D2.1

Model of interdependencies in the electricity supply chain – Tool User Manual

DOCUMENT	D2.1	**WORKPACKAGE**	WP2
DELIVERABLE STATE	FINAL	**PROGRAMME IDENTIFIER**	H2020-SU-DS-2020
REVISION	V0.5	**GRANT AGREEMENT ID**	101020560
DELIVERY DATE	30/04/2022	**PROJECT START DATE**	01/10/2021
DISSEMINATION LEVEL	PU	**DURATION**	3 YEARS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 101020560

DISCLAIMER

ACKNOWLEDGEMENT

This document is a deliverable of CyberSEAS project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101020560.

The opinions expressed in this document reflect only the author's view and in no way reflect the European Commission's opinions. The European Commission is not responsible for any use that may be made of the information it contains.

PROJECT ACRONYM	CyberSEAS
PROJECT TITLE	Cyber Securing Energy dAta Services
CALL ID	H2020-SU-DS-2020
CALL NAME	Digital Security (H2020-SU-DS-2018-2019-2020)
TOPIC	SU-DS04-2018-2020 Cybersecurity in the Electrical Power and Energy System (EPES): an armour against cyber and privacy attacks and data breaches
TYPE OF ACTION	Innovation Action
COORDINATOR	ENGINEERING – INGEGNERIA INFORMATICA SPA (ENG)
PRINCIPAL CONTRACTORS	CONSORZIO INTERUNIVERSITARIO NAZIONALE PER L'INFORMATICA (CINI), AIRBUS CYBERSECURITY GMBH (ACS), FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V. (FRAUNHOFER), GUARDTIME OU (GT), IKERLAN S. COOP (IKE), INFORMATIKA INFORMACIJSKE STORITVE IN INZENIRING DD (INF), INSTITUT ZA KORPORATIVNE VARNOSTNE STUDIJE LJUBLJANA (ICS), RHEINISCH-WESTFAELISCHE TECHNISCHE HOCHSCHULE AACHEN (RWTH), SOFTWARE IMAGINATION & VISION SRL (SIMAVI), SOFTWARE QUALITY SYSTEMS SA (SQS), STAM SRL (STAM), SYNELIXIS LYSEIS PLIROFORIKIS AUTOMATISMOU & TILEPIKOINONION ANONIMI ETAIRIA (SYN), WINGS ICT SOLUTIONS INFORMATION & COMMUNICATION TECHNOLOGIES IKE (WIN), ZIV APLICACIONES Y TECNOLOGIA SL (ZIV), COMUNE DI BERCHIDDA (BER), COMUNE DI BENETUTTI (BEN), ELES DOO SISTEMSKI OPERATER PRENOSNEGA ELEKTROENERGETSKEGA OMREZJA (ELES), PETROL SLOVENSKA ENERGETSKA DRUZBA DD LJUBLJANA (PET), AKADEMSKA RAZISKOVALNA MREZA SLOVENIJE (ARN), HRVATSKI OPERATOR PRIJENOSNOG SUSTAVA DOO (HOPS), ENERIM OY (ENERIM), ELEKTRILEVI OU (ELV), COMPANIA NATIONALA DE TRANSPORT ALENERGIEI ELECTRICE TRANSELECTRICA SA (TEL), CENTRUL ROMAN AL ENERGIEI (CRE), TIMELEX (TLX).
WORKPACKAGE	WP2
DELIVERABLE TYPE	OTHER
DISSEMINATION LEVEL	Public
DELIVERABLE STATE	FINAL
CONTRACTUAL DATE OF DELIVERY	30/04/2022
ACTUAL DATE OF DELIVERY	25/04/2022
DOCUMENT TITLE	Model of interdependencies in the electricity supply chain – Tool User Manual
AUTHOR(S)	RWTH
REVIEWER(S)	ENG, STAM
ABSTRACT	SEE EXECUTIVE SUMMARY

HISTORY	SEE DOCUMENT HISTORY
KEYWORDS	

Document History

Version	Date	Contributor(s)	Description
V0.1	12/01/2022	RWTH	First draft
V0.2	13/01/2022	SYN, RWTH	Mapping of CyberSEAS asset classes to vulnerability classes of CyberSEAS in Section 2.2 using standards.
V0.3	18/01/2022	INF, RWTH	Included Annexure on AHP in this document. Received filled questionnaire for AHP analysis - *CyberSEAS_T2_1_Criteria_comparision_Informatika_v01.docx* and review of *CIA_AssetClasses_Comparison_AHP_Informatika.xlsx*.
V0.4	21/02/2022 – 24/03/2022	ACS, SYN, RWTH	Implemented suggenstions on improvements to the tool and input excel file. Updated the document to accommodate the same. Also added connection to other tasks.
V0.4	19/10/2021 – 13/01/2022	ALL	Provided reference to the final general list of assets in the document. (collection of a general list of smart grid assets along with its fit over SGAM. List populated together with partners during this task to also assist other tasks running in parallel).
V0.5	05/04/2022	ENG, STAM, RWTH	Incorporated internal review comments. Updated repository link in the document. Updated figure 3 and other minor changes.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AHP	Analytical Heirarchical Process
CL	Cascading Level
DER	Distributed Energy Resources
EPES	Electrical Power and Energy System
IED	Intelligent Electronic Device
IM	Information Management
PES	Power and Energy System
PLC	Programmable Logic Controller
SGAM	Smart Grid Architecture Model
VS	Vulnerability Score
WAN	Wide Area Network
WP	Work Package

Executive Summary

This document explains the contributions from Task 2.1 of project CyberSEAS, which is on the "Analysis of interdependencies in the electricity supply chain". To understand interdependencies, it is essential to begin by analyzing the type of assets involved in the electricity supply chain. With this motivation, first, assets of Electrical Power and Energy System (EPES) are categorized utilizing the Smart Grid Reference Architecture Model (SGAM). Various assets involved in the electricity supply chain and their spread over SGAM is analyzed. Next, to understand dependencies, a flexible framework to depict dependencies of different nature between assets is presented. Further, a tool for multi-directional graph-based representation of assets and dependencies of a given infrastructure is developed. The tool provides functionalities to analyze effects of the interdependencies in the electricity supply chain. This document additionally serves as a user manual for the usage of developed graph-based tool and presents capabilities of the tool.

1 Introduction

To ensure efficient and secure functioning of modern day grid, there has been increased interactions between stakeholders, exchange of data, new policies and higher digitization. This advancement has led not only to an increase in types of assets, but also to an increase in ways of interaction and dependencies between assets. To ensure secure and uninterrupted supply of electrical energy it is important to consider risks involved with different types of assets in the entire supply chain and their relations to one another. To this end, a graph-based tool has been developed which provides end users a graph-based visualization of their assets and dependencies. The tool provides functionalities such as

- finding possible paths from source to target asset using depth-first search,
- viewing cascading effects of a failure/attack on an asset and indicating the level of cascading,
- showing importance of an asset based on complex network theory indexes and asset class importance based on Analytical Heierarchical Process (AHP) weights.

