Security of Air Transport Infrastructures of Europe

D7.8 – Specification of the Impact Propagation Model

Deliverable Number	D7.8
Author(s)	FHG, DLR, ALS, ERI, INOV, AIA, NIS, SAT, CCS, KEMEA, SEA, ZAG
Due/delivered Date	M18/2020-10-30
Reviewed by	KEMEA, DLR
Dissemination Level	PU
Version of template	1.06

Start Date of Project: 2019-05-01

Duration: 27 months

Grant agreement: 832969

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 832969

Document contributors

No.	Name	Role (content contributor / reviewer / other)
1	Corinna Köpke (FHG)	Content contributor
2	Tim Stelkens-Kobsch (DLR)	Content contributor
3	Andrea Roland (FHG)	Content contributor
4	Eric Herve (ALS)	Content contributor
5	Filipe Apolinario (INOV)	Content contributor
6	Victoria Peuvrelle (ERI)	Content contributor
7	Nikos Papagiannopoulos (AIA)	Content contributor
8	Meilin Schaper (DLR)	Content contributor, reviewer, quality review
9	Kelly Burke (NIS)	Content contributor
10	Leonidas Perlepes (SAT)	Content contributor
11	Thomas Oudin (ACS)	Content contributor
12	Eftichia Georgiou (KEMEA)	Content contributor, reviewer
13	David Lancelin (ACS)	Reviewer
14	Mirjam Fehling-Kaschek (FHG)	Reviewer
15	Alexander Stolz (FHG)	Reviewer
16	Elena Branchini (SEA)	Content contributor
17	Olga Carvalho (INOV)	Content contributor
18	Marko Licina (ZAG)	Content contributor
19	Vasileios Kazoukas (KEMEA)	Security review
20	Sven Hrastnik (ZAG)	Reviewer

Document revisions

Revision	Date	Comment	Author
V0.1	2020-09-24	Initial draft – updating D2.5	Corinna Köpke
V0.2	2020-10-06	Reviewed by NIS and ZAG, some further content removed by FHG	Corinna Köpke Kelly Burke Sven Hrastnik
V0.2	2020-10-12	Security check and advice for changes	Vasileios Kazoukas, Project Security Officer
V0.3	2020-10-19	Review comments from KEMEA and SEA are applied	Corinna Köpke Elena Branchini
V1.0	2020-10-29	Final quality check and approval for submission	Meilin Schaper, Quality Manager

Executive summary

This report presents and discusses the Impact Propagation Model, which is developed in Task 2.4 to be finally implemented in an executable software tool in Task 5.1. The main findings described in this report are the following:

- A hybrid model has been developed to represent the airport processes. It consists of a layer model (airplane, baggage, information and people), a network model and an agent-based model. The model is based on an already existing asset list (compare D2.3) and a review on modelling approaches.
- For the quantification of resilience, key performance indicators (KPIs) have been discussed. It was found that for the real-time assessment of system's performance passenger-related indicators such as waiting time or space per person in waiting areas is more suitable than long-term financial KPIs. Also graph-based KPIs are suggested based on a review included in this report.
- The business impact assessment with a focus on impacts of cyber-threats has been described and the interfaces to the impact assessment model need to be further defined. Also, the interrelations to the Incident Management Portal (IMP) and the Crisis Alerting System (CAS) are discussed in this deliverable.

Finally, the Impact Propagation Model is reviewed in three different ways:

- The network model has been analysed with respect to impact propagation for each scenario based on cyber-physical threat scenarios. The main outcome of this validation is that complete asset lists and complete interrelation description between assets are essential for the impact propagation. With this data available, the developed model is able not only to provide insights on where an impact propagates but also how fast, i.e. through how many steps, it propagates.
- The agent-based model was reviewed and enriched from a societal and human viewpoint. It is recommended to consider social and psychological aspects to strengthen the meaningfulness of the model.
- The interfaces between the Impact Propagation Model and the two adjacent tools on the simulation platform, namely the Incident Management Portal and the Crisis Alerting System, are discussed. It is concluded that the Impact Propagation Model will add valuable information to the systems.

Table of Content

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
ABM	Agent-based model
ABC	Automatic border control
AC	Access control
ADF	Airport Development Fund
AIA	Athens International Airport
ALS	Alstef
AOC	Airport Operation Centre
AODB	Airport operation data base
ATCO	Air traffic control
ATM	Air traffic management
BC	Betweenness centrality
BHS	Baggage handling system
BIA	Business Impact Assessment
BP-IDS	Business process intrusion detection system
BPMN	Business Process Modelling Notation
BSM	Baggage source message
CaESAR	Cascading Effects Simulation in urban Areas to assess and increase Resilience
CAS	Crisis Alerting System
CCTV	Closed-circuit television
COTS	Commercial of the shelf
CPS	Cyber-physical system
D	Deliverable
DAG	Directed acyclic graph
DB	Database
DoA	Description of Action
DOS	Denial of service
EDS	Explosive detection system
ENT	Emergent norm theory

Acronym	Definition
FIDS	Flight information display system
GDPR	General data protective regulation
IAM	Impact assessment module
IT	Information technology
IDS	Intrusion detection system
IMP	Incident Management Portal
IoT	Internet of things
ISO	International Organization for Standardization
KPI	Key-Performance Indicator
MAS	Multi-agent system
MIA	Mission impact assessment
MITM	Man in the middle
NIS	Network Integration & Solutions S.r.l.
NMAP	Network Mapper
OSI	Open system
PA	Public announcement
PLC	Programmable logic controller
RMS	Resource management system
SAC	Sort allocation computer
SCADA	Supervisory control and data acquisition
SOC	Security Operation Centre
SysML	System Modelling Language
T	Task
UML	Unified modelling language
ULD	Unit load device
VTAC	Virtual terrain assisted impact assessment for cyber attacks
ZAG	Zagreb

1 Introduction

The project SATIE aims to develop a security toolkit to face combined cyber-physical threats in a coordinated and effective way supported by a shared situational awareness system. One part of the toolkit is the impact propagation system to simulate the impact of cyber-physical threat scenarios and to study cascading effects in the airport's systems.

Within this deliverable the Impact Propagation Model developed in Task 2.4 "Definition of an impact propagation and decision support model" is described which will be later implemented in software within the Impact Propagation Simulation (T5.1). To this end, the impact propagation in different areas including airport services, passenger flow and societal impact will be analysed. This analysis will be then included in a model implementable by a software tool. For defining the model, the airport infrastructure and its services are analysed as well as the impact of possible threats (see Task 2.2). Impacts are divided in categories and the dependencies between the airport services are analysed.

In chapter 2, the general approach of quantifying the resilience of the airports infrastructure towards specific threats is described. The resilience management process is based on risk management and comprises nine overall steps that are represented by different Tasks in this project whereby the impact assessment is one of the steps.

In chapter 3, a general review of modelling approaches, graph-based performance measures, impact propagation techniques and past airport incidents is provided. Possible modelling approaches such as conceptual models, probabilistic models, network or graph models, SysML models and layer models are described.

Chapter 4: In contrast to what was described in the DoA, the model is a combination of three main approaches, that is (i) a layer model, (ii) a network representation and (iii) an agent-based model (ABM). The layer model consists of four main layers, which are information, baggage, aircraft and people flow, whereby people can be divided into passengers and employees.

To build scenario and airport specific network structures, a template has been developed that was completed by the airports. Still, the collection of network information for implementation and validation will continue in the course of this project, particularly in Task 5.1. Especially, the interrelation between assets has been studied and still needs further analysis. The collection of information is less focused on the four airport systems mentioned in the DoA and more on scenario specific assets which facilitates the delivery of information for the airports and naturally leads to an analysis of asset interrelations beyond system boundaries as several systems are involved in each scenario.

The model is checked and validated by using cyber-physical threat scenarios. Further, the compliance between potential simulation outputs and the Incident Management Portal and the Crisis Alerting System (D5.3 and D5.4) is verified. Based on the model, the software developed in the course of this project will be able to perform an Impact Propagation Simulation.

Finally, chapter 5 summarizes the findings and provides an outlook to the upcoming tasks with respect to impact propagation.

2 Resilience management in the context of SATIE

The resilience management can be derived from risk management processes described in standard ISO 31000 (1800) (International Organization for Standardization, 2018) and comprises nine general steps (Häring, et al., 2017) visualized in Figure 2.1 (compare D2.4).

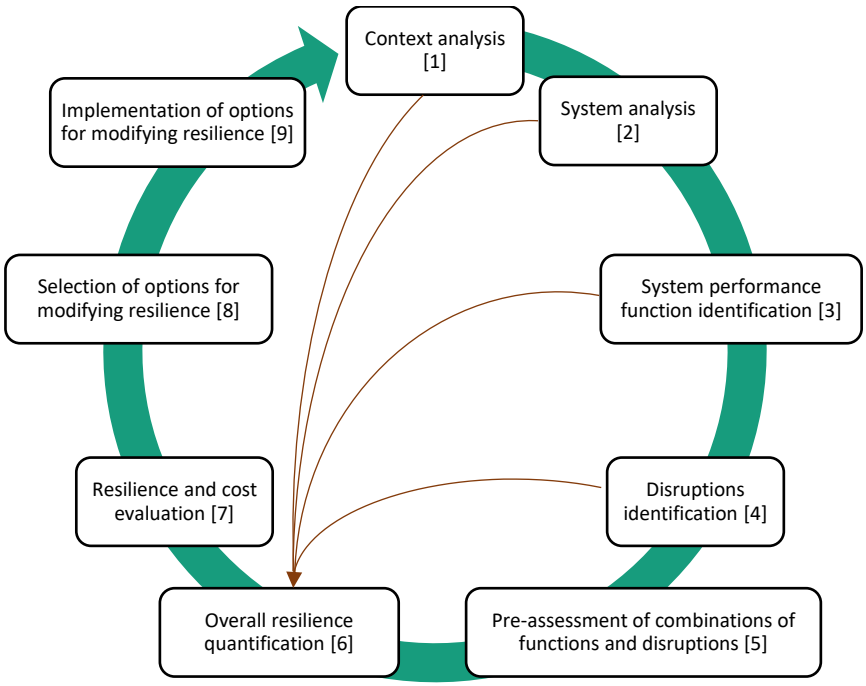

Figure 2.1: Resilience management cycle following (Häring, et al., 2017).

2.1 Context analysis

This first step in the resilience management cycle involves a general description of the system, identification of stakeholders and definition of scenarios to consider. This information can be found in the DoA and in the risk assessment (see Deliverable D2.3 – Cyber-physical risk analysis).

2.2 System analysis

This step comprises the analysis of system components, functions and interfaces for modelling the system. Based on lists of system components (see D2.3), so called assets, that were established during the risk assessment, the interrelations of these assets were analysed (see Annex 1) which also define the interfaces between systems. In D2.3, assets have been categorized into different types and operations have been defined for each asset which represents the functionality. The system analysis also involves the description of recovery times for each asset. Here some high-level recovery times in case of failure are collected for main assets and systems with respect to the considered scenarios for the three different airports. They range from 40 minutes to 160 hours, where the latter is e.g. the case for specific components of the BAGWARE system. BAGWARE® is a trademark by ALS, a sort allocation computer (SAC) software that is employed to sort bags, to control the baggage handling system, and for the screening of checked baggage.

System layout maps are given exemplarily for Athens International Airport in Figures 2.2 and 2.3. These maps are needed to model passenger movement in the airport area during normal operation and in the case of incidents.

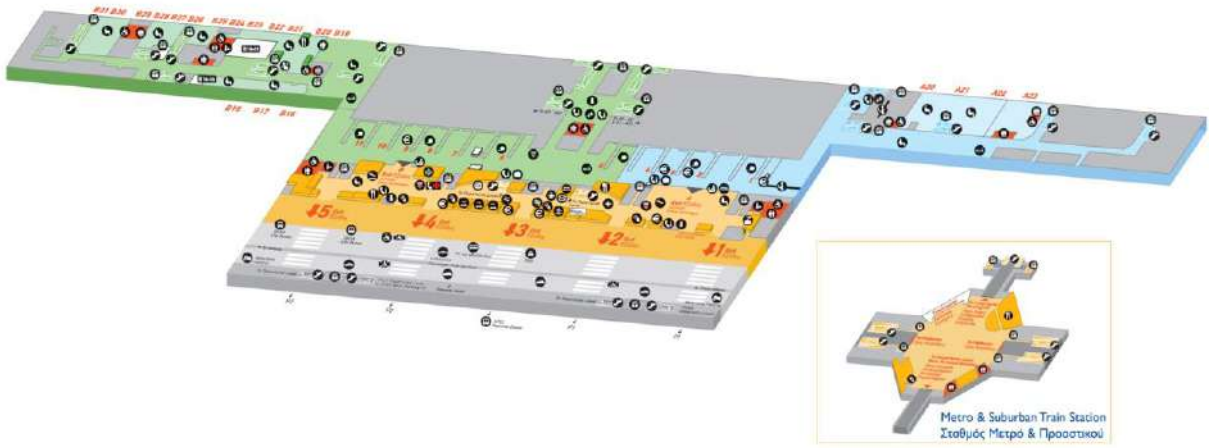

Figure 2.2: Athens International Airport arrivals layout (http://www.athensflights.gr/images/arrival-terminal.pdf).

Figure 2.3: Athens International Airport departure layout (http://www.athensflights.gr/images/departure-terminal.pdf).

An agent-based model (ABM) is developed to represent passengers and employees in the airport. This ABM represents one of four main layers that have been identified in discussions with NIS and the airports to summarize the processes in an airport, that is, (1) aircraft-flow, (2) baggage-flow, (3) information-flow and (4) people-flow which can be divided into passenger-flow and employee-flow (see Figure 2.5). Each flow layer contains several assets which are interrelated within in the flow layer but also between flow layers. Interrelations are defined in the way that if one asset fails/degrades which other assets will be affected, i.e. disturbed/degraded/confused/destroyed.

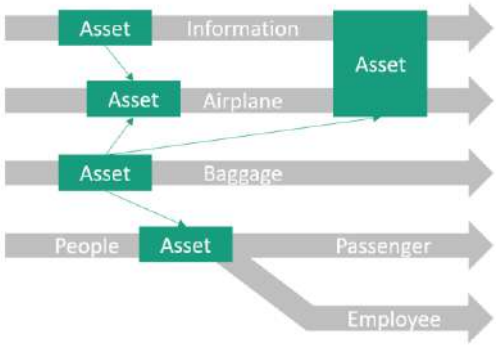

Figure 2.4: Flow layers identified for a generic airport. Each layer contains several assets which can also be included into two or more flow layers dependent on their function. Assets in one layer are connected through the respective flow and they are interrelated between flow layers.

To collect the information of asset interrelations scenario-specific excel sheets based on the asset lists of Task 2.2 have been distributed to AIA, SEA, ZAG and ALS. The asset lists with interrelations can be found in Annex 1. Based on the network information, scenario specific graph structures have been developed.

2.3 System performance function identification

Performance functions are data that are monitored as a function of time and that provide insights to the system's performance. Availability of a service or the number of service interruptions are common examples for performance functions or so-called performance indicators. The identification of performance functions can be based on existing KPIs. Performance functions can also be based on network and graph analysis (see section 3.2) or they can be derived from the ABM used here to represent passenger movement such as e.g. waiting times at check-in desks or security check-points. Further, the flow layers represent the airport's functionality and can be used to summarize system's performance.

Airport specific KPIs can be financial or can be developed from an operational, airport service or stakeholder perspective. KPIs are typically derived in score cards. For Athens International Airport, from an operational perspective the focus is on the aviation safety and airports efficiency which is represented by the KPIs system availability and the maximum duration of a single event failure. The airport service perspective defines airport service quality as KPI. The KPI from stakeholder perspective is corporate sustainability which is based on e.g. climate change actions, local communities' actions and human resources development.

For the quantification of resilience in real-time or near real-time, performance measures or KPIs are needed that can be measured in near real-time and that are informative for the current situation of the airport's systems.

2.4 Disruptions identification

This step comprises the analysis of threats, hazards and disruptions. Airport specific threats have been identified in the scenarios and within the risk assessment (D2.3). Threats have been defined with respect to their effect on specific assets and lead to potential disruptions. During the scenarios,

disruptions are identified corresponding to the incidents processed by the Incident Management Portal (Task 5.3). Incidents, which are validated alerts, are sent to the Impact Propagation Simulation (Task 5.1) based on defined ontologies (Task 4.1).

2.5 Pre-assessment of combinations of functions and disruptions

This step is a combination of step 3 and 4 to identify critical combinations. As an example, for a pre-assessment, the threats for each asset have been weighted with respect to their impact on the asset, which is derived from the risk assessment in D2.3. Impacts are assigned to each threat acting on a specific type of asset and vary between 1 and 10, whereby 1 stands for a low impact and 10 for a high impact. This information is combined with the airport's performance represented by flow layers. Independent of the frequency of each threat, the impact enables to pre-assess the significance of threats and to focus the modelling towards critical combinations of threats and flow layers. Figure 2.7 shows the results of the analysis of 80 general threats and their impact on specific assets that are grouped in different flow layers. The impact is summed for all affected types of asset in one flow layer and normalized by the number of types of asset. In Figure 2.7, there is one outlying threat with a significant impact affecting most assets on all flow layers, which is social engineering. Figure 2.8 shows the impact of some specific threats on the types of asset in the flow layers. Again, social engineering appears to have the largest impact on every flow layer. The importance of social engineering can be explained regarding the results shown in Figure 2.7. The overall impact of social engineering is with an average of almost 8 quite high but more significantly, it affects almost every type of component in an airport. This is because a corrupted, bribed or threatened person, having access to the airport's facility, has the potential to interact with nearly every type of asset.

