats is an iterative process, the user needs to go through the assets one by one, selecting the options from the Control Set and checking which threats have been eliminated. In this User Guide due to the limited space we have described the threat resolution steps for one asset (Host1) but these steps are applicable to all assets. By following these steps the user should be able to resolve or at least mitigate all threats associated with the given Trust Model.

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

5G-IPPP	5G Infrastructure Public Private Partnership
DoS	Denial of Service

7 References