1.1 Relation to other activities

Task 2.2 : The Task 2.2, *Vulnerability assessment on each electricity operator*, makes use of the asset categories identified in Task 2.1 and the overall list of smart grid assets. The asset categories from Task 2.1 are mapped to the vulnerability classes of NIST and ISO/IEC 27005 standards, which are further mapped to the vulnerability classes of CyberSEAS (as discussed in Section 2.2 of the present deliverable). Task 2.2 also utilizes the inventory of general set of assets collected in Task 2.1. The vulnerability assessment tool of Task 2.2 has implemented and enforced the connection between a) each vulnerability and b) a set of asset categories and the respective asset(s). This applies for the assessment of both the statically and dynamically (i.e., through pen tests) identified vulnerabilities. This way, the pilots can associate each (applicable) vulnerability with a set of assets. In this regard, the results of Task 2.1 are useful for Task 2.2, whose combined results can be further processed in subsequent tasks (example Task 2.3 or within WP3) which may consider the full scope of the threat/attack (including cascading effects, exploiting the relationships among the assets).

Task 2.3 : Task 2.3 utilizes the provided asset list from Task 2.1 for an extensive asset – MITRE attack technique mapping, in the context of analyzing cyber threat scenarios and their impacts across the electricity supply chain. Hereby, the extensive list of techniques is determined in their applicability to the specific assets. This provides the basis for the development of a multitude of attack scenarios and consequences. Furthermore, the input Excel sheet for the tool input has been revised to standardize the pilot asset and dependency data. This then enables the actual visualization of those assets in the tool from Task 2.1. This will be utilized in the collaboration with the pilots to generate possible threat scenarios, based on their infrastructure. The generated data also serves as input for further research regarding possible future threat scenarios. The cascading threat path approach hereby is essential to determine and visualize the impact propagation of attack scenarios within a network of assets.

2 CyberSEAS Asset Classes

2.1 SGAM Reference Architecture to CyberSEAS Asset Classes

The CEN-CENELEC-ETSI Smart Grid Reference Architecture Model (SGAM) [1] is used as a standard reference to categorize various types of assets involved in the smart grid. For this purpose, an analogy is drawn between the interoperability layer of SGAM and the nature/type of asset, and further based on this analogy; asset classes for CyberSEAS are derived. Figure 1 shows the adaptation of SGAM reference architecture for use in CyberSEAS project to define asset classes. As indicated, the interoperability layers are treated analogous to the asset nature.

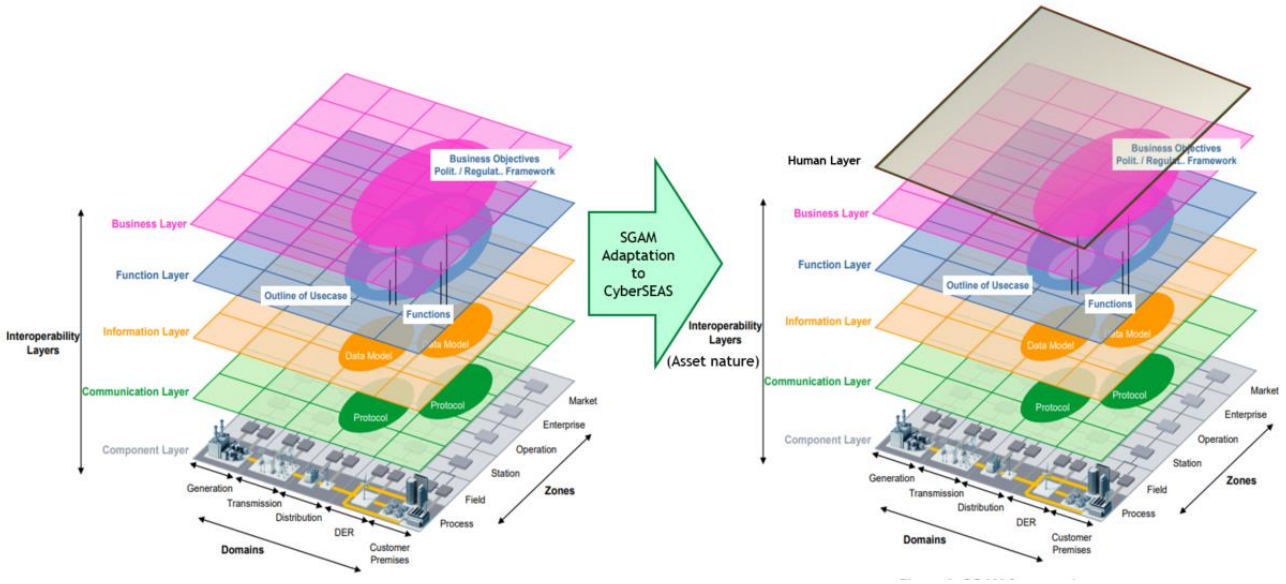

Figure 1: Adaptation of SGAM to categorize asset classes for CyberSEAS

Following are the asset classes derived from the SGAM reference architecture with examples:

1. **Power and Energy System (PES) Components**: These assets are mostly tangible and physical in nature. Assets, which are associated to the process zone and component layer of the SGAM architecture, are considered under PES Component asset class. Examples include generator, transmission line, transformers and loads.
2. **Information Management (IM) Components**: These assets are mostly tangible and physical in nature. Assets, which are associated to the zones field, station, operation, market or enterprise and to the component layer of the SGAM architecture, are considered under IM Component asset class. Examples include relays, PLC, IEDs, physical communication links, routers, gateways, computers and servers.
3. **Communication**: This asset class is derived by mapping logical communication networks across the SGAM grid plane to the communication layer of SGAM reference architecture. Therefore, such assets are considered under Communication asset class. These assets are mostly intangible and cyber or logical in nature. Examples may include wide area network (WAN), neighborhood area network (NAN) and field area network (FAN).

4. **Information**: This asset class is derived by mapping various data created and exchanged across the SGAM grid plane to the information layer of SGAM reference architecture. Therefore, such assets are considered under Information asset class. These assets are intangible and cyber in nature. Examples include measurement data, grid data, market data, customer information data, contractual agreements and various databases.
5. **Functional**: This asset class is derived by mapping various software executing different functionalities across the SGAM grid plane to the functional layer of SGAM reference architecture. Therefore, such assets are considered under functional asset class. These assets can be intangible and cyber in nature. Examples include state estimation programs, SCADA functions, optimal power dispatch programs and aggregation software.
6. **Business**: This asset class is derived by mapping various policies, processes, procedures and objectives across the SGAM grid plane to the business layer of SGAM reference architecture. Therefore, such assets are considered under business asset class. These assets are mostly organizational in nature. Examples include patching processes, asset management processes.