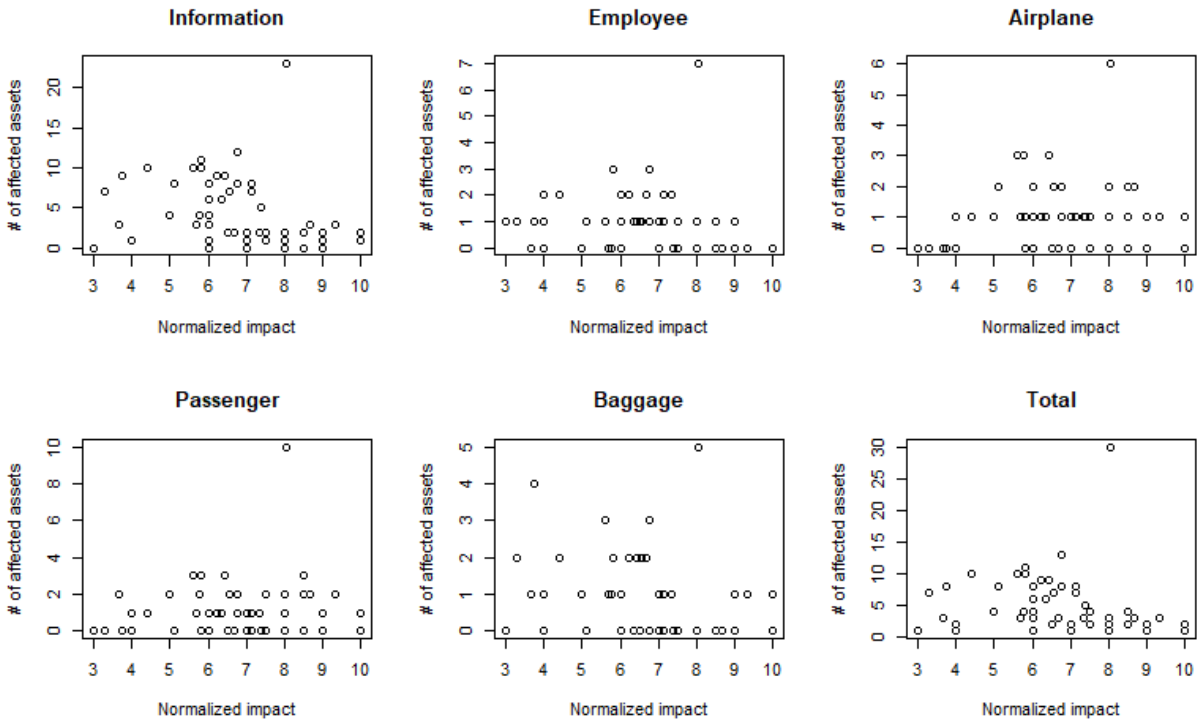

Figure 2.5: The impact for a collection of 80 threats for a generic airport infrastructure on each flow layer and in total.

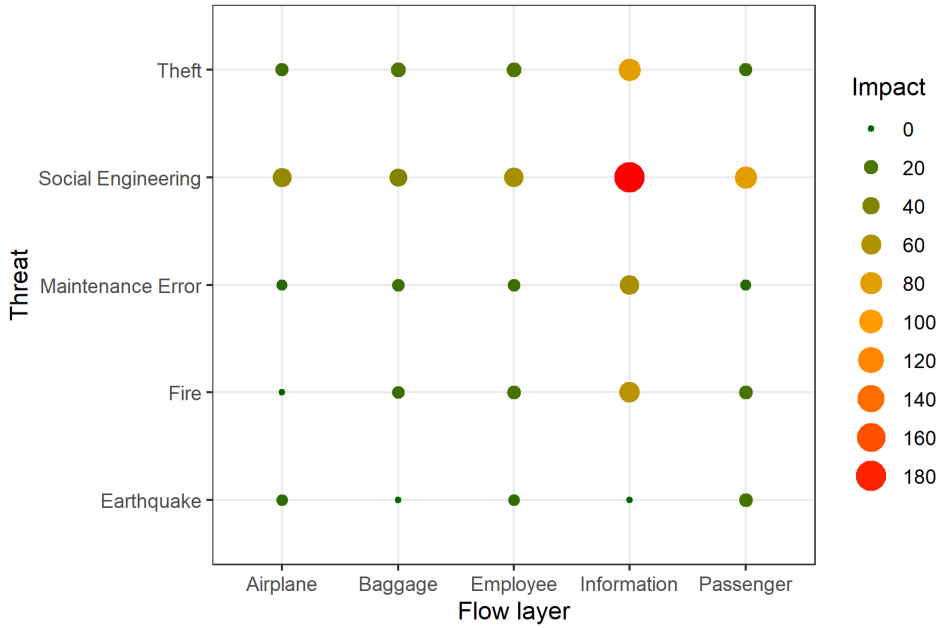

Figure 2.6: The impact as a function of threat and flow layer for five selected threats from the list of 80 threats.

2.6 Overall resilience quantification

In this step, all the information of step 1 to 4 is combined and implemented in a simulation environment, which is used to simulate the airport system's performance under threat scenarios. As already introduced in these steps, here ABM is employed to represent the people movement in the airport in combination with the simulation tool CaESAR (Cascading Effects Simulation in urban Areas to assess and increase Resilience) which simulates cascading effects in critical infrastructure and across infrastructure borders (Hiermaier, Hasenstein, & Faist, 2017). The implementation of the developed model takes place in Task 5.1.

In SATIE, this step corresponds to the Impact Propagation Simulation, which will be triggered by the incoming incidents.

2.7 Resilience and cost evaluation

This step combines findings of step 5 and 6 and compares the system's performance during a threat scenario under varying conditions considering uncertainties. Resilience curves (see D2.4) are plotted based on the simulation results in step 6 whereby repeated simulations provide different results because of uncertainties in the model. Especially for ABM, random processes are implemented which will lead to uncertain resilience curves. Further, resilience curves with different initial-conditions could also be analysed, such as e.g. varying passenger densities, to compare the impact of a threat during calm periods and rush hours.

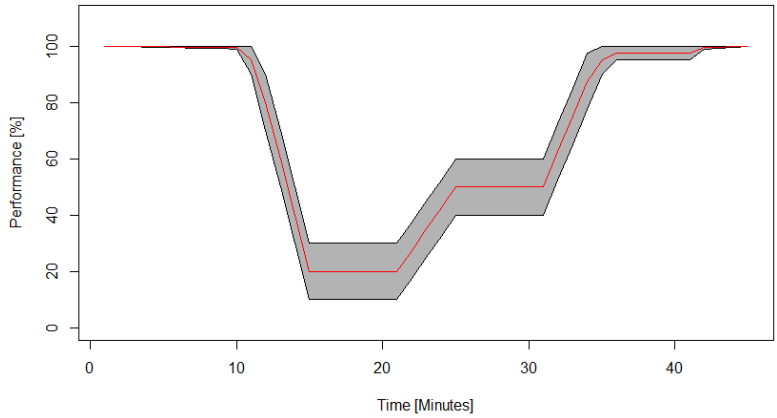

Figure 2.7: Example resilience curve with uncertainty band. The mean is given as red line and the variance is given in grey.

2.8 Selection of options for modifying resilience

Here, different mitigation strategies and improvement measures are compared. These could be e.g. redundancies for critical components in the system, security and maintenance procedures, detection of threats and corresponding early-warning. To analyse the effect of identified mitigation measures, simulations can be used which result in different resilience curves and might provide a base for the decision support (see Figure 2.10). Mitigation measures in the airport could be e.g. the order of repair actions or additional information desks in the airport area that make the system more resilient against certain threats.

In Figure 2.10, the alternative mitigation strategy offers a faster recovery and makes the system more resilient with respect to the specific threat. However, in this example, larger uncertainties have to be considered. The uncertainties result from unknowns in the model such as e.g. the initial position of passengers in the airport, which might influence the impact propagation. These parameters will be chosen randomly in each simulation based on pre-defined distributions. Thus, repeated runs of the simulations will be required and enable to provide uncertainty estimates for the resilience curves.

Finally, a human decides on the measure to be implemented as financial aspects, legislation and uncertainties have to be considered.

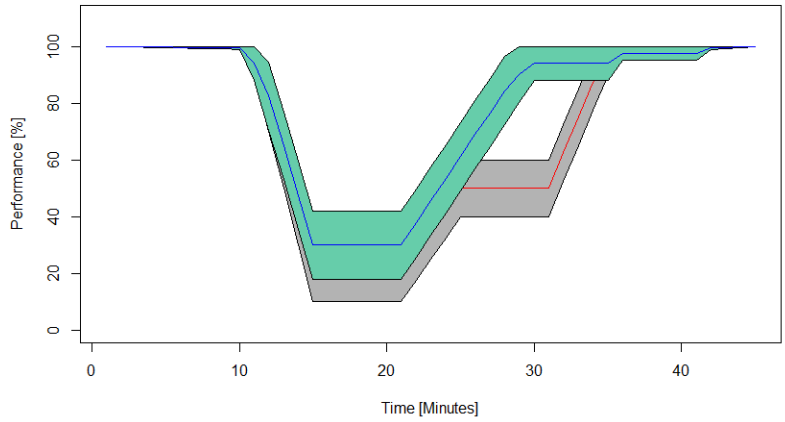

Figure 2.8: Example resilience curves for different mitigation strategies. The blue curve with variance visualized in green represents an alternative mitigation strategy, which minimizes the area above the curve, but it involves larger uncertainties.

2.9 Implementation of options for modifying resilience

In this last step selected mitigation options (step 8) are implemented and monitored. In this project, the decision support of step 8 will be communicated to the Crisis Alerting System being part of the Airport Operation Centre (AOC) which further distributes the information to local authorities.

Finally, all steps are summarized in the context of the SATIE project. In Table 2.2 an overview is provided where the respective information for the resilience management cycle is collected and how it is used.

Table 2.1: Summary of all steps in the resilience management cycle in the context of SATIE.

Resilience management step		SATIE task	Input by end users	Tools
1	Context analysis	Task 2.2 DoA	Scenarios	
2	System analysis	Task 2.2 Task 2.4	Asset lists Network information	Excel template
3	System performance function identification	Task 2.4	KPIs	Excel template
4	Disruption identification	Task 2.2 Task 5.3	Scenarios and threat on asset list	Incident Management Portal
5	Pre-assessment of the criticality of combinations of system functions and disruptions	Task 2.2 Task 2.4	Impact of threat on asset	
6	Overall resilience quantification	Task 5.1		CaESAR + ABM
7	Resilience evaluation	Task 5.1	Uncertainties	
8	Selection of options for improving resilience	Task 5.1	Mitigation strategies	CaESAR + ABM
9	Development and implementation of options for improving resilience	Task 5.4		Crisis Alerting System

3 Preparation of the Impact Propagation Model

3.1 Review on modelling approaches

By using models to represent a complex system, one can easier achieve a better understanding of the system of interest. The models can be used to get a structural overview built on a rule-based representation. Additionally, it provides the opportunity to easier describe an incident or situation, or predict future behaviour in the system. All the approaches have a challenge in the way that a model is a manageable simplification of real complex systems, which can lead to deviation from the real system behaviour. Different modelling approaches of the same system may differ in the outcome dependent on the level of complexity and the information available.

In the following sections, a literature review of modelling approaches is given. The final choice of modelling approaches for this project is discussed in chapter 4.

3.1.1 Conceptual models

The goal of conceptual models is to describe concepts or complex information in a simple way. Conceptual models should enhance the understanding of abstract systems in a systematic and structured manner. Further, the model should ensure efficient exchange of system details among organizations, and follow rules and standards for better comparability.

OSI model

The Open System Interconnection (OSI) model is developed especially for telecommunication and computer network, and describes the functions of these two kinds of networks. The aim of the model is to define standard protocols to enable interoperability of communication systems, which are internally based on different methods and techniques. Seven abstract hierarchical layers, consisting of different functions to execute, represent the communication in the model. These mentioned layers are: application, presentation, session, transport, network, data link and physical. The interchange or relationship between entities within one layer is defined as standardized protocols.

SysML

The System Modelling Language (SysML) is a general-purpose modelling language directed towards system engineers. SysML supports analysis, specification, design, verification and validation, and can be applied to several types of complex systems. The main purpose of SysML is to specify and architect systems, and it is capable of modelling structures, components and behaviour of systems.

SysML is an extension of the Unified Modelling Language (UML), providing additional language features which are aimed toward system engineering tasks. SysML only uses the part of UML which is relevant for system engineering. Due to this, SysML is more flexible and expressive and has more advantages for the use in this field than UML.

3.1.2 Probabilistic models

Probabilistic models help understanding risk and uncertainties in the decision making of a system. These types of models are based on statistical probability distributions and take uncertainties for different types of scenarios into account (Solomon & Sharpe, 2016).

Bayesian network

Bayesian network modelling describes the dependencies between independent, random variables and is widely used in the field of reliability engineering. The network is usually visualized in a directed acyclic graph (DAG), where nodes represent the variables in the graph. The directed connection between the nodes describes how dependent one node is to another node. To evaluate the probability of failure in the network, failure events can be binary, either true of false depending on the occurrence of failure. The Bayesian network model can be used to find the contributing factors for a disruptive failure event in infrastructures (Eldosouky, Saad, & Mandayam, 2017).

Monte Carlo

Random objects or processes can occur in a model of a real-world system, and the Monte Carlo method can account for these uncertainties. The method can be applied to solve quantitative problems in a wide range of fields; some of the present main areas are science, engineering, and finance. This broad applicability is an advantage, as well as the simplicity of the method (Kroese, Brereton, Tamire, & Botev, 2014). As the Monte Carlo method considers uncertainties, it can be used in risk assessments. One example is (Stroeve, Blom, & Bakker, 2009), where a Monte Carlo simulation is used in accident assessment regarding air traffic.

3.1.3 Network and graph models

Network models can be used when assessing vulnerabilities in infrastructures by identifying critical components and by simplifying the infrastructure network. Network models describe objects and the relationship among them with nodes or vertices that represent the components in a system, and edges or links that represent the connections between the nodes. The set of nodes and edges defines the topology of the network.

Topological models

Topological models can be used to evaluate weaknesses in a network, but have a drawback in evaluating the functional relationship among the elements, which is important when assessing the performance in a network (Hasan & Foliente, 2015). Topological models make it possible to model an infrastructure with a high level of abstractness; this degree of abstractness depends on what should be analysed. When using a highly abstract model, the possibility for performance and computation optimization increases and consequently faster results are expected. It is also feasible to model parts of the system more abstract than other parts. By only using a topological model, the information needed for a risk and resilience analysis is not sufficient. The model must therefore be used in combination with other models for this type of research.

Flow-based models

In contrast to the topological models, a flow-based model can evaluate the flow characteristic through a network, e.g. electricity in a power network. The model is also applicable to interconnected networks. Each node and each edge in the network can produce, load and deliver flow (Ouyang, 2014). Based on this, the performance of the network can be more precisely assessed, and the model can be used as an extension of the topological model. One disadvantage of the flow-based model is a very high computational effort, due to the complexity of the model (Hasan & Foliente, 2015).

3.1.4 Agent-based models

Agent-based modelling (ABM) can be employed to analyse system's emergent behaviour from a bottom-up perspective (Dam, Nikolic, & Lukszo, 2013). Independent agents interact in a specified environment and the resulting system's states can be observed. Further, ABM can also analyse how the system and agents need to be designed to achieve pre-defined system states.

In contrast, a multi-agent system (MAS) is a self-organized system, used for solving complex problems by splitting them to smaller tasks. The agents in MAS are components of a software or algorithm performing an action independently without any support from a central unit (Wooldridge, 2009). These agents solve problems by using e.g. previous actions, input information, interaction with its neighbour and its goal. MAS is widely applicable and used for modelling, e.g. complex systems, smart grids and computer networks (Dorri, Kanhere, & Jurdak, 2016). One of the advantages of MAS is the possibility to speed up the problem-solving process, due to the opportunity of independent and parallel computing by the single agents.

3.1.5 Layer models

One single perspective on a complex system may be limiting. Therefore, a model can be divided into multiple layers, where each layer is a representation of a different group or phenomena in the same system (Oliva, Panieri, & Setola, 2012). Each layer has a specific function within the model. Usage of layers in the model result in multiple dimensions of connectivity, also named aspects. How the layers are coupled together will change the structural properties of the model. A model with multiple layers allows for more accurate representation of coupled structures and processes within one system, which result in a more extensive and detailed study (Domenico, Granell, Porter, & Arenas, 2016).

Infrastructure systems are organized as networks in the way that they include nodes, edges and paths. In infrastructure networks various types of flows pass through the network, e.g. trains, vehicles or airplanes in a transportation infrastructure. These systems can be looked at as coupled layers which can share physical networks, environmental concerns, information and functional interdependencies. It is important to understand and consider these different interactions between the layers, as it will give a more thorough representation. Based on this, layers models are often used when modelling an infrastructure today (Zhang, Peeta, & Friesz, 2005).

3.2 Graph analysis as performance measure

Performance measures for a resilience analysis can be derived in different ways (see section 2.3). In the following sections, a literature review of graph-based performance measures is presented. These measures help to analyse the network situation and functionality during varying conditions such as normal and disturbed operation having only the graph structure available. The implementation and final choice of employed performance measures in general will be done in Task 5.1.