As CyberSEAS [2] aims to consider impacts of threats arising from and aiming at personnel involved in the complete operation of EPES, an additional asset class "Human" is introduced.

7. **Human**: This asset class consists of various personnel involved in different roles across the SGAM grid plane. Therefore, such assets are considered under human asset class. Examples include state network operators, maintenance personnel, customer service personnel and database administrators.

Typically, to determine, the asset class to which an asset belongs, the assets are mapped based on their functionality in the electricity supply chain to the SGAM-Plane of domain and zone, and based on the nature of asset to the layer of SGAM. For example, a generator, based on its functionality, belongs to the domain Generation, zone Process and based on its nature to the layer Component, therefore, it can be mapped to the asset class PES Component. For the purpose of graph-representation, assets form the nodes.

2.2 Mapping of CyberSEAS Asset Classes to CyberSEAS Vulnerability Classes

Based on ISO/IEC 27005 [3] vulnerabilities are classified according to the asset class they are related to. We can map the CyberSEAS Asset Classes (depicted on the left column in Figure 2) to the ISO/IEC 27005 Asset/Vulnerability Classes (depicted on the middle column in Figure 2) and, subsequently, map them to the CyberSEAS Vulnerability Classes (depicted on right column in Figure 2).

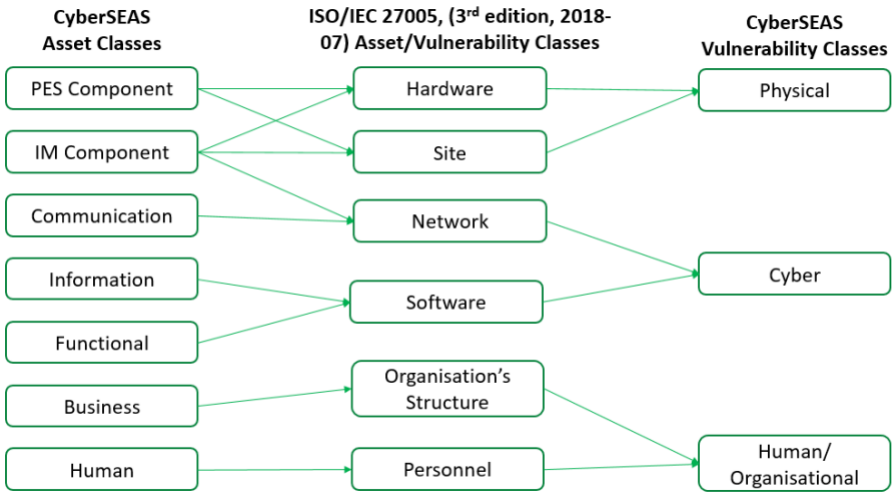

Figure 2: Mapping of CyberSEAS asset classes to vulnerability classes

3 Dependencies Between Assets

The relation between various assets is represented by directional edges in the graph-based representation. The convention of the edge direction is such that the direction of the arrow indicates the direction of propagation of a failure or threat. Therefore, by referring to Figure 3 (a), if the functioning of the supporting asset is impacted then, following the direction of the edge, the dependent asset may be impacted next.

The relation between two assets is analyzed in the light of following nature of dependencies [4]:

1. Physical
2. Logical
3. Cyber
4. Geographical
5. Social

The dependency types shown in Table 1 serve only as an example of possible dependencies between two assets.

Dependency Type	Dependency Nature	Edge Type	Example
Creates	Cyber	Unidirectional	an application creates a specific data, example load forecast
UsedBy	Cyber	Unidirectional	measurement data is used by energy management system applications
ProvidesSA	Social	Unidirectional	HMI provides situational awareness (SA) to the operator
WorksOn	Social	Unidirectional	field crew works on an equipment or operator works on SCADA application
BelongsTo	Logical	Unidirectional	an asset is a property that belongs to a company
Runs	Physical	Unidirectional	a server runs an application
supplies	Physical	Unidirectional	generator supplies power to the ICT devices
Stores	Physical/Cyber	Unidirectional	Datahub stores data
Carries	Physical/Cyber	Unidirectional	a communication link carries a data
NetworkHas	Cyber	Unidirectional	a communication network has many ICT devices
Connects	Physical/Logical	Bidirectional	to define physical/topological connection

Table 1: Dependency types

The dependency type describes the support action/relation verb which the "Supporting Asset" provides for the "Dependent Asset". The supporting asset, the dependency type, and the dependent asset, together help in building the graph in a user-friendly readable format. Any two node and an edge connecting between them can be read as follows: <Supporting Asset> <Dependency Type> <Dependent Asset>. Considering as an example of a dependency scenario where the meter data stored on a server, then this can be represented as follows:

Supporting Asset: Server

Dependency Type: stores

Dependent Asset: Meter data

This can be read as *server stores meter data*. In terms of graph representation, a directional edge named "stores" will be built from asset node "server" to asset node "meter data", and this would be indicative of the direction of propagation of the failure. That is, in this example case, if server is affected by an adversary, then the meter data stored on it may be affected next (following the direction of the arrow).

The dependency type is to be specified such that each it explains the purpose of the dependency or relation from supporting asset to the dependent asset. For the dependency

type "Connects", the tool automatically constructs bidirectional arrow (two unidirectional arrows, in opposite direction) between the assets, indicating a physical or topological connection between the assets. For other dependency types, only a single unidirectional arrow is constructed from supporting asset to dependent asset. This is shown in Figure 3(b). To consider the case that two assets may have more than a single relation, multi-directed graphs are selected to represent assets and their relations. It is to be noted that the dependency types are not limited to Table 1, and the user of the tool may provide dependencies beyond this, however must be consistent with the edge direction convention specified here to obtain a meaningful graph-representation.

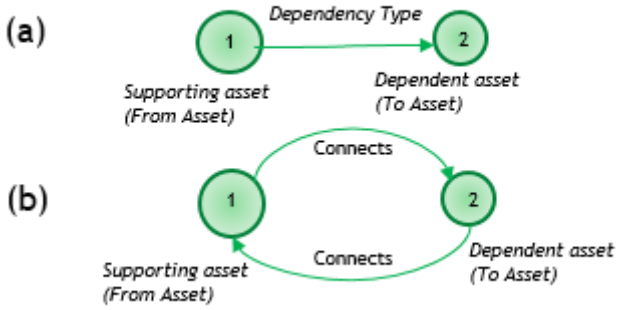

Figure 3: Building of multi-directed graph (a) Building of a unidirectional edge for all dependency type except "Connects". (b) Building of two unidirectional edges for dependency type *Connects*.

4 Smart Grid Assets and Dependencies Model Tool

4.1 About the tool

This tool enables the user to visualize their infrastructure assets and their dependencies in a graph-based representation. The tool uses graph-based indexes and AHP (Annex 1) outcomes to indicate the criticality of asset and asset classes respectively.

The tool is developed using Python scripting language and following are the packages used:

1. User interface and display: Streamlit
2. Graph visualization: PyVis
3. Graph processing: NetworkX
4. Array handling and data manipulation: NumPy and Pandas
5. Excel read and write operations: openpyxl

The Repository of code is available here:

https://git.rwth-aachen.de/acs/public/software/sgadm

Other files related to the task are available here:

https://git.rwth-aachen.de/acs/public/deliverables/cyberseas/sgadm-example-data

Following documents are available in this repository for reference:

1. Example of assets and dependencies input-excel sheet: *asset_dep_example.xlsx*. Kindly make use of this excel sheet for providing asset and dependencies information to the tool.
2. Presentation of Task 2.1 : *CyberSEAS_Task2_1.pptx*
3. AHP questionnaire response: *CyberSEAS_T2_1_Criteria_comparision_Informatika_v01.docx*
4. Weights calculated using AHP technique: *CIA_AssetClasses_Comparison_AHP_Informatika.xlsx*
5. User-manual : *CyberSEAS_Deliverable_2.1.pdf*
6. General list of EPES assets : "*General_List_of_Assets_SGAM_v04.xlsx*"

The tool can be installed by following steps given in the *readme.md* file of the code repository. The tool can also be run without the network connectivity.

4.2 Input To The Tool

The user of the tool must use the *asset_dep_example.xlsx* and provide information on their infrastructure assets and dependencies or of a given scenario. This excel sheet will serve as input to the tool (henceforth referenced in this document as the input-excel sheet). The rest of this section guides on preparation of input-excel sheet.

The input-excel sheet must consist of two tabs named "*Assets*" and "*Dependencies*".

The "*Assets*" tab must contain a list of assets with the following columns:

1. "ID" – serial numbering of assets. (Integer)

2. "Asset Name" – *Unique* name of asset.
3. "Domains" – Association of asset to the domains of SGAM. Selectable from drop-down in *asset_dep_example.xlsx* input-excel.
4. "Zones" – Association of asset to the zones of SGAM. Selectable from drop-down in *asset_dep_example.xlsx* input-excel.
5. "Layers" – Association of asset to the nature of asset. This must be a single entry. Selectable from drop-down in *asset_dep_example.xlsx* input-excel.
6. "Location" – Location of the asset, if relevant, otherwise select "None". (String)
7. "Vulnerability score" – vulnerability score of the asset (Real Number), if any, otherwise select "0".

An asset may have multiple entries (separated by comma) for columns *Domains*, *Zones*, *Locations*. For column *Layers*, since the Layers in this case associates to the nature of the asset, the entry for *Layers* column must be unique. For example, if we have an asset which is a software tool for aggregation function, as the aggregation function spans DER, Distribution and customer domains, the corresponding *Domains* and *Columns* entries for this software asset in input-excel sheet may be specified as follows: *Domains* column may have "*Distribution,DER,Customer*" and "*Functional*" for the *Layers* column entry. In case, the *Location* and *Vulnerability Score* are not available, the user should select "*None*" and "*0*" in the input-excel respectively.

The "*Dependencies*" tab must contain list of dependencies with following columns:

1. "ID" - serial numbering of assets. (Integer).
2. "From Asset" – name of supporting asset (as given in "*Assets*" tab, "*Asset Name*" column). Selectable from drop-down in *asset_dep_example.xlsx* input-excel.
3. "To Asset" - name of dependent asset (as given in "*Assets*" tab, "*Asset Name*" column). Selectable from drop-down in *asset_dep_example.xlsx* input-excel.
4. "Dependency Type" – Dependency type between these two assets as per support relation or action verb.

For reliable graph-representation and functionalities, the user of the tool must follow the convention of the direction of edges explained in section 3 while entering dependencies in input-excel sheet. The infrastructure specific input-excel sheet can be imported to the tool using the button shown in Figure 4.

For the purpose of displaying tool working and functionalities in this document, a hypothetical list of assets and dependencies as given in *asset_dep_example.xlsx* is used as input.

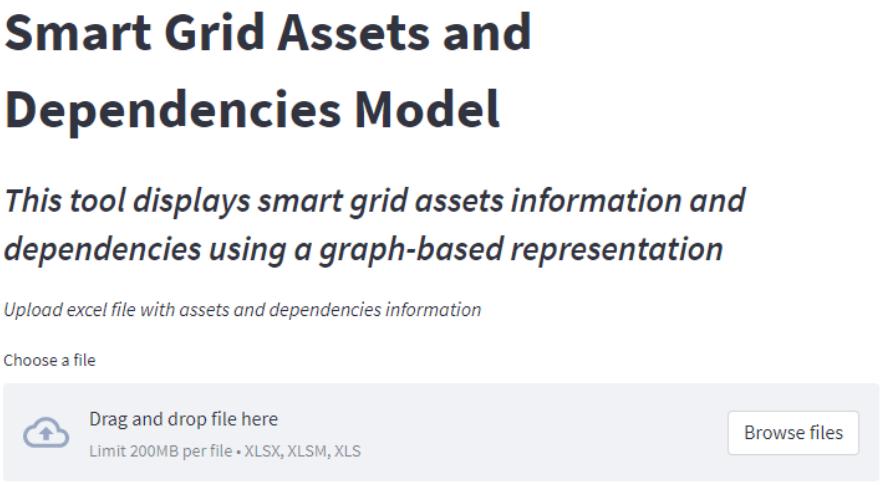

Figure 4: Smart Grid Assets and Dependencies Model tool with button to upload input-excel sheet with assets and dependencies

4.3 Tool Functionalities

Once the input-excel sheet is uploaded, three sections with different functionalities are constructed as shown in the Figure 5. Following sub-sections explain each functionality in detail.