3.2.1 Number of connected nodes

For an undirected graph to be connected it needs to consist of at least one node and a path between every pair of nodes. In other words, a node in a connected graph is reachable from any other node by following a path in either direction. By only having one node, the graph is still connected, but if there are a pair of nodes and no path between these, it is disconnected.

3.2.2 Shortest path

Distance measures, including shortest path, eccentricity, diameter, radius, centre and periphery relate to the broadness of a graph structure. The shortest path intends to identify the path from a source node to a destination node associated with minimum weight. These weights can be distance, time or cost which needs to be optimized in a feasible way. One challenge is that in many networks, there can be lack of information about these weights, or uncertainties. There are many different algorithms developed which can be used to compute the shortest path between nodes, one of the most common is Dijkstra's algorithm (Broumi, Bakal, Talea, Smarandache, & Vladreanu, 2016). However, the algorithm has a drawback as it requires a large computational effort.

3.2.3 Centrality measures

Centrality measures are used to identify the most important nodes in a graph. A node with high centrality has a larger impact on the other nodes in the network than a node with low centrality. This type of measure can be used in infrastructural networks to identify the most critical and vulnerable nodes (Demsar, Spatenkova, & Virrantaus, 2008). There are different types of centrality measures, and some of these types are described in the following. Degree centrality derives the number of connections one node has to other nodes. This is helpful to understand to which degree a node can influence the other nodes. Closeness centrality focus on the distance between one node to all the other nodes in the graph, which indicates the node's influence on the whole network. The last centrality measure is betweenness centrality, which emphasizes on how many shortest paths that goes through a node (Yan & Ding, 2009).

Betweenness Centrality (BC) is a fundamental metric of centrality already applied in transportation sector for the identification of topological criticalities. In general, it is preferred to other centrality metrics such as degree and closeness centrality. In addition, BC has also been used for the analysis and predictions of traffic flows and vulnerability assessment in transportation networks (Furno, Faouzi, Sharma, Cammarota, & Zimeo, 2018).

3.2.4 Similarity measures

Graphs are assumed similar if a node in one graph has the same respective neighbourhood as a node in another graph. One way to measure this is in an iterative process: For each time step the similarity score between two nodes is calculated and then the propagation continues along neighbouring nodes (Zager & Verghese, 2008). Another commonly used method is the "Graph Edit Distance". This method is based on how many operations are needed in order to transform a graph into being identical to another graph (Gao, Xiao, Tao, & Li, 2010).

3.2.5 Spanning tree

A spanning tree is a connected and undirected subgraph that includes all the nodes in the graph, with a minimum number of possible paths. The spanning tree has no loops, but it is still possible to go from one node to another node without repeating a path. There exist no spanning trees if the graph is disconnected, and the number of spanning trees increases with more paths between the nodes (Standish, 2008). In a graph with weighted edges between the nodes one can use the minimum spanning tree to find the spanning tree with the minimum total edge weight. There exist many types of different algorithms to execute this computation (Mamun & Rajasekaran, 2016).

In a graph with weighted edges, one can use the minimum spanning tree to find the spanning tree with the lowest total edge weight among all the spanning trees. Total edge weight is the sum of all the weighted edges in the spanning tree. There exist many types of different algorithms to execute this computation. Examples of algorithms that calculate the minimum spanning tree is Boruvka's algorithm and Jarnik-Prim algorithm (Mamun & Rajasekaran, 2016). It is also possible to compute the total number of all possible spanning trees in a graph (Chakraborty, Chowdhury, Chakraborty, Mehera, & Pal).

3.2.6 Clustering coefficient

The clustering coefficient represents how nodes in a graph tend to cluster together. This is computed by the number of closed triangles in the graph, which can be used to indicate the small-world property. The clustering coefficient can be calculated for various types of networks. There exist two types of clustering coefficients, the local and the global. The local coefficient is computed for a single node, and the global for the overall graph (Green & Bader, 2013).

3.2.7 Connectivity

Connectivity is an indication of how strong the connections in the network are. This is normally measured by the number of nodes that need to be removed before the graph is disconnected. Two classical approaches to measure the reliability of the graph are the vertex connectivity or the edge connectivity (Hellwig & Volkmann, 2008).

3.2.8 Degree distribution

The variation in the number of edges connected to a node is represented by a distribution function. This function describes the probability of exactly how many edges that are connected to a randomly chosen node (Holmgren, 2006). One of the most common indicators of network resilience is the variation on the fraction of nodes in the largest connected component upon link removals. Relevant studies on the Internet autonomous system (AS) topology reveal that networks with power-law degree distributions are relatively robust to a random failure. However, this type of networks is vulnerable to attacks against the central nodes, which are likely to cause the network's fragmentation (Hernandez & Mieghem, 2015).

3.2.9 Assortative mixing

Assortative mixing in the graph means that nodes with many connections tend to attach to other nodes which also have many connections. On the other hand, a graph shows disassortative mixing when nodes with many connections attach to nodes with few connections (Newman, 2002). Assortative mixing also applies in the field network resilience. As it has been discussed the connectivity of many networks can be destroyed by the removal of just a few of the central nodes (highest degree). In mixed assortative networks it seems that removing high-degree vertices is a relatively inefficient strategy for destroying network connectivity, whereas the attacks on the highest degree vertices of a disassortative graph are much more effective.

3.3 Review on threat propagation and mission impact assessment methods

Current approaches in the realm of cyber-mission impact assessment (MIA) (S. Noel, 2016) (A. de Barros Barreto, 2014) (Temin, A Cyber Mission Impact assessment tool, 2015) (J. R. Goodall, 2019) are commonly built to help organizations to pinpoint the amount of damage a cyber-threat can cause on the organization's assets, operational objectives and goals (i.e., mission). Common approach to identify damage caused on the assets and mission is to employ threat propagation analysis methods that study the relationships between *IT systems* of an organization, to identify how connections between assets can contribute for attackers to propagate their attack to other assets of the organization. Knowing the assets compromised through threat propagation, MIA methods measure the actual *impact* of the attacks to the services and goals of the organization based on the criticality of IT systems analysed. To do so, most of MIA techniques gather knowledge from different cyber-security resources, such as: reconnaissance tools for IT system discovery; vulnerability scanners and/or public vulnerability databases for vulnerability affecting organizations; intrusion detection systems (IDS) and Security information and event management (SIEM) systems for information about attacks; and impact identification methods that identify services and goals affected by those attacks.

The remainder of this section presents an overview on the current available mechanisms to provide the necessary information for MIA. This section is structured as follows: Section 3.3.1, presents an overview of the current reconnaissance methods for identifying IT systems exposed to attackers; Section 3.3.2 presents an overview of the current methods to gather information about threats; Section 3.3.3 presents an overview of current methods to gather information about cyber-attacks happening in IT infrastructure of an organization using intrusion detection systems; Section 3.3.4

presents the current impact assessment methods available to estimate the damage caused by cyber-attacks on the organization IT infrastructure, and pinpoint the vulnerabilities that allowed the attack to be successfully performed.

3.3.1 Overview of the current reconnaissance gathering methods

The first step to perform cyber-threat analysis is to gain knowledge about the IT systems that may play a role in the execution of a cyber-attack. In this step, system administrators identify the critical IT devices they want to protect and the IT infrastructure that can be used by the attacker to compromise those assets.

The most common approach to identify the critical IT devices of a system is to map the services and goals of the organization (also referred as mission of the organization) with the IT devices used to accomplish those services. The level of criticality of the IT device can be classified according to: the amount of work performed in the mission; or on how dependent the mission accomplishment is on the asset (level of operability if the asset is compromised). There are several tools to aid system operators to classify critical assets: COTS (Commercial of the Shelf) software that allows operators to manually identify critical assets (such as Microsoft Threat Modelling Tool (What's New with Microsoft Threat Modeling Tool 2016, 2015)), or machine learning methods that infer criticality of the asset based on the mission processes the asset is involved (Aalst, 2011) (Rinderle-Ma, 2016).

To identify the attack surface on the IT infrastructure, it is crucial to pinpoint the main entry points in the IT network (i.e., assets on the IT infrastructure likely for attacks to initiate) and the overall topology of the network (to gain insight on how attacks can propagate from entry points to other assets). There are several tools that identify the attack surface: COTS software that allows operators to manually identify entry points and network topology (such as Microsoft Threat Modelling Tool (What's New with Microsoft Threat Modeling Tool 2016, 2015)), or automated reconnaissance methods (such as: NMAP (Wolfgang, 2002), Moloch (Moloch, 2019), Silk (SiLK, 2019)) that monitor network traffic and perform asset identification and network topology inference.

3.3.2 Overview of the current cybersecurity threat analysis methods

The main objective of threat gathering methods is to identify and classify the security flaws present in the organization. To do so, common threat gathering methods are typically divided into two phases: 1) threat identification, where IT devices (inner hardware peripherals, operating system, and software) and network communications (involving those devices) are subject to security tests/analysis in order to identify vulnerabilities; 2) vulnerability classification, which happens after the identification phase, and involves cataloguing vulnerabilities according to hindering effect they may on the systems based on standard vulnerability classification methodologies.

Current methods to identify and catalogue vulnerabilities in organizations can be divided into two categories: product-based vulnerabilities and organization-based vulnerability identification methods.

- The first category, product-based threat scanning is typically performed to COTS products (hardware devices, operating systems or software applications) and is normally employed by vendors of the products or independent security consultants. In such methods, product families are subjected to security tests (employed by product-vendors or in most cases third parties) to identify possible security threats related with bad implementation of products or malfunctions affecting one or more versions of the product. The results of these security tests are then usually communicated to the product consumers by placing them in public databases (e.g., CVE (CVE - Common Vulnerabilities and Exposures (CVE), 2019), NVD (NVD - Vulnerabilities, 2019)). Organizations then perform an inventory of the threats affecting their IT infrastructure by collecting information from product-vendors, either by direct accessing the public databases or by resorting to threat scanners (e.g., NESSUS (Download Nessus Vulnerability Assessment | Tenable®,

2019), Nikto (Nikto2 | CIRT.net, 2019), Zap (ZAP, 2019), Archery (Tiwari, 2018), OpenVAS (OpenVAS - OpenVAS - Open Vulnerability Assessment Scanner, 2019)) that identify threat present in the products of an organization based on the publicly available threat databases.

- The second category, organization-based threat identification methods, are typically performed within the organizations IT infrastructure and are centred on identifying security threats that appear due to the operational deployment of those products, namely bad configuration of products and/or deployment without assuring the necessary conditions to provide proper secure execution. Typically, these methods are performed manually by system administrators of the organization or by external contractors, by following threat modelling methodologies (such as STRIDE) and managed by the organization security team either by storing it on excel sheets or by resorting to risk assessment tools such as OWASP Threat Dragon (OWASP Threat Dragon - OWASP, 2019), DefectDOJO (DefectDojo | CI/CD and DevSecOps Automation, 2019), ThreadFix (ThreadFix Vulnerability Management Platform, 2019).

3.3.3 Overview of the current intrusion detection systems

Intrusion detection systems (IDS) are an important source of information used in the context of security risk management (Joint Task Force Transformation Initiative, 2012) for monitoring the security level of an IT infrastructure. They help detecting potential active threats and the assets directly hit by attacks. The rationale behind these technologies is to detect incidents based on data collected from sensors that monitor IT devices (host-based sensors such as Ossec (R. Bray, 2008)); network (network-based sensors such as Snort (Roesch, 1999), BroIDS (V. Paxson, 2006)); or both (hybrid IDS such as Trend Micro (Enterprise Intrusion Prevention (IPS) Software & Solutions, 2019), BP-IDS (INOV / BP-IDS – Business Process Intrusion Detection System, 2019)). Several methods can be used for detection: signature-based IDS where collected data is compared to patterns of known attacks; anomaly-based IDS compare data with the normal behaviour (baseline) of the system taught beforehand during an initialization period; specification-based IDS which compare data collected with a set of specification rules stipulating the acceptable behaviour of the system.

Despite being widely commercialized products with several COTS solution providers available (2018 Gartner Magic Quadrant for Intrusion Detection and Prevention Systems (IDPS) | Alert Logic, 2019) intrusion detection is still an important topic in academic research. Proposals for novel schemes that improve anomaly-based IDS mechanisms with new fingerprinting techniques for detecting stealthy attacks are occurring in cyber-physical systems (CPS). This class of attacks occur between IoT sensor and actuators and avoid IDS detection of physical attacks, by changing sensor measurements with false data injections. It is typically assumed that the attacker has enough knowledge of sensor and processes involved when such attacks take place, such as: detection of these new classes of attacks in vehicle intrusion attacks (Shin, 2016) (al., Detecting Attacks Against Robotic Vehicles: A Control Invariant Approach, 2018) (Huth, 2018); water control systems attacks (Urbina, 2016) (C. M. Ahmed, 2018) (Y. Chen, 2018) (W. Aoudi, 2018); and SCADA attacks (Y. Chen, 2018) (D. Formby, 2016) (L. Cheng, 2017). Several of these new approaches distinguish themselves based on the type of fingerprints used for profiling the normal behaviour, which can be: (a) physical fingerprinting, by measuring time taken for physical component (actuator) to perform an operation (D. Formby, 2016), or characteristics of the physical sensor measurements (Urbina, 2016) (C. M. Ahmed, 2018) (W. Aoudi, 2018) (e.g., water level, Pre-treatment the Conductivity, and Oxidation-Reduction Potential), with some of these proposals incorporating noise handling mechanisms during physical fingerprinting to optimize results (C. M. Ahmed, 2018) (W. Aoudi, 2018);(b) fingerprints based on metrics in the communication between master-slave components (Shin, 2016) (Huth, 2018) (D. Formby, 2016); (c) fingerprints based on code analysis of the software used in control units to determine normal and abnormal behaviours (al., Detecting Attacks Against Robotic Vehicles: A Control Invariant Approach, 2018) (Y. Chen, 2018) (L. Cheng, 2017). An alternative method to using anomaly-detection for detecting stealthy physical attacks has been demonstrated in the ECOSSIAN project. The proposed system used Business Process

Intrusion Detection System (BP-IDS) (INOV / BP-IDS – Business Process Intrusion Detection System, 2019) a specification-based IDS that collected data from multiple sources of sensors installed in cyber-physical networks. BP-IDS was tested on railway and gas distribution, and identified incidents by comparing information collected with the organization's business processes and rules that stipulate normal behaviour. The results of this project have shown that BP-IDS is very capable of detecting incidents caused by man in the middle (MiTM) attacks in SCADA system that communicate through IP, IEC 60870-5, and Modbus network protocols.

Intrusion detection allows the identification of attacks and reacts promptly with the security measurements to minimize the impact of the attack. Although knowing the time and type of attack taking place is essential for the success of the contingency plans, other information needs to be evaluated regarding the impact of the attack to properly plan the reaction, namely: business assets affected and business goals compromised by the attack. Within this generic scope, impact assessment procedures used by organizations evaluate the damage caused by a cyber-threat to its assets and goals.

3.3.4 Overview of the current threat propagation and impact assessment identification methods

Mission impact assessment (MIA) (A. Kott, 2017) over cyber-attacks often includes methods to evaluate whenever an attack happens, the amount of damage caused to the organization's assets, and which operational objectives and goals (i.e., mission) were compromised by those assets.

Cyber-attack impact assessment is largely influenced by approaches such as M-Correlator (P. A. Porras, 2002) which prioritizes and clusters incident alerts published by different IDS solutions installed in organizations, according to the impact the incident has on the assets of the organization. M-Correlator uses Bayesian networks (N. Friedman, 1997), to classify the impact of threats present on a given network topology based on the incident alerts reported by intrusion detect systems. VTAC (Yang, 2008) used the foundation of this work and extended it by automating the vulnerability identification and ranking with Calderon approach (S. Jajodia, 2011) of using vulnerability scanners to provide that information, and evaluated the overall impact of the organization after an incident (or a series of incidents) caused by cyber-attacks.

Even though both M-Correlator and VTAC evaluate the impact a cyber-attack had on the organization assets, neither method is capable of exactly pinpoint the amount of damage caused to the mission's objectives/goals. (Jakobson, 2011) augmented VTAC to allow this estimation by mapping mission information (tasks) with the assets and network topology of the organization using dependency graphs (Balmas, 2004) to calculate the operational capacity of each mission after an incident (or a series of incidents) caused by cyber-attacks. Further research to improve cyber-attack impact assessment (C. Liu, 2017) (C. Ten, Vulnerability Assessment of Cybersecurity for SCADA Systems Using Attack Trees, 2007) (A. Kott, 2017) adapted this work to be compatible with SCADA environments using vulnerability trees, PetriNets or SysML instead of dependency graphs; (A. de Barros Barreto, 2014) (A. Kott, 2017) proposed using BPMN (Business Process Modelling Notation) for modelling high level mission information which was put in to motion in (C. Liu, 2017) by implementing a tool that provides conversion of BPMN to dependency graphs and integrates with the augmented VTAC; and (A. Motzek, 2015) (Y. Yang, 2018) introduce the notion of temporal dependency to improve impact calculation.

3.4 Analysis of the impact of possible threats on airport systems

In this section, two specific example systems, namely the baggage handling system and air traffic management are described in more detail with a focus on impacts of potential threats. These systems have been selected exemplarily to better understand the level of detail that is needed to best model the airport's systems. Not all potentially relevant systems could be addressed with the same level of detail as the required level for modelling and simulation still needs to be defined in Task 5.1.