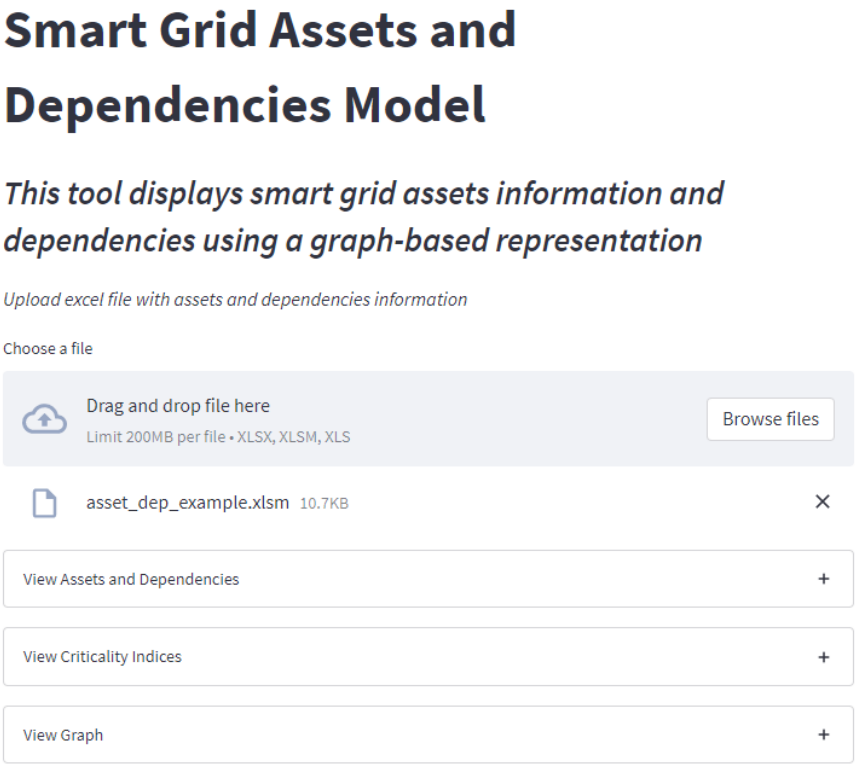

Figure 5: Three sections of the tool, View Assets and Dependencies, View Criticality Indices and View Graph

4.3.1 View Assets And Dependencies

This section of the tool displays in tabular form the list of assets and list of dependencies as imported from the input-excel sheet, as shown in Figure 6. Users may view number of assets distribution across SGAM domains, zones, layers (asset nature), via the checkbox *view assets distribution*.

Figure 6: *View Assets and Dependencies* section of the tool

4.3.2 View Criticality Indices

4.3.2.1 Local Versus Global Criticality of an Asset

The local criticality and global criticality of an asset is inferred from the graph-based indexes namely node-out degree centrality and closeness centrality respectively.

In complex network theory, the out-degree centrality of a node indicates the fraction of nodes its outgoing edges are connected to. In the graph-representation where assets are represented as nodes of the graph and edges are represented as the dependencies, the out-degree centrality of a node(asset) can serve as an indication of first level cascading outreach starting from that asset, and therefore referenced here as the local criticality of an asset.

The closeness centrality of a node is the reciprocal of the average shortest path distance of this node to all other nodes in the network. This is an indication of how close (number of edges) a given node is to other nodes in the network. In the graph-representation of assets and dependencies, closeness centrality serves as an indication of levels of cascading required from a given asset to reach all other assets in the network and therefore referenced here as the global criticality of an asset.

The out degree centrality and closeness centrality are found using functions from NetworkX package networkx.algorithms.centrality.out_degree_centrality and networkx.algorithms.centrality.closeness_centrality respectively [5].

4.3.2.2 Asset Class Criticality

The criticality of an asset class for each security objective (Confidentiality, Integrity, Availability) is inferred based on:

i. Number of dependencies (total number of out-going edges) in which assets from a given asset class support assets from other asset classes.
ii. Weights calculated for importance of a security objective for asset class and importance of asset class for a security objective

Here it assumed that all dependencies types affect the security objectives equally. Refer to Annex 1 for more information on finding asset class criticality.

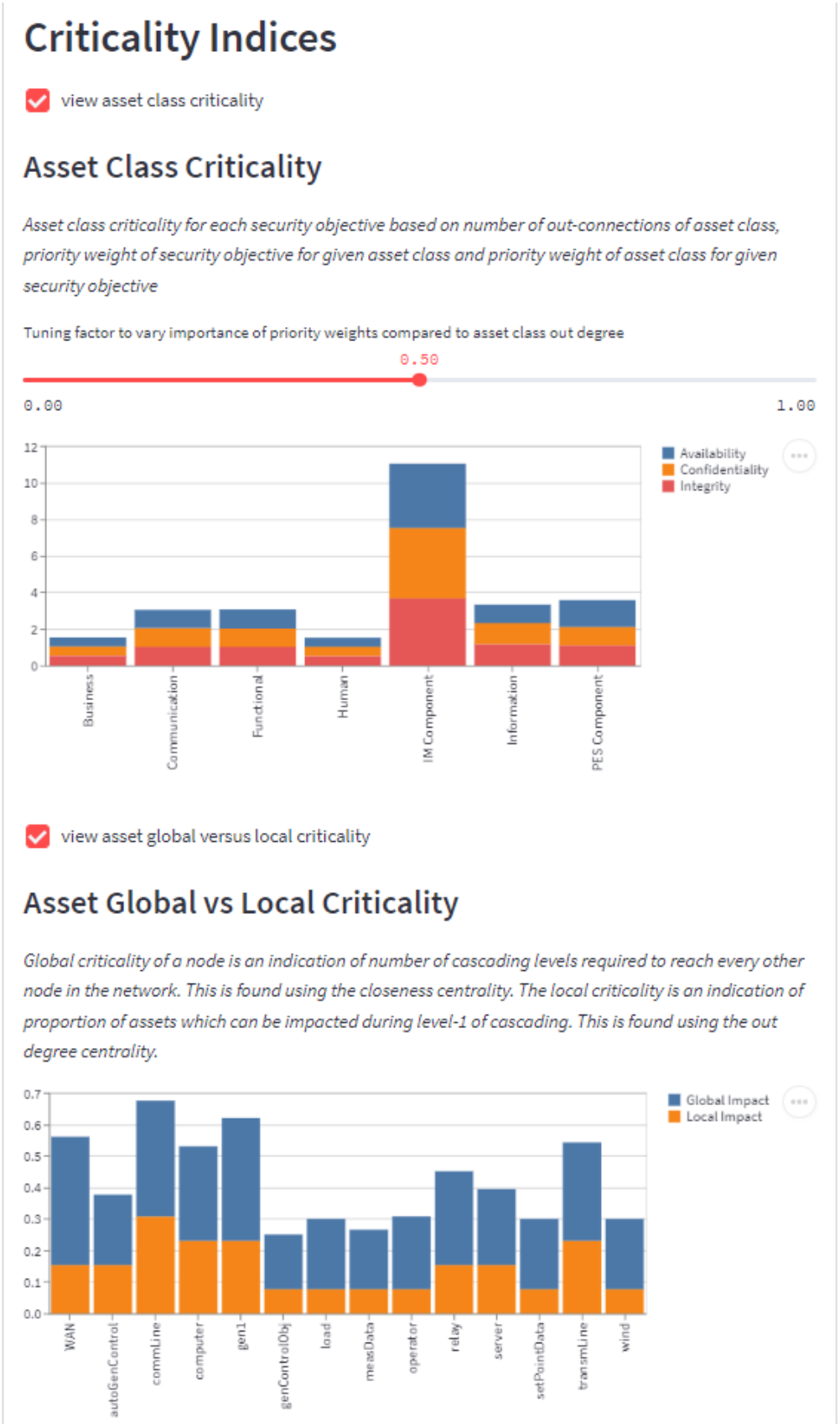

Figure 7: *View Criticality Indices* section of the tool

4.3.3 View Graph

Figure 8 shows the tool snapshot from the section of *View Graph*. This section consists of three sub-sections with various functionalities, which are explained in this section.

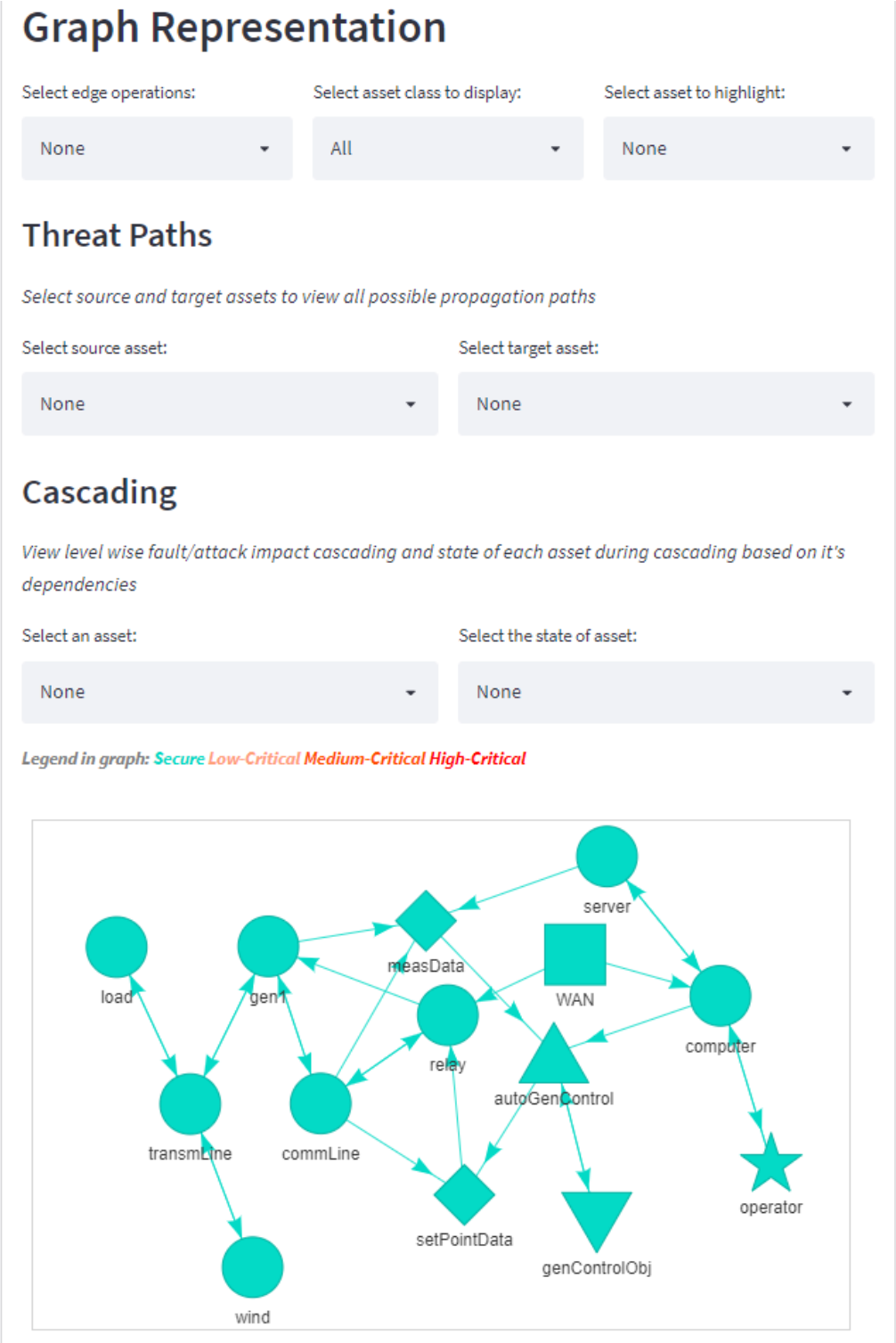

Figure 8: *View Graph* section of the tool

4.3.3.1 Graph Representation

From the imported assets and dependencies list, the tool constructs a multi-directed graph, with each asset represented as a node of the graph and dependencies as the edges. The *Graph Representation* sub-section provides user with functionalities to help orient in the graph visualization:

a) *Select edge operations* - This selection box provides two option, to *Show each dependency* and to *Adjust edge length*. The *Show each dependency* option enables the user to view each dependency as a separate edge between assets on graph. The *Adjust edge length* option enables user to vary the edge length by dragging nodes apart.
b) *Select asset class to display* – This selection box enables the user to view only assets and dependencies from the selected asset class.
c) *Select asset to highlight* – This selection box enables the user to view a particular asset on the graph by highlighting the selected asset in a different color.

The node shapes are drawn based on the nature of asset that the node represents and is as follows:

1. PES Component and IM Component : Circle
2. Communication: Square
3. Information: Diamond
4. Functional: Triangle
5. Business: Inverted Triangle
6. Human: Star

By hovering over an edge on the graph-representation, the dependency type represented by the edge is displayed, and by hovering over the nodes, the vulnerability score (VS) of the asset is displayed.

The generated graph representation can be saved as an image file using save options provided on right-click.

4.3.3.2 Threat Paths

This functionality enables the user to find possible paths between two assets (named source and target asset in the tool) by using depth-first search algorithm on the graph model. This functionality can be helpful to define and analyze threat scenarios.