3.4.1 Architecture of a baggage handling system, exemplarity for ZAG airport

In general, the baggage handling systems include:

- Outbound (departing) baggage systems
  - Baggage conveyors from passenger check-in facilities
  - Transfer baggage input conveyors to transport bags from one flight to another
  - Security screening areas
  - Conveyors that sort baggage to airlines or flights within an airline
  - Baggage equipment (conveyors and carousels) for loading baggage onto an aircraft
- Inbound (arriving) baggage systems
  - Baggage input conveyors delivering baggage to the baggage claim area
  - Baggage transport conveyors
  - Baggage carousels

The processes and different levels of baggage screening are shown in Figure 3.1.

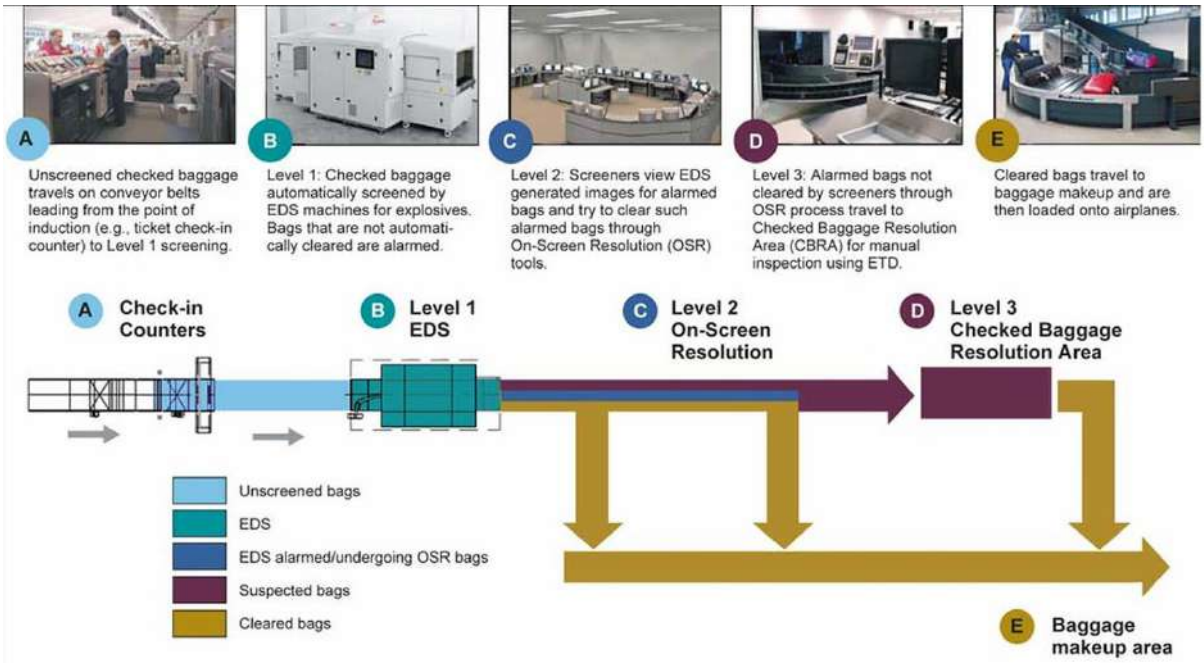

Figure 3.1: Schematic of the BHS at ZAG airport with processes described from check-in to carousel showing different levels of baggage screening.

3.4.2 Analysis of impact on Baggage Handling System of a cyber-attack

A weak protection can result in both data leakage and disruption of operations. Some generic examples for possible cyber-attacks on the BHS are given below.

1. General Data Protection Regulation (GDPR) compliancy

By hacking the link between sort allocation computer software and other baggage management solutions a data leakage can be caused.

2. EDS machine security

Any attack of an EDS machine causes an interruption of service and stops the BHS operation.

3. Network security

The PLC network has evolved and now all the components are not linked anymore in ASI network but now in Ethernet network and that increases the risk of attacks. The consequence of the evolvement of Ethernet network is that a malicious person can apply different cyber-attacks to disturb the BHS: A Denial of Service (DOS) Attack through Ethernet port, a Man in The Middle (MITM) Attack through Ethernet port, etc.

3.4.3 Analysis of impacts on air traffic management

ATM operations have basically two groups of actors: (1) Air Traffic Controllers (ATCO) for controlling and monitoring the air traffic and (2) the cockpit crews/pushback vehicle drivers, who move aircraft by performing the advisories from the ATCOs. Beside those ATCO controlled movements on dedicated taxiways and runways, there a several service roads or areas, where vehicles like e.g. baggage carts or catering vehicles move. Those vehicles are not controlled by any ATCO, they are not allowed to enter taxiways or runways and they have to give priority to aircraft. However, if a vehicle needs to enter a taxiway or runway (e.g. for runway inspection) it has to contact the ATCO, report the intentions and wait for commands.

Having this in mind, there are four options to attack ATM operations:

1. Disturbing or eliminating ATCOs

By manipulating or spoofing data, which is used by ATCO for guidance of vehicles on ground or in the air, a serious incident in ATM can be created. The magnitude of the incident can vary between disruptions in the process with possible significant financial impacts up to accidents with in worst case lethal consequences. This can even be aggravated by either physical destruction of ATC facilities on the ground or an intrusion in an ATC centre or tower. An example for this kind of attack is the malicious arson of the air traffic control centre in Chicago in 2014, which led to the cancellation of more than 1500 flights (CBS News, 2014).

2. Disturbing or eliminating cockpit crews

Like for the ATCOs it is also possible to create serious incidents by manipulating or spoofing data for cockpit crews. Also, a physical attack of the crew, e.g. hijacking of one or more aircraft at the same time is a critical hit of the ATM system. The most terrible incident in this regard happened on the 11th of September 2001 in U.S. airspace, when four airplanes were hijacked and used as weapon against civil population, resulting in several thousand victims (History, 2019).

3. Disrupting communication between ATCO and crew

Today's air traffic control is based on the communication between the ATCO, who is responsible for safe and efficient traffic flow in his sector or area, and the vehicle crew operating the aircraft or ground vehicle according to the advisories of the controller. If this communication line is interrupted or spoofed, serious incidents can occur, which might result in a significant disruption of the air traffic (e.g. delays, cancellations) or even in lethal accidents. Several incidents for this course of action are known (LTBA-ISTANBUL, 2019) (The Telegraph, 2016).

4. Physical attack from outside

Attacking air traffic from outside can take place at ground as well as in the air with various means. These attacks are not limited to free accessible areas. Also, restricted areas like the airport ground can be affected, either by attacking from the outside like flying a drone in critical areas or by breaching the airport fences. An example for the latter one is the attack at Hanover airport in 2018, where a car entered the tarmac. Subsequently all airside operations were shut down for several hours (DW, 2018).

All these four attack options can result from various threats, which have already been developed within this project. In a next step the various threats are matched to one or more of these four options to determine their impact on air traffic management operations.

4 Impact Propagation Model

In this chapter the approaches are presented that are used to build the Impact Propagation Model. The latter build the basis for the implementation of the Impact Propagation Simulation (T5.1).

4.1 Modelling approach

To represent airport systems and processes, a hybrid model is developed that consist of (1) a layer model, (2) a network model and (3) an agent-based model. In the following sections the modelling approaches for the Impact Propagation Simulation are presented.

The choice of modelling approaches is based on the review provided in Section 3.1. The layer model is suitable to describe the airport processes because they consist of different flow layers. The representation of the airport through a network model is chosen (i) to be compliant with the requirements to use the CaESAR tool and (ii) to use the already existing asset lists and enrich this information with the interrelations between the assets. The connections between the assets cannot be exhaustively described only by the network model as in socio-technical infrastructure such as an airport the human component plays an important role. To this end, the agent-based model is employed which describes in case of equipment failure/attack the response of the passengers and employees and also the influence of people on airport assets (e.g. blocking of entrances, occupying security personal, etc.). These approaches are combined into one simulation environment and in the following each approach is presented in more detail.

4.1.1 Layer model

As already introduced in section 2.2, a layered model is used to represent the airport and distinguished into four flow layers, namely aircraft flow, baggage flow, information flow and people flow. The latter can be further divided into passenger and employee flow. In Figure 4.1 the passenger, baggage, information and aircraft flow are visualized. This model is used to further categorize the assets to describe impact propagation in specific flow layers. It also links the network model including the categorized assets introduced in 4.1.2 and the ABM in 4.1.3. The assets comprised in the passenger flow layer are defined to impact the passenger flow which is represented in the ABM.

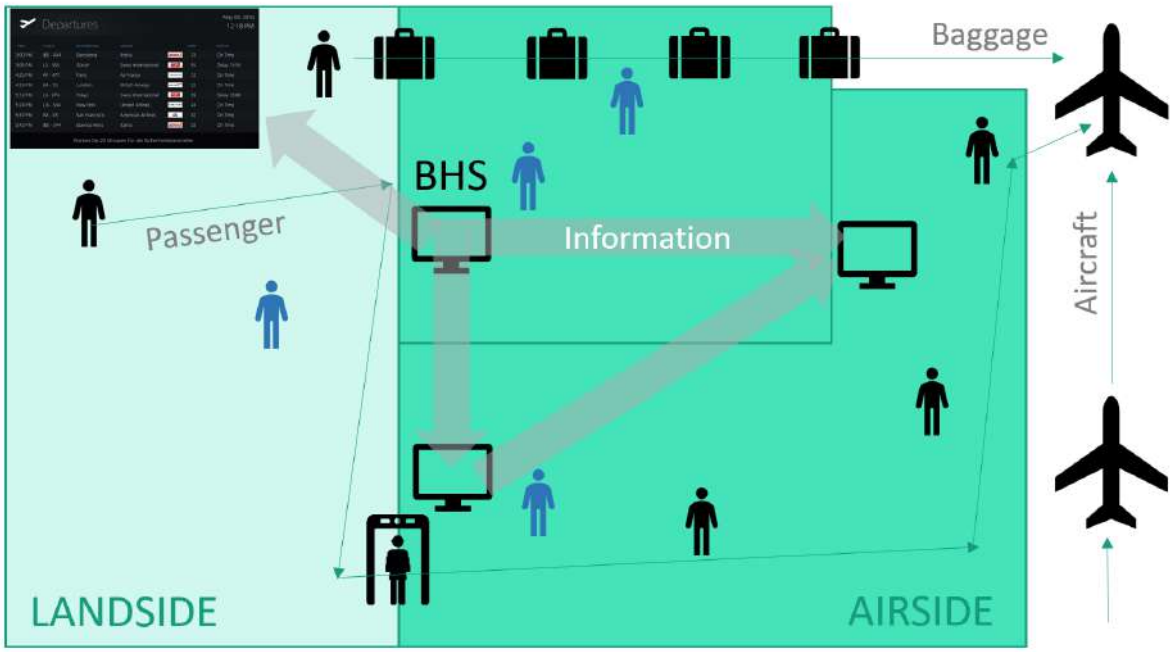

Figure 4.1: Airport schematic to visualize passenger, baggage, information and aircraft flow. Employees are given as blue persons. Information flow is shown exemplarily as data exchange between BHS, security check-points, gates and Flight Information Display System (FIDS) monitor.

4.1.2 Network model

For the SATIE Scenarios, network models have been developed which are presented in Figures 4.2 to 4.6. Each node in the network represents an asset and the interrelations between the assets are visualized through a directed graph. For security reasons, no detailed information about the assets can be provided here. The interrelations are defined in the following way: An asset that fails influences other assets and is thus connected to them. As a part of the pre-assessment (see section 2.5), the graph structures are analysed. The colour coding of the asset circles corresponds to the flow layer and the size of the asset circles corresponds to the number of incoming arrows which means the number of influencing factors. Alternatively, the number of outgoing arrows can be used as a weight for the node size to represent the importance for the system of a single node as an influencing factor.

Failures due to cyber-physical attacks will be propagated through these networks. Therefore, the properties of nodes and arcs are crucial for a realistic representation of the airport's operations. In the following, five generic example network structures are presented where nodes and arcs do not represent the airport networks which are used in the scenarios.

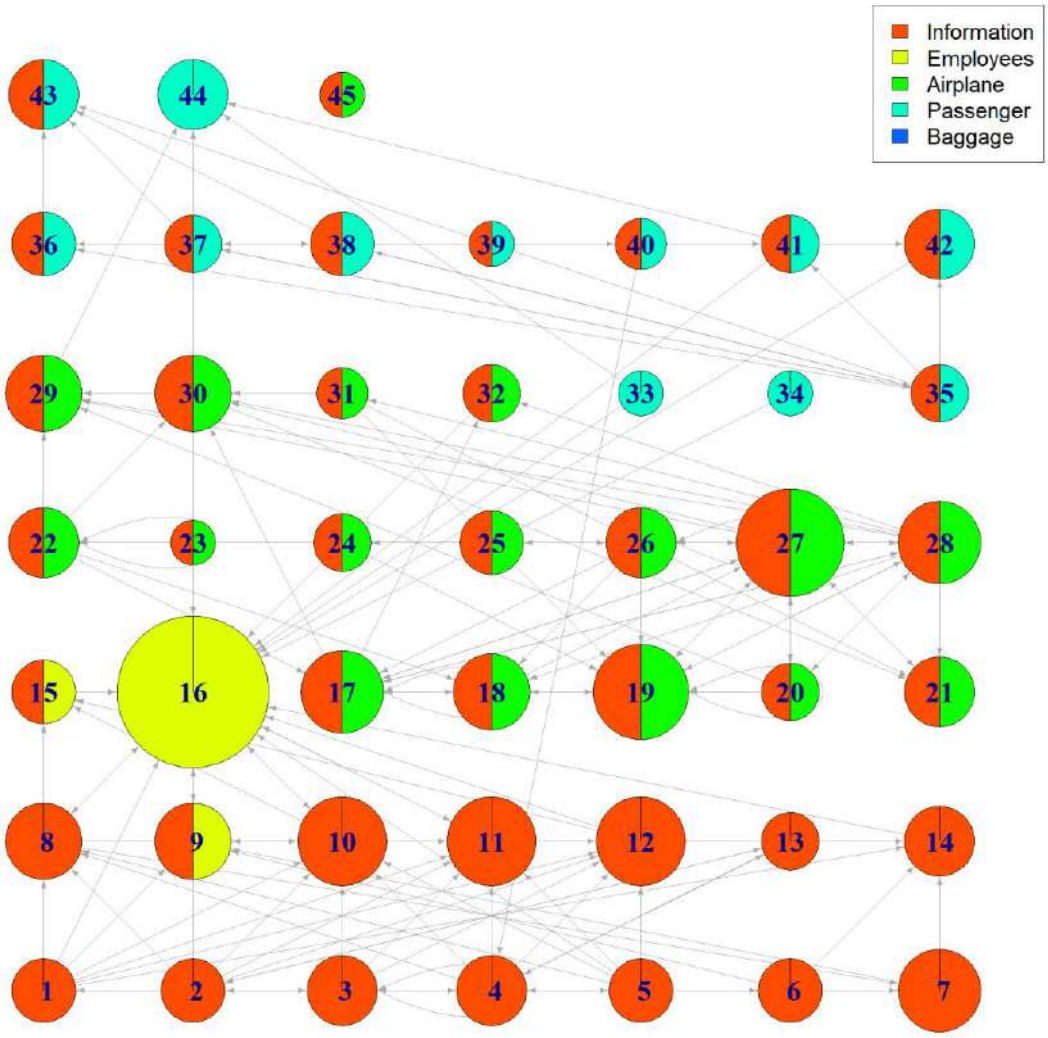

Figure 4.2: Example network structure. The colour code represents the flow layer; the size of the nodes (circles) is proportional to the number of incoming arcs.

The criticality of nodes can be defined by their importance in the network which could be e.g. as visualized in Figure 4.2 the number of influencing factors. Here, it can be concluded that employees play a critical role in the network and most IT components.

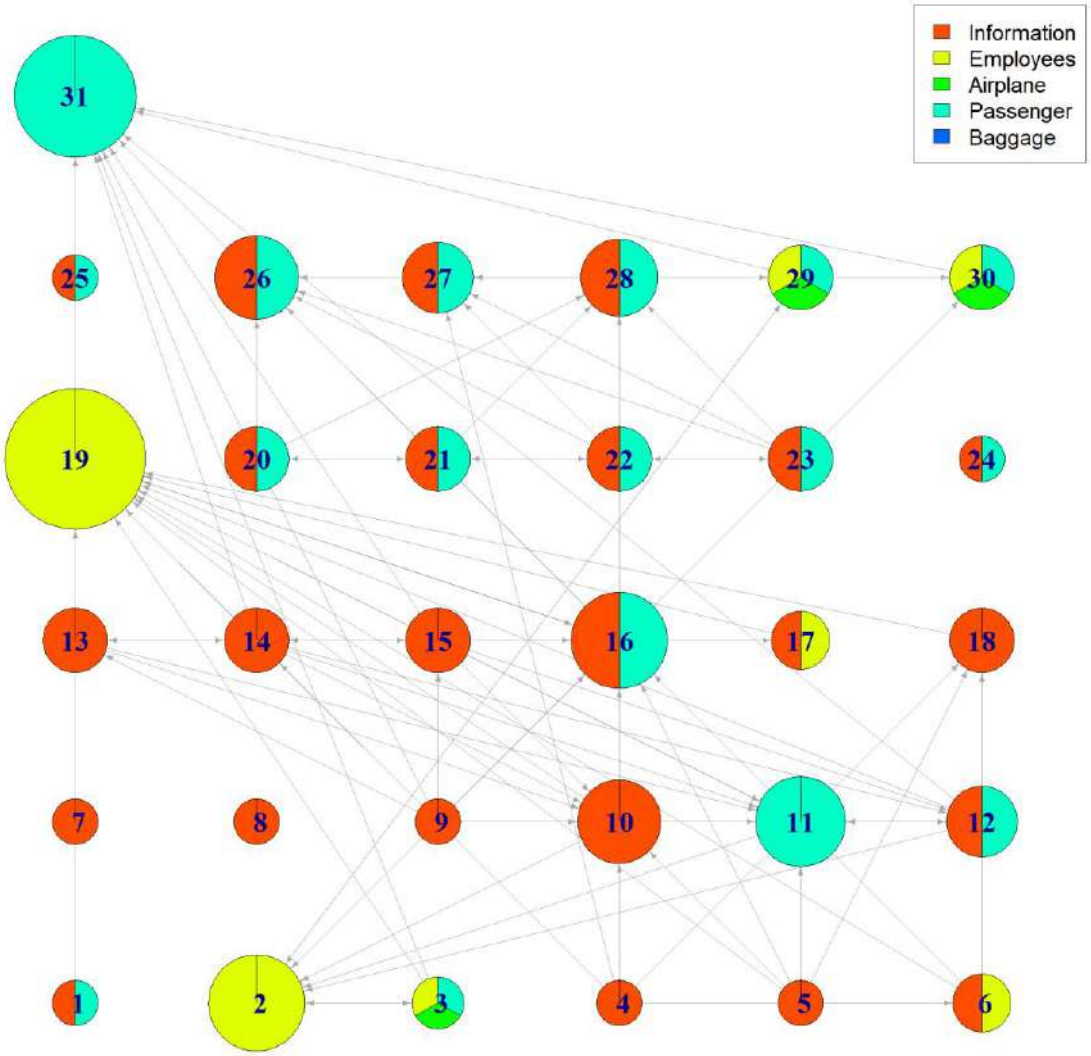

Figure 4.3: Example network structure. The colour code represents the flow layer; the size of the nodes (circles) is proportional to the number of incoming arcs.

Analysing the network presented in Figure 4.3, employees and passengers seem to be the most sensitive components. They are influenced by various other assets in case of failure or degradation of functionality.

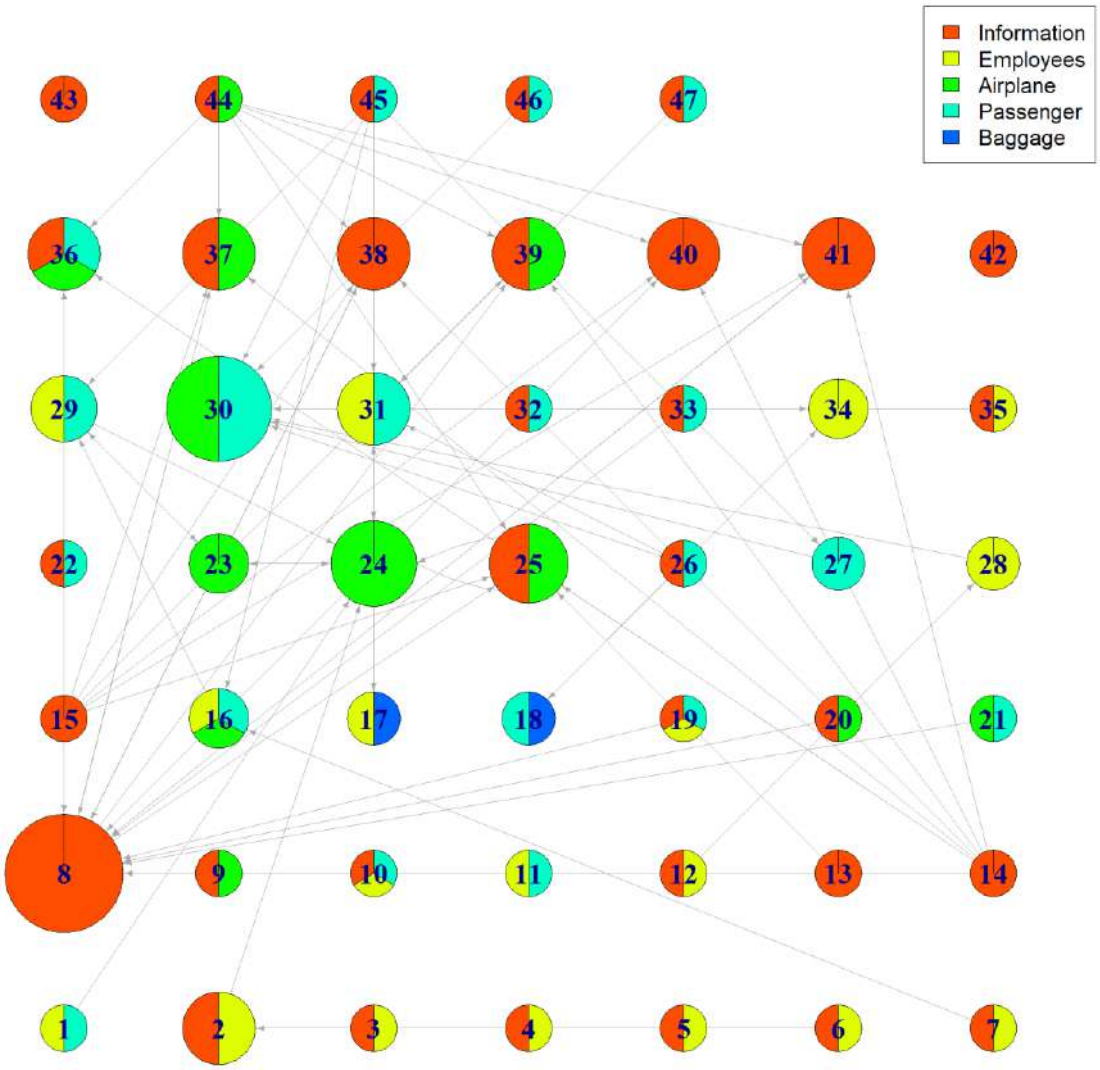

Figure 4.4: Example network structure. The colour code represents the flow layer; the size of the nodes (circles) is proportional to the number of incoming arcs.

In Figure 4.4, asset number 8 is affected by many other assets in case of failure or degradation. Asset number 8 corresponds to an airport data base. Also, the asset number 30 seems to be a sensitive component which corresponds to the sterile area after security checkpoints. Which means that several assets will influence the boarding gate area, if their functionality is degraded.

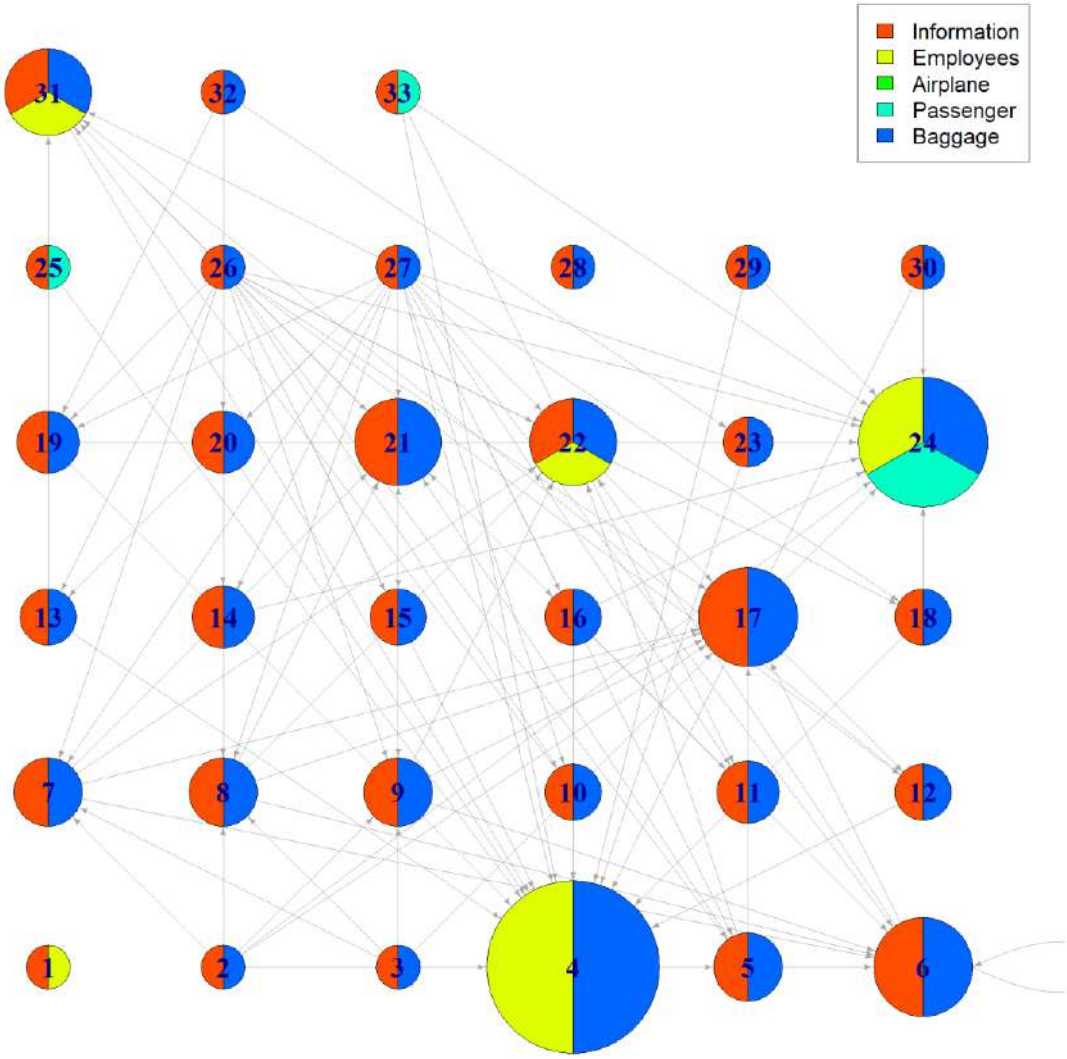

Figure 4.5: Example network structure. The colour code represents the flow layer; the size of the nodes (circles) is proportional to the number of incoming arcs.

In Figure 4.5, the most critical component seems to be asset number 4 which is the employee working in the airport. By any kind of failure of assets of the BHS the employees work seems to be affected. The other highly connected asset is the number 24 which is the baggage itself.

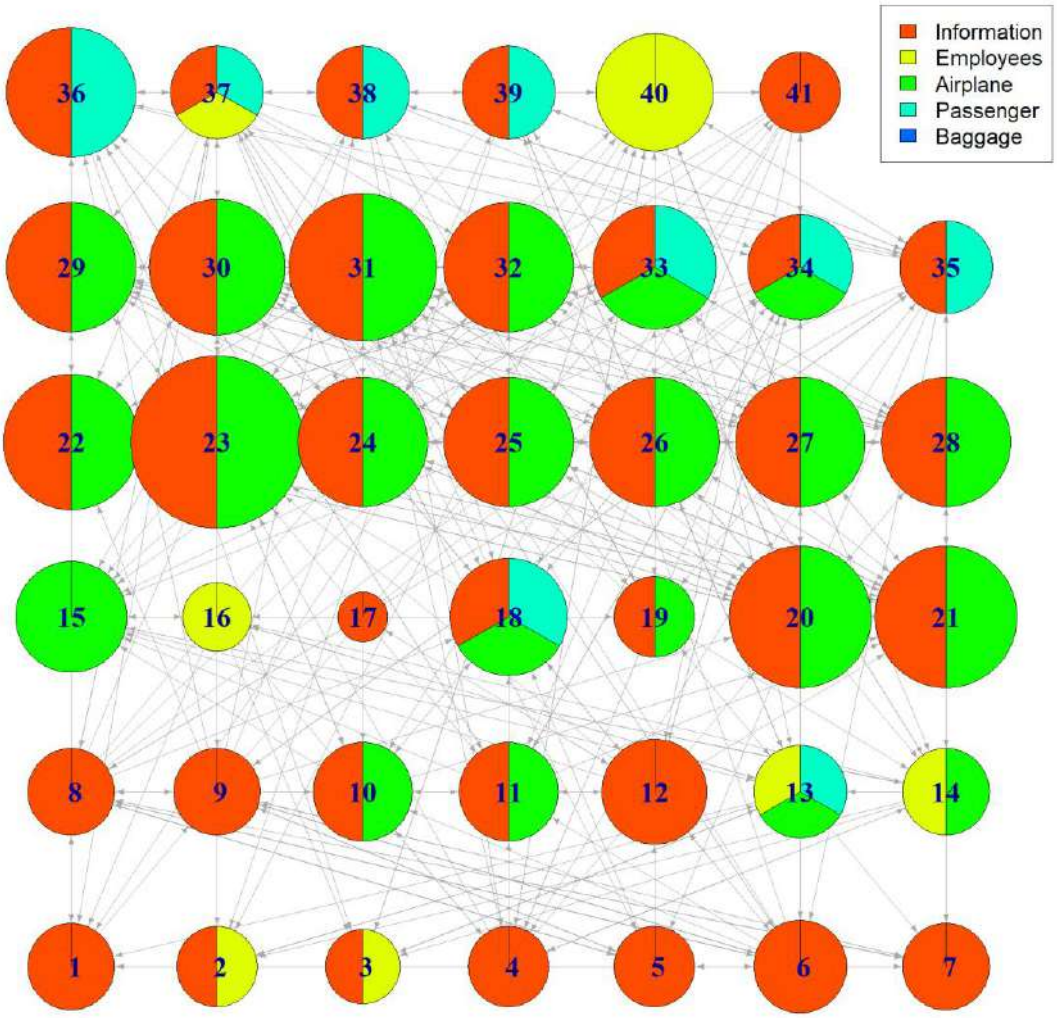

Figure 4.6: Example network structure. The colour code represents the flow layer; the size of the nodes (circles) is proportional to the number of incoming arcs.

In Figure 4.6, the interrelations between the assets are defined very detailed and thus the circle size reflects the overall large number of connections. The assets 20 to 32 are related to the FIDS which seems to be an important system in this scenario which is affected by several asset failures/disturbances.

Similar network models (Figures 4.2 to 4.6) will be combined into one network structure comprising all flow layers for implementation in the Impact Propagation Simulation to be able to predict impact propagation beyond system borders. The interrelations between assets are defined on a very generic level and need to be further qualified for implementation, e.g. by defining flow capacities. For the Impact Propagation Model, it is crucial to understand how assets affect each other; which will be represented through attributes for nodes and arcs in the network (see Section 4.1.4).

4.1.3 Agent-based model

In addition to the network model, an ABM is developed to represent passenger movement in the airport. The passenger flow (see e.g. Figure 4.1) is represented with independent agents as the individual behaviour of passengers and employees in the airport cannot be modelled realistically with a network structure. In Figures 4.7 and 4.8 a small example airport is presented with two entrances, four FIDS monitors, 18 check-in desks, five security lanes and three gates. The passengers move independently in the airport and decide due to their personal situation where they go first. In the context of SATIE, this model will be adjusted to represent the airport layout of the scenarios or a generalized airport model to analyse the impact of certain cyber-physical attacks on airport assets.

This ABM will be coupled with the network representation of the airport assets and failures due to an attack (i) propagate though the network and (ii) influence the passenger movement in the airport. The latter case is demonstrated in Figure 4.8 where a crowd forms due to a cyber-attack on the FIDS which displays the wrong information. In this case, it directs every passenger to one specific check-in counter. This small test scenario was developed to demonstrate the general idea behind the ABM approach used in this project and will be further extended to represent relevant scenarios considered in the project.

The agents follow specific rules whereby the most general rule is that once agents enter the air-side area they cannot return to the land-side area anymore. They move with a pre-defined individual velocity and have specific properties that guide their path in the airport. A list of passenger and airport properties implemented so far in the ABM can be found in Table 4.1. There are much more passenger properties that will be implemented in Task 5.1 such as personal needs, preferences and group behaviour. Specific rules have been implemented already how agents avoid collisions such that agents that are about to collide both rotate their direction of movement to the right by 45 degree and afterwards return to their original path. This was assumed to be representative for most cultures.

With a specific probability, agents move to the FIDS monitor in their area and check flight information. This represents that they could have forgotten the information or just want to be on the safe side and check again from time to time. From there they obtain a new target such as e.g. check-in desk or security which depends on their status, if they have checked-in already or if they have bags.

Table 4.1: Partly validated agent and airport properties implemented in the ABM.

Agent properties	Airport properties
Probability for online check-in	Number of flights/ passengers per hour
Number of check-in bags	Number of check-in desks
Average walking velocity [m/s]	Number of security lanes
Flight information (airline, flight number, check-in desk, gate, departure)	Number of gates
Checked-in (yes/no)	Check-in waiting/treatment time
Land-side or air-side	Security waiting/treatment time
Size (radius) [m]	Probability that gate changes
Minimum distance between agents [m]	Number of FIDS monitors
Waiting time at FIDS [s]	Probability that flight is cancelled
	Number of airport personal

The ABM to represent passenger movement is not final or complete and will be further developed in the course of this project. The passenger as well as the airport modelling will be enhanced with additional features to be more realistic. However, note that the model aims to represent the airport and passengers in an abstract way to capture main features but also to save computation time and to be able to calculate predictions. The abstraction level will be adjusted to allow the computation of realistic predictions for the scenarios.