User must select source asset and target asset via the select boxes named *Select source asset* and *Select target asset* on the user interface as shown in Figure 9. The outcome is a table consisting details on paths found between the source and target assets and a select box *Select path to highlight* to enable user to view each path on the graph-representation. Following information are displayed per path:

a) Domains: Number of domains traversed in the path
b) Zones: Number of zones traversed in the path
c) Layers: Number of types of assets traversed in the path
d) Location: Locations of the assets traversed in the path
e) Total Assets: Total number of assets in the path including the source and target assets
f) CIA order: Order of security objective C,I,A priority for the group of assets in the path. Refer to Annex 1 for more information.

g) Group Out Degree: This is group-out degree centrality for the group of assets in the path, calculated using function networkx.algorithms.criticality.group_out_degree_centrality [5]. Group out degree centrality is the fraction of non-group members connected to group members by outgoing edges. This is indicative of the proportion of assets affected by first level cascading, if threat traverses along this path. This can help the user to investigate an extension of the evaluated scenario, to analyse how many other assets are within the reach of the attacker while executing the original attack.
h) Vulnerability Score: Sum of the vulnerability score of assets traversed in a given path

To avoid larger execution times, a limit on maximum number of paths between any two assets for displaying as output on the tool is set to 200.

User may select a path from the *Select path to highlight* select box to view on the graph-representation. The source and target assets are displayed in a different colour than other assets in the path for easier interpretation. Upon selection of a path, the tool also displays additional information about the path such as names of domains, zones, asset classes, asset nature of assets traversed in the path, as shown in Figure 9.

An option is provided to export the details of paths (domains, zones, layers (nature of assets) and locations traversed in a path, total assets of a path, names of assets in a path, CIA order, group-out degree, total vulnerability score of assets of a path) generated into an excel sheet using the button *Export paths details*. The exported excel sheet is saved under the project folder as *paths_result.xlsx*.

Threat Paths

Select source and target assets to view all possible propagation paths

Select source asset: autoGenControl

Select target asset: gen1

All possible paths from asset autoGenControl *to asset* gen1

	Domains	Layers	Zones	Locations	Total Assets	CIA Order	Group Out D
1	1	3	3	1	5	C, I, A	0
2	1	3	3	1	4	A, I, C	0

Export paths details

Select path to highlight: 2

Domains: Generation

Zones: Station,Field,Process

Asset Nature/Layers: Functional,Information,Component

Locations: Area1

Asset Classes: PES Component,IM Component,Information,Functional

Figure 9: List of paths and path details from source to target asset

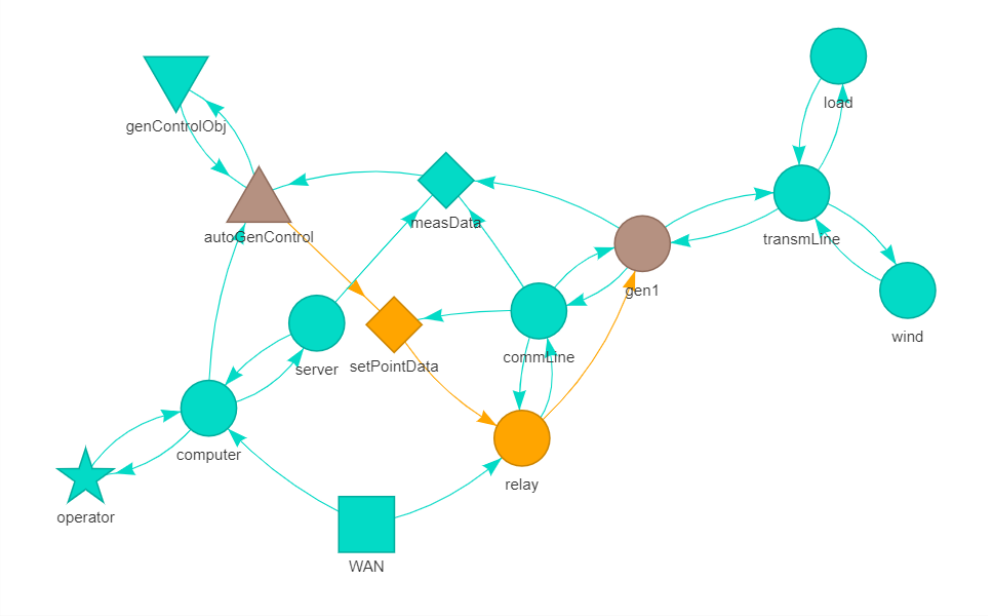

Figure 10: Graph-visualization of path selected in Figure 9

4.3.3.3 Cascading

The tool enables the user to view level wise cascading effect of the failure or attack due to the dependencies between assets. Along with that, for each cascading level, the state of the asset is also assessed based on states of the dependencies it requires to function.

The state of all out-going edge of an asset assumes the state of the asset.

The states of the assets (and their out-going edges) are categorized as Secure, Low-Critical, Medium-Critical and High-Critical. The states of an asset is assessed based on following rules:

1. Secure:
    a. if all of its required dependency (in-coming) edges are in state of Secure
2. Low-Critical:
    a. if more than 0%, but less than 50% of the incoming edges of a given dependency type are either Low-Critical or Medium-Critical or High-Critical
3. Medium-Critical:
    a. if >=50% but <=100% of incoming edges of a given dependency type are either Low-Critical or Medium-Critical or >= 50% and <100% of a given dependency type are High-Critical. For example, let us consider an asset "*data*" stored on 4 servers, "*server1*", "*server2*", "*server3*" and "*server4*". Then asset "*data*" has dependency on assets "*server1*", "*server2*", "*server3*" and "*server4*" of dependency type "*stores*". That is, the asset "*data*" has 4 incoming edges of type "*stores*". If 3 servers are, say, high-critical, then the asset "*data*" will be marked as medium-critical, since at least 1 server is still in Secure state.
    b. One incoming edge from each type of dependency is either Low-Critical or Medium-Critical or High-Critical
4. High-Critical:
    a. All incoming edges of a given dependency type are in High-Critical state
    b. All incoming edges are either Low-Critical or Medium-Critical or High-Critical

To view cascading starting from an asset, the user must select an asset and initial state of the asset from the select box option named *Select an asset* and *Select the state of asset* respectively as shown in Figure 11. The tool then provides list of cascading levels in a select box (*Select cascade level* to view in Figure 11), from which the user may select to view selected cascading level and states of the assets in that particular cascading level on the graph representation. The state of the asset and edges are displayed using a color legend shown in Figure 11. User may hover over an asset in graph-representation to know the cascading level (CL) in which the asset attained the state indicated by the color.