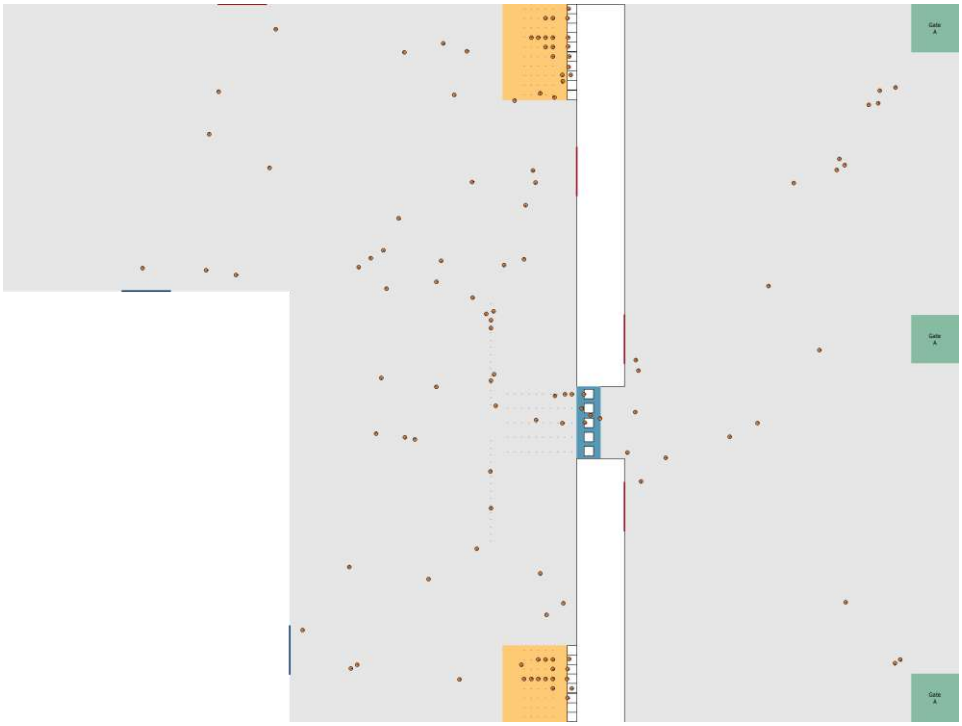

Figure 4.7: Screenshot of the ABM representing a small airport during normal operation. The passengers (orange dots) move in the airport area from entrance (blue lines) to check-in counter (yellow areas), to FIDS (red lines), to security check-point (blue grid) and finally to the gate (green areas).

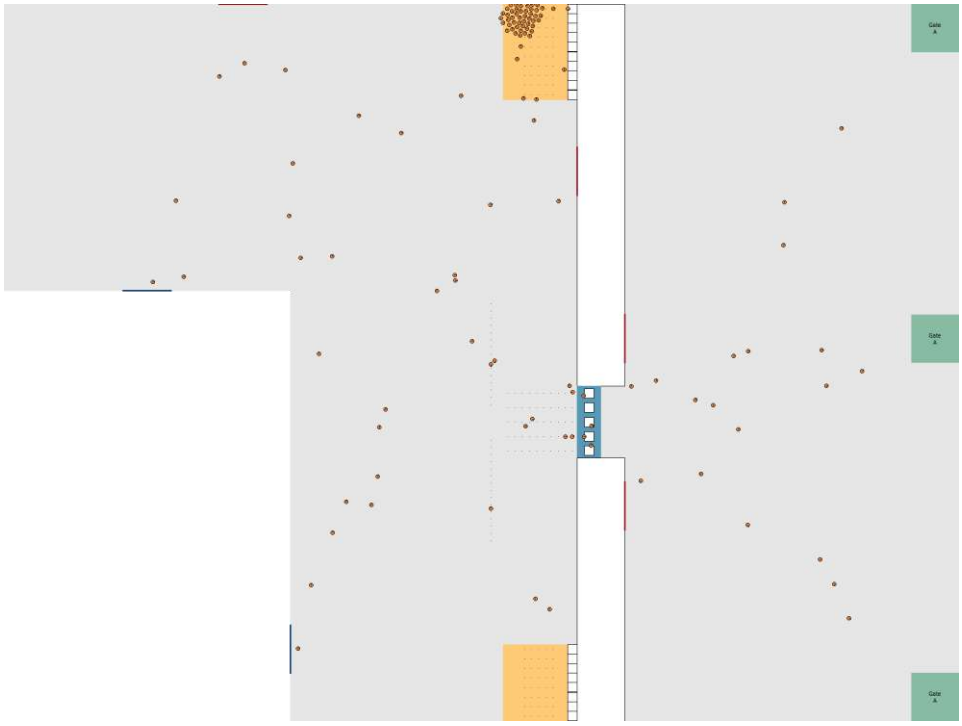

Figure 4.8: Screenshot of the ABM representing a small airport during disturbed operation. The passengers (orange dots) move in the airport area from entrance (blue lines) to check-in counter (yellow areas), to FIDS (red lines), to security check-point (blue grid) and finally to the gate (green areas).

The interface between the graph model and the ABM is accomplished by locating the graph nodes into the airport layout where the agents will be simulated.

To derive performance indicators and to model the airport's capacities, waiting times and passenger space values are considered. Further performance indicators, based on feedback from the airport operators, could be the flight delays, the percentage of people missing their flights and the turn-around times, i.e., the time an airplane takes from arriving at the stand until leaving the stand again.

4.1.4 Impact Propagation Model

The impact propagation through the network follows the interrelations between assets which are represented by connected nodes. Unconnected nodes do not affect each other. Consequently, cascading effects propagate from an initial node that is attacked/fails to the adjacent nodes under specific conditions.

If adjacent nodes are affected depends on the nodes' properties and on the type of connection. However, to specify the type of connection, information is still missing but will be collected in Task 5.1. Information about the nodes has been provided by the end-users and will be reflected in the implementation as so-called attributes such as asset type, flow layer and location (see Table 4.2). For the implementation of impact propagation through the networks, the simulation tool CaESAR developed by Fraunhofer EMI is used. It propagates failures through the network based on interconnectivity and flow capacities. If a node is not working properly its capacity can be reduced and the flow needs to be redistributed. If capacities of adjacent nodes are exceeded in the course of redistribution they can also fail or degrade in performance dependent on their properties. Not all types of networks can be represented by flow capacities and different strategies to interpret and represent the connections between assets need to be developed.

Here, passenger flow will be represented by an agent-based model. For information flow through the network, exceeded capacities might lead to information loss and redistribution needs to be introduced. Impact on the IT infrastructure is further assessed in detail in section 4.1.5. Baggage flow through the BHS requires baggage capacities for all involved assets to represent accumulations of baggage and the corresponding disruptions of service. The capacity information will be collected for implementation in Task 5.1. Airplane flow will be considered for the ABM to simulate the number of passengers arriving at the airport and passengers accumulating at check-in desks, security check-points and gates with respect to airplane frequency.

Nodes that have been attacked or damaged restore after a certain amount of time, the so-called mean time to repair. These restoration times are essential to estimate resilient behaviour of the system.

Table 4.2: Asset properties for the network model that are needed to predict the impact propagation and recovery of the system for the resilience assessment.

Attributes for impact propagation	Resilience assessment
Flow layer	Restoration times
Asset type	Performance measures
Connected nodes	Mitigation strategies
Location (geo-coordinates, buildings, etc.)	
Vulnerabilities	
Capacities for each flow layer	
Status (in service, mobile/stationary, etc.)	
Real/virtual	
Depth (underground)	
Height (above ground)	
Number of redundancies	
Material (e.g. temperature/water resistance)	

4.1.5 Business impact assessment methodology for cyber-threats

SATIE proposes a cyber-security MIA (mission impact assessment) methodology that simulates how cyber-threats propagate to the organization assets and assesses the impact caused on the organization business-processes and goals (i.e. mission).

As seen in Figure 4.9, the methodology is divided into two phases: threat propagation and impact assessment. In the first phase, threat propagation, users specify the initial conditions to start the simulation. These initial conditions include: (1) information about airport infrastructure (identified during Task 2.2) namely: airport asset lists; network connections between assets and threats present on assets; (2) information about business processes related to each asset (provided by SATIE cyber-threat detection system on business processes on Task 4.3); (3) threat affecting a given asset to simulate impact propagation. Threat propagation identifies the complete list of assets affected by a given simulated threat, by searching for paths from the initial compromised asset to the organization targets (specifically the mission assets), based on network connectivity and asset threats. Once the complete surface of the threat has been identified, the second phase of this methodology, impact

assessment, evaluates the damage the simulated threat can cause to the organization by analysing how compromised asset affect the execution of the business processes they are involved.

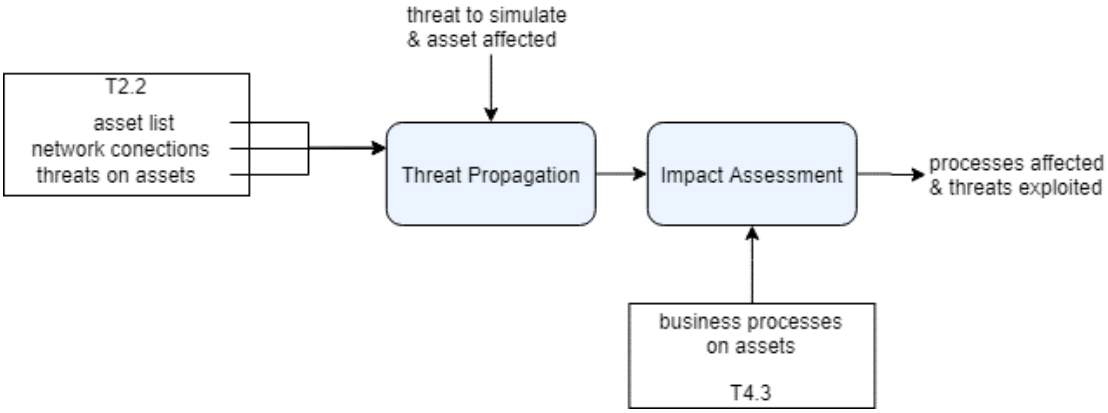

Figure 4.9: Architecture of the business impact assessment methodology.

4.1.5.1 Threat propagation phase

The threat propagation phase aims to assess how a compromised asset, chosen by the user, can reach and compromise the target assets (mission assets).

To this end, the methodology uses information about airport infrastructure and services (infrastructure assets, mission assets and connectivity), to search how a given compromised asset (entry point for simulation) can affect other assets by leveraging network connections and infrastructure's threats. The methodology starts from the conditions given by the entry-point and ends when all paths that reach the mission assets were explored. This results in a dependency attack graph Figure 4.10), where the root node depicts the entry-point, the end nodes depict mission assets (in this example: sorter and PLC) to which the threat has propagated (red circles in figure), and each arrow represents the attack paths that were found iteratively on the threat propagation. Threat propagation can be represented as a series of Horn clauses, a logical formula that takes a particular rule-like form: $L_0 \leftarrow L_1, \dots, L_n$, where $L_i \, \forall i \in N$ are literals, and if $L_1, \dots, L_n$ are true then L_0 is also true. Semantically, this type of clause represents the preconditions to reach a goal, which gives it useful properties to produce a dependency attack graph. To this end, the methodology being presented in this document, defines threat propagation using Horn clauses present in Table 4.3.

Table 4.3: Horn clauses defining threat propagation.

Propagation Step	Clause	Precondition
Initial asset compromised	compromiseAsset(asset 1)	attackerLocated(asset), threatExists(asset,threat A)
Asset compromised reaches other network assets	accessExists(asset 2)	compromiseAsset(asset1), connectivity(asset1, asset2)
Threat propagated to reachable asset	compromiseAsset(asset 2)	accessExists(asset 2), threatExists(asset 2, threat A)

Observing the first clause, *compromiseAsset(asset)*, it is possible to recognize how it can be used to represent the initial asset compromised by the simulated threat: if the attacker gets control of an asset,

represented by the condition *attackerLocated(asset 1)*, and there is a threat in the controlled asset given by the condition *threatExists(asset 1, threat A)* then all the preconditions of the first *compromiseAsset(asset 1)* clause are satisfied and one can derive that the asset 1 was compromised by the exploitation of threat A.

On the second clause, *accessExists(asset 2)*, if there is a compromised asset (derived, for instance, by the previous clause), and that asset, asset 1, has a way of communicate with another asset, asset 2, specified by a connectivity between the two, *connectivity (asset 1,asset 2)*, then there exists a way to access *asset 2* given by the satisfied clause a*ccessExists(asset 2)*.

On the third clause *compromiseAsset(asset 2)* considering that there is a way to access an asset, *accessExists(asset)*, and that asset has a threat *threatExists(asset,threat)*, then the asset can be compromised. This cascading effect resulting of iteratively validating each clause with existing literals can produce an attack graph step by step.

To illustrate this propagation effect, a small network consisting of four infrastructure assets is considered: a SCADA and a database; and two mission assets, a sorter and a programmable logic controller (PLC). The SCADA communicates with the database, the database with the sorter and the sorter with the PLC. Consider the assets in this example can be subjected to three types of threats: spoofing, tampering and denial of service (DoS). SCADA asset has a spoofing threat, database has a spoofing and tampering threat, sorter has a tampering threat, and PLC has a DoS threat. If threat propagation is to be applied to this example, for detecting how a simulated threat happening on SCADA can reach mission assets, sorter and the PLC, the initial conditions for threat propagation would be modelled as present on Table 4.4. In this example there are four main conditions: *connectivity(asset1, asset2)* representing existing connectivity among assets in the network; *threatExists(asset, threat)* depicting the threats identified in the network's assets; *attackerLocated(asset)* that represents the entry-point of the simulation – the user-chosen compromised asset; and finally, the *attackerGoal(asset)* condition used to define all the target assets (mission assets).

Table 4.4: Initial Horn conditions for threat propagation.

Condition	Description
connectivity(SCADA, database)	SCADA communicates with the database
connectivity(database,sorter)	Database communicates with the sorter
connectivity(sorter, PLC)	Sorter communicates with the PLC
threatExists(SCADA, spoofing)	SCADA asset has a spoofing threat
threatExists(database, spoofing)	Database has spoofing threat
threatExists(database, tampering)	Database has tampering threat
threatExists(sorter, tampering)	Sorter has a tampering threat
threatExists(PLC, DoS)	PLC has a DoS threat
attackerLocated(SCADA)	Active threat for propagation to be evaluated
attackerGoal(sorter)	Defining mission asset, Sorter, as a goal to reach in threat propagation
attackerGoal(PLC)	Defining mission asset, PLC, as a goal to reach in threat propagation

To perform threat propagation of the spoofing threat affecting the SCADA asset, the methodology proposed takes the initial conditions given in Table 4.4 of the network under test and compares them to the preconditions of the clauses previously defined in Table 4.3.

From the initial conditions, *attackerLocated(SCADA)* and *threatExists(SCADA, spoofing)*, it can be inferred that the attacker has compromised the SCADA by gaining access to this asset, and thus deriving a new condition: *compromiseAsset(SCADA)*. Next, the simulation continues to explore the system conditions (initial and derived) and by validating the clauses tries to advance throughout the network until it reaches the mission assets. The resulting attack graph to reach the mission assets, sorter and the PLC, would be given by the graph in Figure 4.10.

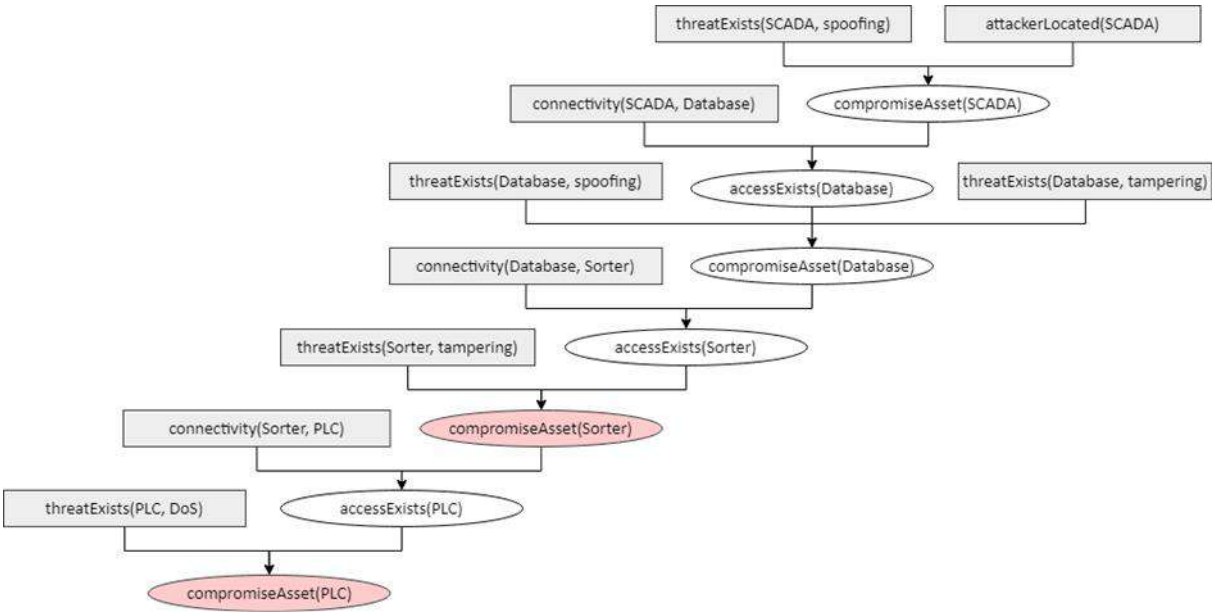

Figure 4.10: Example of an attack graph.

The attack graph in Figure 4.10 shows how the example simulation reached the mission assets, the PLC and sorter (red nodes): Beginning with the initial conditions defined by the user, the simulation derives the asset directly compromised by the simulated threat is the SCADA, and selects it as the entry-point for threat propagation. Combining this newly derived condition with the initial condition that SCADA is directly connected to the database, *connectivity(SCADA, database)*, the second clause is satisfied and derives a new condition the attacker can access the database, *accessExists(database).* Moreover, taking into account attackers located on SCADA have access the database and that, database can be subjected to tampering and spoofing threats, attackers can also compromise the database from the SCADA. Thus, database is identified as an asset indirectly affected by the simulated threat. Which lets the simulation advance to the database (*compromiseAsset(Database)* condition). This procedure of combining initial conditions (rectangle nodes) and iteratively validating the clauses defined (ellipse nodes) is repeated until the simulation reaches the mission assets (red nodes), sorter and PLC.

Threat propagation ends, once the complete list of assets compromised by the threat is identified. The impact assessment can then be conducted individually in each asset in order to understand, based on business processes the asset is involved, the damage caused to the organization operation capabilities.

4.1.5.2 Impact assessment phase

Following the result of the threat propagation phase, the impact assessment aims to map the compromised mission assets and their role on business-processes.

This phase uses information about business processes related to each asset (provided by SATIE cyber-threat detection system on business processes on Task 4.3) and the attack graph created on the previous phase. It traverses the attack graph, searches for compromised mission assets, and identifies for each asset the services affected and how they have impacted the organization mission based on the business activities and processes affected.

To illustrate how the mission impact assessment is accomplished the attack graph resulting from the example specified for the previous module (Figure 4.10) and a process called baggage screening is considered, specified as in Figure 4.11. This process is composed by two activities: the first activity give sortation order is supported by the application service baggage sorting that runs on the asset sorter; the second activity, move baggage to screening conveyor, is accomplished by the service baggage routing that is running on the asset PLC.

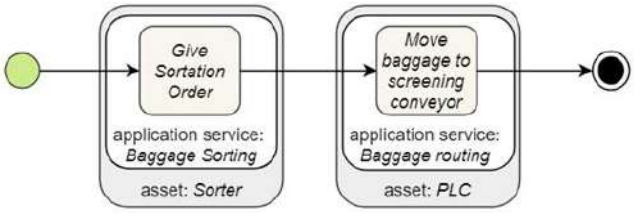

Figure 4.11: Example of business process specification describing baggage screening procedures.

The business-processes' specification is given as list of tuples <asset_identifier, service_name, activity_name, process_name>. In this case, two would be defined as:

(1) <Sorter, baggage sorting, give sortation order, baggage screening>
(2) <PLC, baggage routing, move baggage to screening conveyor, baggage screening >

As an illustrative exercise, combining the attack graph and the *baggage screening*'s process specification for the current example together would result in the Figure 4.12.

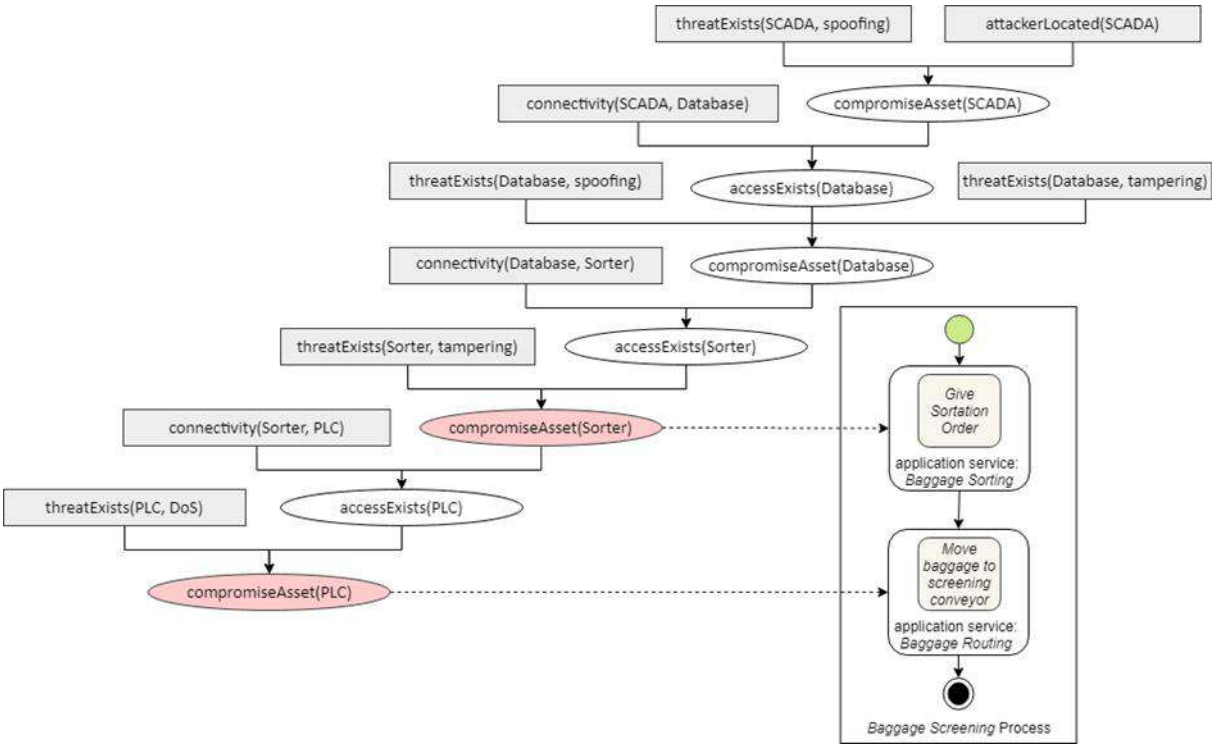

Figure 4.12: An illustrative example of how the attack graph relates to the baggage screening process given as example.

Traversing the attack graph (Figure 4.12 on the left) in the impact assessment phase it is found that the mission asset sorter was compromised. Looking at the information given by the tuple (1) this module can assess that the compromised asset sorter is running an application service baggage sorting that supports the give sortation order activity that belongs to the baggage screening process (Figure 4.12 on the right), which means the process baggage screening was compromised by the exploit of a tampering threat on the sorter asset.

From the second mission asset found compromised in the attack graph, the PLC, the same reasoning around the tuple (2) can be used to infer the baggage screening process was compromised by a DoS threat exploit. It is to be noted that the same process was twice found as impacted, thus this module further proceeds to remove redundancies. Finally, the proposed solution presents at the end the impacted processes, threats exploited, attack paths used and the compromised assets involved in the organization's mission.

4.1.6 General inputs and outputs of the Impact Propagation Simulation

In this section, an outlook to the Impact Propagation Simulation that will be implemented in Task 5.1 is given which is based on the modelling described in this report. The input and outputs of the simulation are needed for the discussion of interrelations with the adjacent tools (see sections 4.2.3 and 4.2.4).

The simulation requires several inputs as general preparation such as information about the flow layers, the interrelations between assets and systems, specifications for the agent-based model and performance indicators. However, in real-time the simulation needs to be fed with incident information which will be provided by the Incident Management Portal. The Impact Propagation Simulation produces time series data of performance which quantifies the system's resilience. The propagation of an incident through the system enables to make predictions about additional assets being affected in the future. Further, it is able to compare different mitigation strategies and to deliver decision support. Generally, the Impact Propagation Model is able to identify critical assets in the system mainly by analysing the general network structure.

Time series data of impact propagation and performance will be provided to the Incident Management Portal and the Crisis Alerting System (see Figure 4.14). Both tools will develop a visualization of the impact propagation through the systems for SOC operators and local authorities. Predictions of affected assets through propagation following an incident will be provided to the Incident Management Portal and decision support will be provided to the Crisis Alerting System. However, the final definition of information exchange and the technical implementation will be done in Task 5.1.

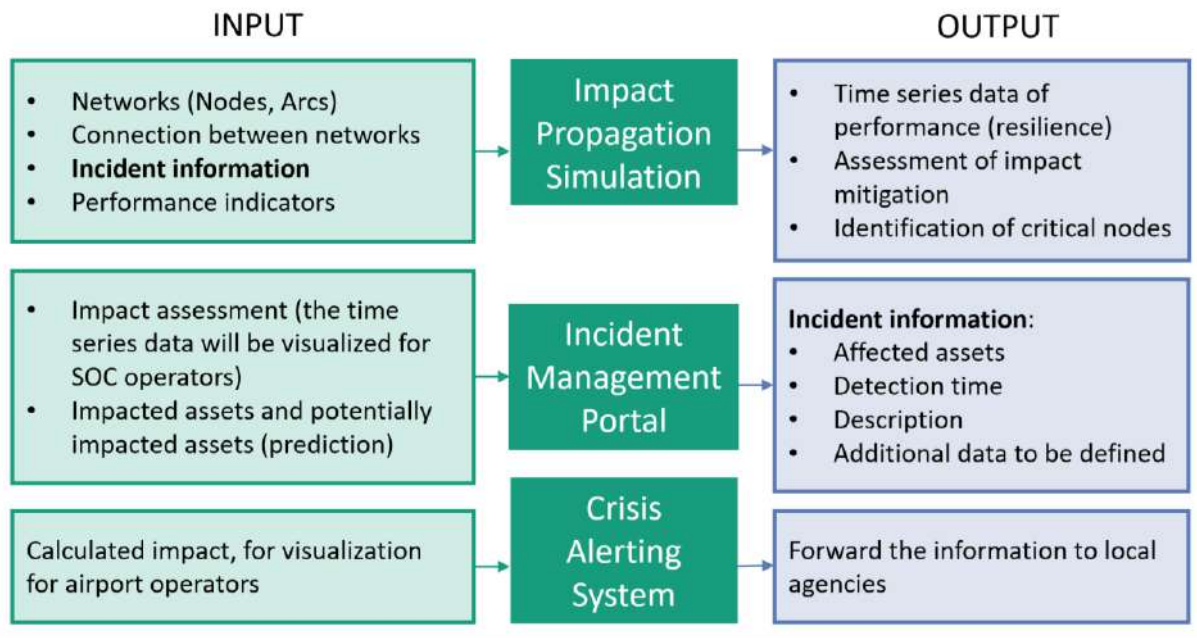

Figure 4.13: Interrelations and information exchange between the tools Impact Propagation Simulation, Incident Management Portal and Crisis Alerting System.

4.2 Check and validation of the model

In this section, the Impact Propagation Model is reviewed from different perspectives. First, cyber-physical threat scenarios are used to understand and test the propagation through the network structures. Next, the model is reviewed with respect to societal and human impacts with a focus on the ABM. Finally, the interfaces between the Impact Propagation Simulation, the Incident Management Portal and the Crisis Alerting System are analysed.

4.2.1 Using cyber-physical threat scenarios

In order to validate the models described above, the asset interrelationship information in Figure 4.2, Figure 4.3, Figure 4.4, Figure 4.5 and Figure 4.6 along with information from the cyber-physical risk analysis about which threats can impinge on those assets, a validation approach has been devised. By considering other attributes of the assets such as their physical locations with respect to each other, the asset type, and whether they are physical or digital assets, these attributes will be used to better determine which threats can propagate between which connected assets, and which threats can transform during this propagation.

The attributes included are based on the types of attributes commonly used in risk assessments. By creating a threat scenario for each of the airport scenarios in the SATIE project and determining how one of the steps could propagate through the asset network, considering the above properties, the results can be compared to those of the models proposed in this deliverable. This will elucidate whether the model, though simplified, contains enough information to reasonably predict how the threat will propagate or if one or more information attributes are vital to add.

It is important to note that this information does not reflect the probabilities of propagations based on real vulnerabilities or risks in a particular airport environment, but are hypothetical, potential propagations for airports in general based on how assets are likely connected.

4.2.1.1 Threat transfer conditions

To not be biased in this validation phase, the rules for how threats can transfer in the models were created before the creation of the models. The two main pieces of necessary information were the asset-threat relationships and the asset-asset relationships. All assets have a subset of threats which can impact them; this is based on the RIS tool created by NIS which is used in the IT world in general. Below is the list of all information which will be used in the model:

- Asset class (defined by end-user).
- Asset type (dependent on asset class).
- Asset sub-category (dependent on asset type).
- Asset category (dependent on asset sub-category).
- Threats which can impact an asset (given by the RIS tool, dependent on asset type).
- Flow layer(s) to which an asset can belong: baggage, information, aircraft, employee, passenger (defined by FHG and the end-users).
- Location of an asset: public, restricted, unknown (given by end-users for Scenarios #1 and #2; see section 4.2.1.4, section 4.2.1.5, and section 4.2.1.6 for how the lack of this information in the other scenarios caused issues).
- Whether an asset is physical or digital (defined by end-users).
- System to which the asset belongs (e.g. FIDS, PA; defined by end-users).

The directed interrelationships of the assets completed by the end-users were used to know which assets could impact which others (the arrows used in the rules below indicate which direction is necessary for the propagation). The following additional rules apply to all scenarios:

1. *Threat A* impacting *Asset A* can also impact *Asset B* if *Asset A* is connected [→] to *Asset B* and if *Asset B*'s threat list includes *Threat A*.
2. A terrorist or malicious person can:
    a. Cause any threats which are connected to the vulnerability *Compromised Personnel*.
    b. Cause the threats in (a) to impact any physical assets (attribute Phys_Dig = physical) within reach (attribute Location = public).
    c. Cause the threats in (a) to impact any digital assets (attribute Phys_Dig = digital) which are connected [←] to one of the above physical assets in (b).
3. All assets in the designated public area, by the end-users, are assigned the attribute *Location = public*. If an asset *security door* is compromised (i.e. impacted by one of its threats), then all assets on the other side of that door are assigned the attribute *Location = public*.

There are some threats which may transform into others. If access is gained to a particular asset through one way, it may allow other attacks on other assets. Therefore, for simplification, below is the list of threat transformation rules which are applicable to these five test scenarios. These rules apply at all times (unless otherwise specified).

- The threat *communication infiltration* can transform into any of the following threats when two assets are connected [→]:
    - *Denial of service*.
    - *Intercepting data in transit*.
    - *False information insertion*.
    - *Forbidden access to network, basic software, applications*.
    - *Listening to unauthorized communication*.
    - *Masquerading*.
    - *Message routing problems*.
    - *Unauthorized network and resource use*.
    - *Wiretapping*.
    - *Information management equipment tampering*.

- *information management equipment tampering* can transform into:
  - *Technological equipment tampering*.
- technological equipment tampering can transform into:
  - *Unauthorized access to physical areas* (assuming access control is altered).
- *Theft* (of credentials or information containing credentials) can transform into:
  - Any other threat of an asset connected [←] to the originating asset.
- *Physical attack and consequent unauthorized access to the secured zone* can transform into:
  - *Unauthorized access to physical areas*.

4.2.1.2 Definition, test and results of test scenario #1

The first test scenario #1 threat propagation is derived from one of the steps contained in Scenario #1 and was chosen to include a cyber-threat that transforms into a physical threat in order to determine whether the model could similarly replicate this transformation. The chosen threat is a tampering of access control credentials in steps 5-6 outlined in the Declaration of Action: "A cyber-attack modifies credentials on security doors, allowing unauthorized personnel to access secure areas creating security breaches and halting airport operations, delays in flights and a possible physical attack." The starting condition of these steps is that the electronic access control information is modified by a cyber-attack such that the doors are no longer selective, allowing unauthorized people to enter and as a consequence causing potential problems to airport operations due to their access.

Hypothesized test scenario #1 threat propagation details:

1. The threat *communication infiltration* impacts the asset *Access Control (AC) – external firewall.*
2. The threat propagates through *AC – switches* and transforms into *false information insertion* of a database.
3. The added credentials then allow an unauthorized person to open the asset security doors.
4. This unauthorized person now has access to assets to which they can cause significant physical and digital damage, bringing the airport to a halt.

The model in this section will be represented with a DAG, which is arranged using the Kamada–Kawai algorithm which is a variant of a force-directed graph drawing algorithm to minimize the number of edge crossings and thus congregate closely related nodes with each other (Kamada & Satoru, 1989). In the following Figure 4.15, the nodes are color-coded according to system (see legend) and for legibility, the nodes are not labelled.

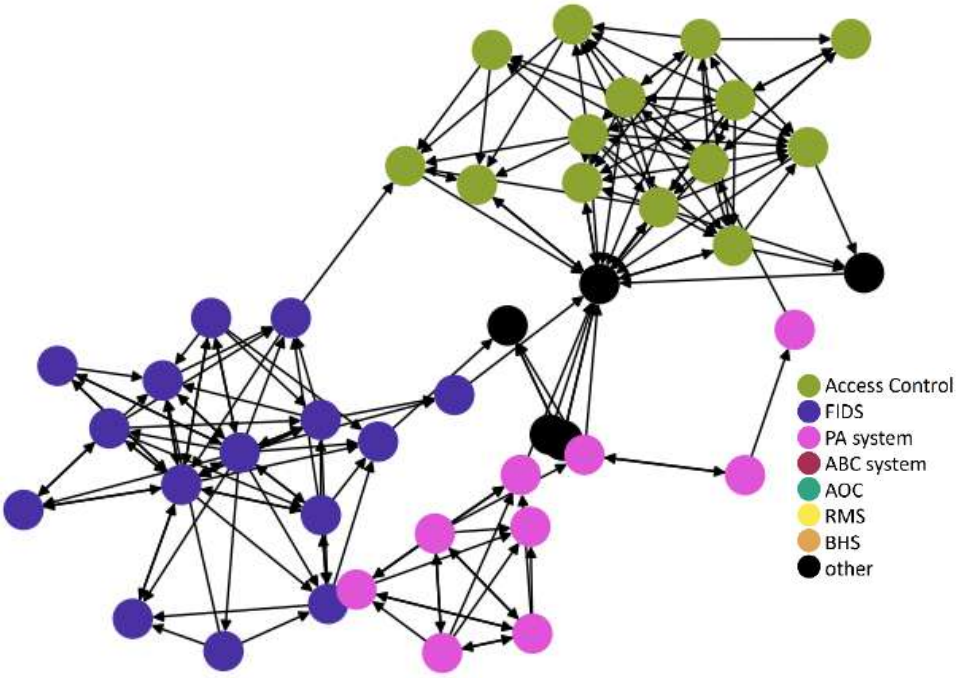

Figure 4.14: Example asset interrelations with each asset represented as a node.