Cascading

View level wise fault/attack impact cascading and state of each asset during cascading based on it's dependencies

Select an asset:

measData

Select the state of asset:

Low-Critical

Legend in graph: ***Secure*** ***Low-Critical*** ***Medium-Critical*** ***High-Critical***

Select cascade level to view:

2

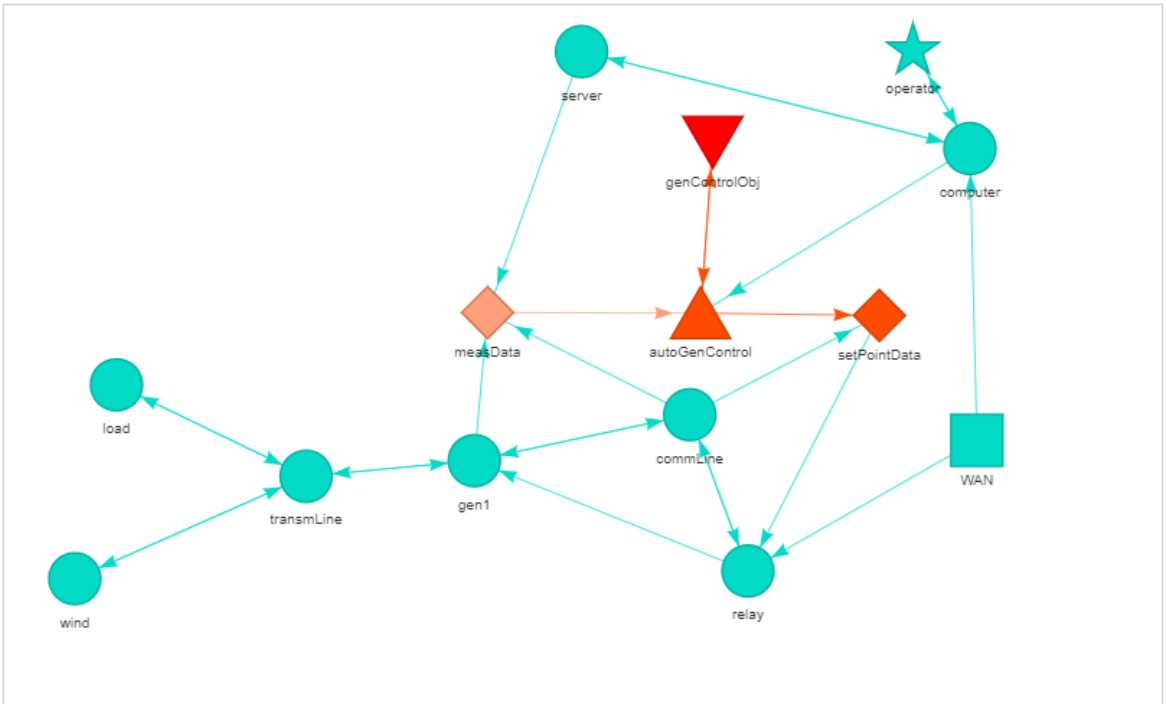

Figure 11: Graph visualization of cascading

5 References

[1] CEN-CENELEC-ETSI Smart Grid Coordination Group, Smart Grid Reference Architecture, 2012.

[2] CyberSEAS, H2020- 101020560, Grant Agreement, 2021.

[3] ISO/IEC, "Security techniques-Information security risk management," ISO/IEC FIDIS 27005, 2018.

[4] Rinaldi, S. M, J. Peerenboom and T. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Systems Magazine,* vol. 21, no. 6, pp. 11-25, 2001.

[5] A. A. Hagberg, D. A. Schult and P. J. Sw, Exploring network structure, dynamics, and function using NetworkX, in Proceedings of the 7th Python in Science Conference (SciPy2008).

Annex 1 Analytical Hierarchical Process

Analytical hierarchical process is a multi-criteria decision making process. It is based on pair-wise comparison between criteria. The security objective triad according to standard IEC 62443-1-1 are confidentiality, integrity, and availability. The priority of security objectives differs depending upon the assets and scenarios in question. Here, AHP technique is used to understand and evaluate following objectives:

A. What is the order of importance of security objective for each type of asset classes.
   a. Example: *For asset class PES Components, is 'Availability' of the asset more important compared to the 'Integrity' of the asset for reliable and secure operation of entire energy supply chain?*
   b. Outcome: *Security objective priority weights per asset class* - $(w_c)_j, (w_i)_j$, $(w_a)_j$; *j is the asset class*
B. What is the order of importance of asset classes for each security objective.
   a. Example: *Is 'Availability' of PES Component more important compared to 'Availability' of IM Components for reliable and secure operation of entire energy supply chain? Similarly, for comparisons for 'Confidentiality' and 'Integrity'*
   b. Outcome: *Asset class priority weights per security objective* - $(w_j)_c, (w_j)_i$, $(w_j)_a$; *j is the asset class*

A questionnaire was distributed to the experts for performing the necessary comparisons and further weights were calculated as per AHP technique. The weights calculated using AHP is used for following purposes:

1. To compare the Confidentiality, Integrity, Availability order (CIA Order) for a group of heterogeneous assets of a network

$$C_{group} = \frac{1}{size(group)} * \sum_{j=1}^{ac} \left((w_c)_j * h_j\right)$$

$$I_{group} = \frac{1}{size(group)} * \sum_{j=1}^{ac} \left((w_i)_j * h_j\right)$$

$$A_{group} = \frac{1}{size(group)} * \sum_{j=1}^{ac} \left((w_a)_j * h_j\right)$$

$sort(C_{group}, I_{group}, A_{group})$ ← Order of security objective for this group of assets

C_{group}, I_{group}, A_{group} : Confidentiality, integrity, availability scores for a group of assets

$(w_c)_j$, $(w_i)_j$, $(w_a)_j$: weights of confidentiality, availability, integrity for asset class j

ac : number of asset classes

h_j : number of assets of asset class j in the group

2. Asset Class Criticality with respect to security objectives:

$$(ACC_j)_c = (1-\alpha) * supp_j + \alpha * n_j * (w_j)_c * (w_c)_j$$

$$(ACC_j)_i = (1-\alpha) * supp_j + \alpha * n_j * (w_j)_i * (w_i)_j$$

$$(ACC_j)_a = (1-\alpha) * supp_j + \alpha * n_j * (w_j)_a * (w_a)_j$$

$(ACC_j)_c$, $(ACC_j)_i$:, $(ACC_j)_a$ asset class criticality of asset class j for security objective of confidentiality, integrity and availability respectively

n_j : number of assets in asset class j

$supp_j$: number of outwards connections from assets of asset class j to assets from other asset classes

$(w_j)_c$, $(w_j)_i$, $(w_j)_a$: is the weight of asset class j for security objective confidentiality, integrity and availability respectively.

$(w_c)_j$, $(w_i)_j$, $(w_a)_j$: is the weight of security objective confidentiality, integrity and availability respectively for asset class j.

α : tuning factor, [0,1]

In the above equations, α is the tuning factor between 0 to 1, which can be used to vary importance between two parts of the equation. If α is made equal to 1, it means the asset class criticality is calculated based on only weights of the asset class as found using AHP technique in Annex 1. And if α is made equal to 0, it means the asset class criticality for a given security objective of the system is calculated based on only the number of ways in which assets from asset class j are supporting assets from other asset classes.