The view in Figure 4.14 shows an overview of the assets and demonstrates that the drawing algorithm worked tremendously well to segregate the assets from the different systems. This organization will aide in the interpretations of the results later on. This test threat scenario is about an unauthorized person gaining access and wreaking havoc on many airport systems. Therefore, first one must know which assets a person can impact in a normal state. Then after the initial threat impacts an asset and it potentially propagates, the subset of impacted assets can be compared to this baseline. Figure 4.15 shows which assets, in yellow, can be impacted by the unauthorized person who has only entered the airport environment.

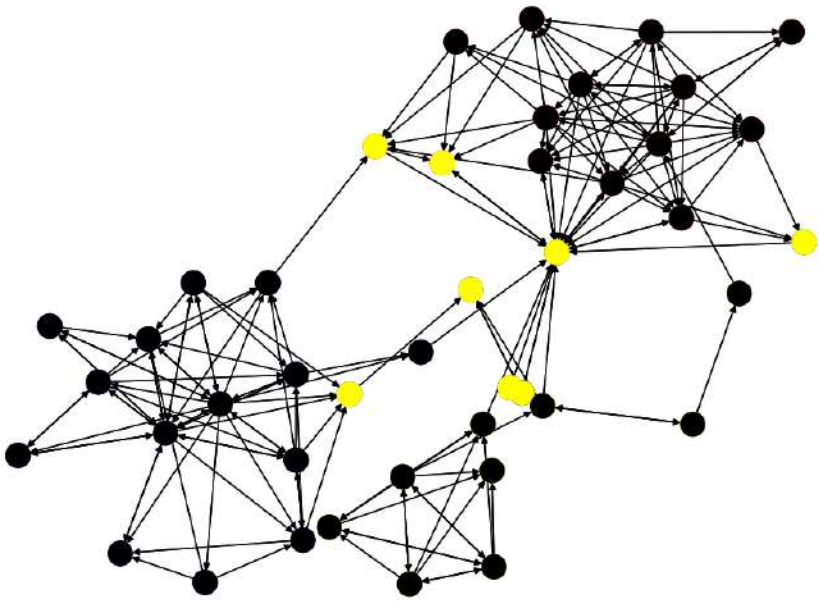

Figure 4.15: The asset interrelations, highlighting which assets (in yellow) an unauthorized person can access without any threat occurrence first during test scenario #1.

The assets which can potentially be impacted by the public in a normal situation are:

- Security doors.
- Card reader.
- Employees.
- ID cards.
- FIDS – 500 monitors.
- Muster station – indoor.
- Muster station – outdoor.
- Passengers.

These are assets which are commonly found in public areas. Assets which require higher security such as passing through access control doors with a legitimate badge (e.g. AC server) are not accessible at this point. This test threat scenario starts with the threat *communication infiltration* on the asset *AC – external firewall* (shown below in red in the upper left-hand network diagram). This asset is not one of the originally-accessible assets to the public, but is only accessible through a cyber-attack. The figure below shows the propagation through the network (impacted assets are in red).

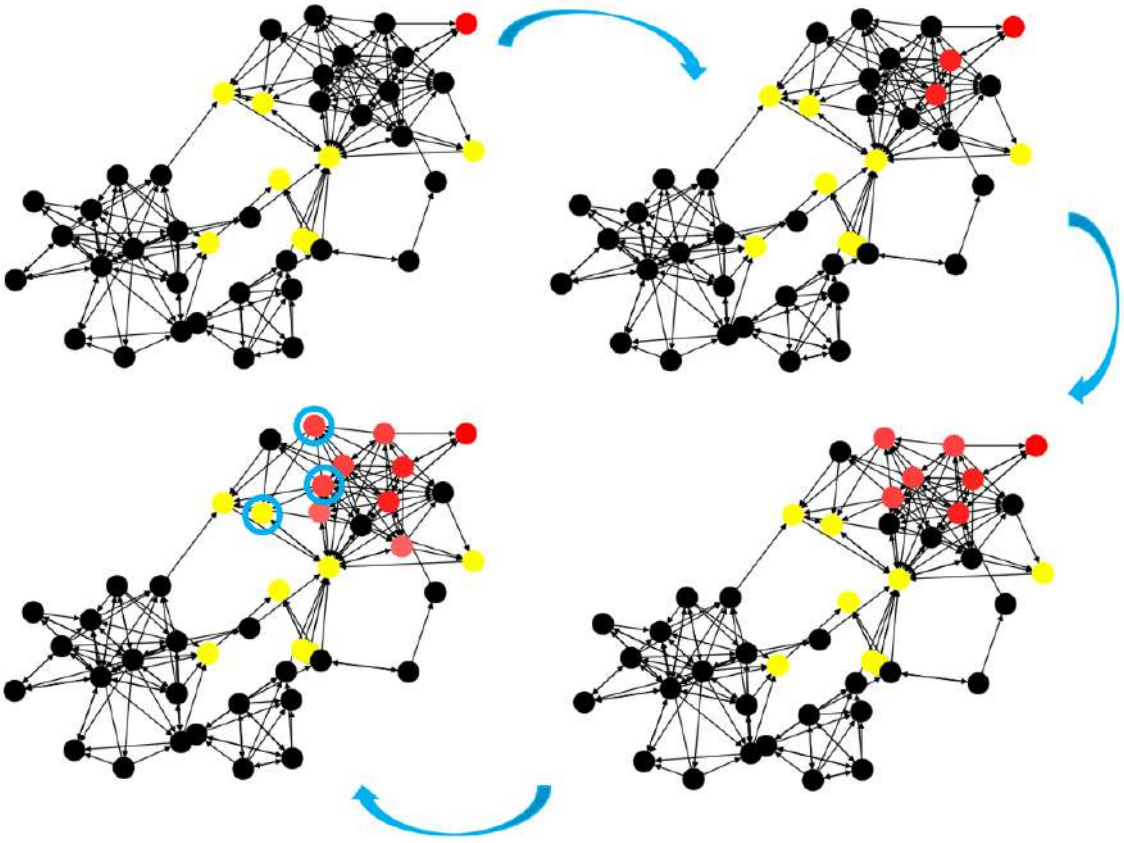

Figure 4.16: Threat propagation during the test scenario #1 to an access control firewall.

The progression of the threat propagation, following the rules outlined in section 4.2.1.1 is shown in a clockwise direction. It is very interesting that these diagrams visually show how a cyber-attack to an asset with restricted access, gradually impacts more assets, and ones which are closer to those accessible to the public (in yellow).

The crux of this test threat was whether the asset *security doors* could be impacted (the yellow node circled in blue in the bottom left-hand diagram), through the false insertion of credentials to the access control system. In this test propagation, security doors are not impacted. The ultimate reason was because there was no asset *access control database* where the falsified credentials could be inserted, even though there is an *access control server* (one of the blue-encircled red nodes) and an *access*

control workstation (the other blue-encircled red node), which were both affected. Therefore, the threat propagated very close to the feared asset but did not reach it. This demonstrates the need for a complete asset inventory, especially in order to model the kind of impacts that could occur in an airport environment. Without that missing asset, it would seem as if unauthorized people could not gain access to restricted systems and therefore the results are misleading.

4.2.1.3 Definition, test and results of test scenario #2

The second threat propagation test is derived from two of the steps from Scenario #2 and was similarly chosen to include a cyber-threat which transforms into a physical one. The chosen steps are Steps 1, 2 & 3: "Malicious users gain unauthorized access to the police network and through a man-in-the-middle attack, simulate the back-end police systems which the Automatic Boarding Control (ABC) system interfaces with for background checks. All travels cross the gates without any actual background checks taking place, due to the MITM attack, allowing potential terrorists with genuine passports to enter onto EU soil." This situation starts with a cyber-virus into the police database and eventually resulting in people not being allowed through the gate, which should be, and consequently congestion and confusion in the security area. In this scenario, it is actually not possible to reach the police background check system through the internet because it is not connected to the internet, so one must assume that a malicious person gains physical access to the room with these systems. Therefore, the following is the hypothesized threat propagation for test scenario 2:

- The threat *forbidden access to network, basic software, applications* occurs to the asset Automatic Boarding Control administration PC.
- That turns into the threat *communication infiltration* to the asset *police backend system database* so that this attack can intervene when the database is being queried whether a person trying to cross an e-gate has a clear background check or should be flagged.
- These falsified responses then turn into *unauthorized access to physical areas* to the asset *e-gate (barrier)*.
- That threat should then be able to lead to a physical threat, such as *bomb* to the asset *Main Terminal Building (MTB) – arrivals gates.*

To start, below in Figure 4.17 is a DAG of the asset interrelationship, according to the end-user, for test scenario 2. This means that when an asset (a node) is negatively impacted and this will impact other assets, there is an arrow leading to the subsequently impacted assets (nodes). Each system has a different colour, and nodes which are not connected to others means that when that asset in particular is negatively impacted, it will not propagate to other assets.

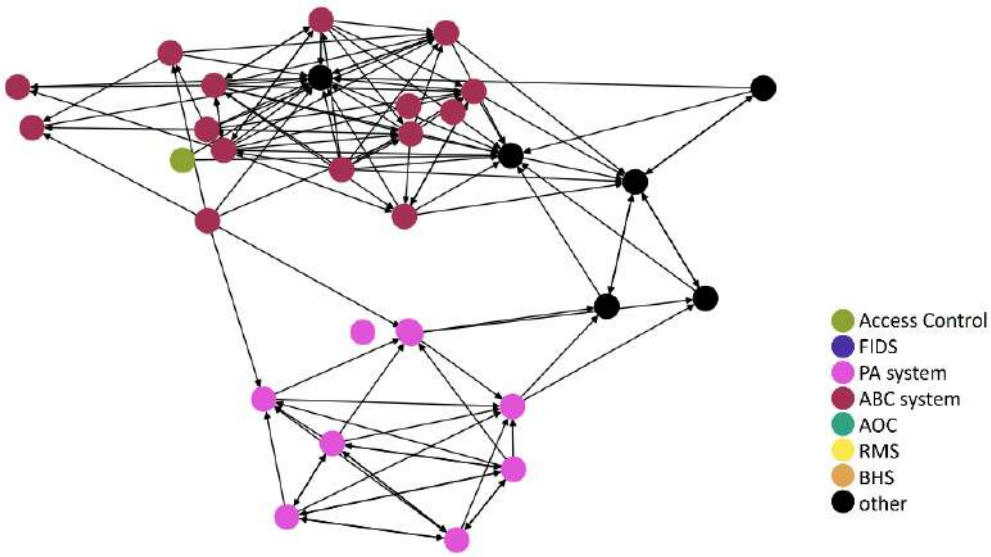

Figure 4.17: Example asset interrelations with each asset represented as a node.

The initial starting position of this test threat propagation is that there will be a threat to the *ABC administration PC* (shown below in red), while the final asset that should be reached if this threat propagates as predicted, is *the MTB – arrivals gates* (shown below in yellow). The figure below shows that surprisingly this threat would stop after reaching only one other asset.

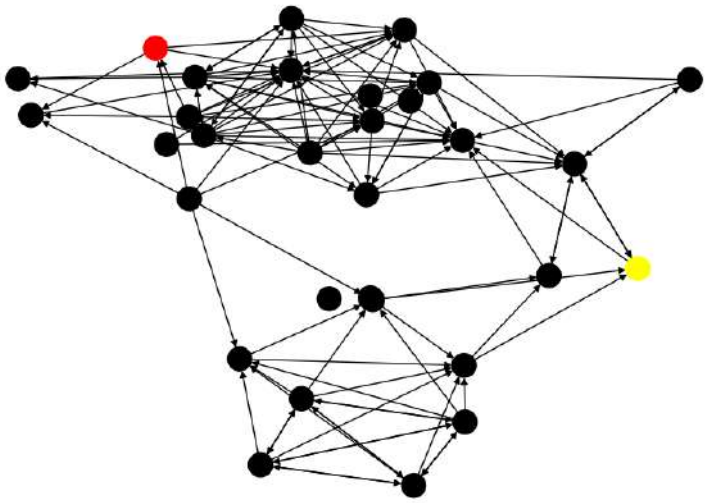

Figure 4.18: The asset interrelations during test scenario #2 with the initial asset in red and objective asset in yellow.

The propagation stops because there are no subsequent assets to be impacted which can be impacted according to the threat rules outlined above in 4.2.1.1. After a closer inspection, and thinking logically about how these systems are set-up, this threat halted because there was no link going from the *ABC administration PC* to the *ABC software*, nor was there a link between the *police database* and the *e-gates*. If a connection from the *ABC administration PC* to the *ABC software* is added, then for the sake of experimentation, below is how the threat would continue to propagate if it were connected to *ABC switches*. Whether or not this propagation is valid depends on the precise setup in the airport environment.

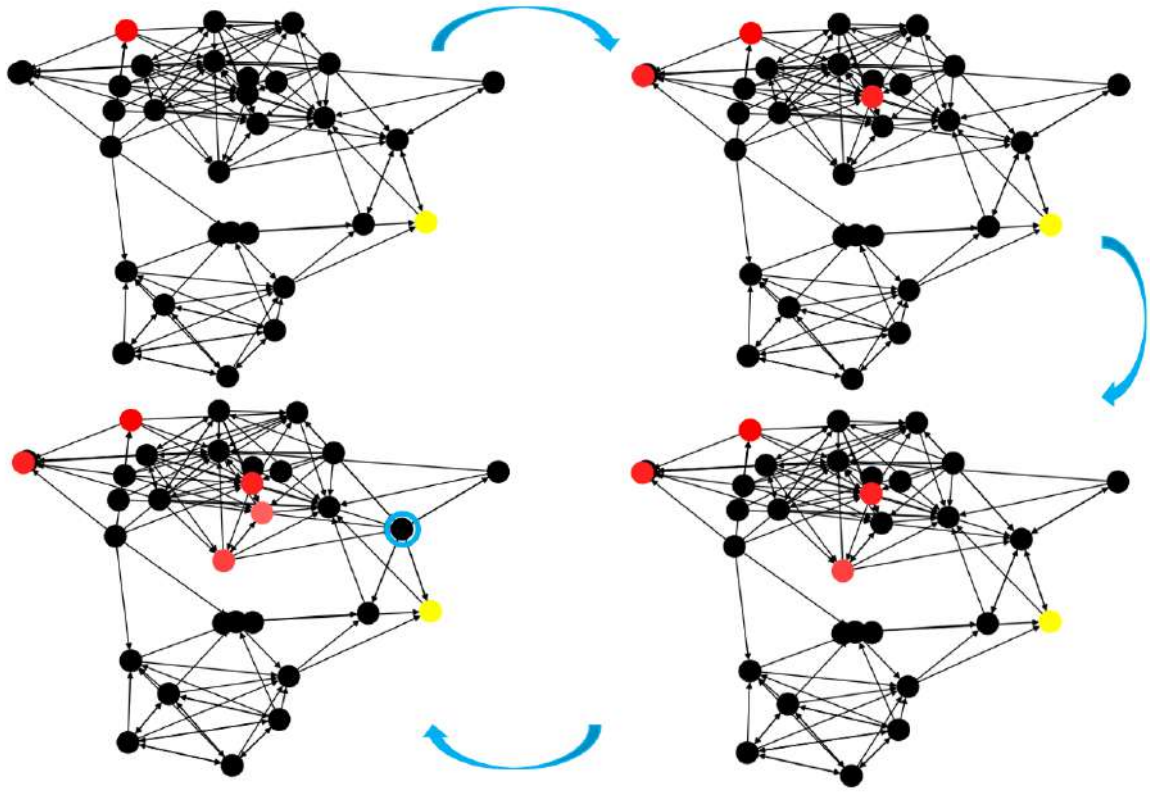

Figure 4.19: Threat propagation of test scenario #2 assuming an added interrelationship.

An interesting thing to note with this threat propagation is that it was very slow; there was not a fast dispersion of threats but almost a step-by-step propagation. This can be a great thing from the organization's perspective but things are not connected excessively or there are proper safeguards in place to prevent that. However, the fact that the last asset reached was the *e-gates (barrier)* with the threat *unauthorized access to physical areas* and yet it stopped there and did not reach a threat to the *arrivals area* (yellow node) ultimately seems to indicate that the asset interrelations were perhaps not complete. In this network, there needed to be threat propagation through *employees* (blue-encircled node) in order to affect the arrivals area.

This scenario demonstrates that the relationships between the assets must be complete in order to get a network which properly represents the real airport systems.

4.2.1.4 Definition, test and results of test scenario #3

The test threat for test scenario #3 is the case where a physical threat then becomes a cyber-threat. It comes from the first two steps of the scenario outlined in the Declaration of Action, where physical access turns into a cyber-attack: "Physical access by an unauthorized person to the Airport Operations Control (AOC) room is gained through using authorized badges to pass to the air-side area, then the satellite building, and then to the AOC room; this physical access then allows the unauthorized person to implement a cyber-attack to the Airport Operation Database (AODB)." The asset interrelationships are shown in the figure below, which each asset (node) coloured according to the system that it belongs to.

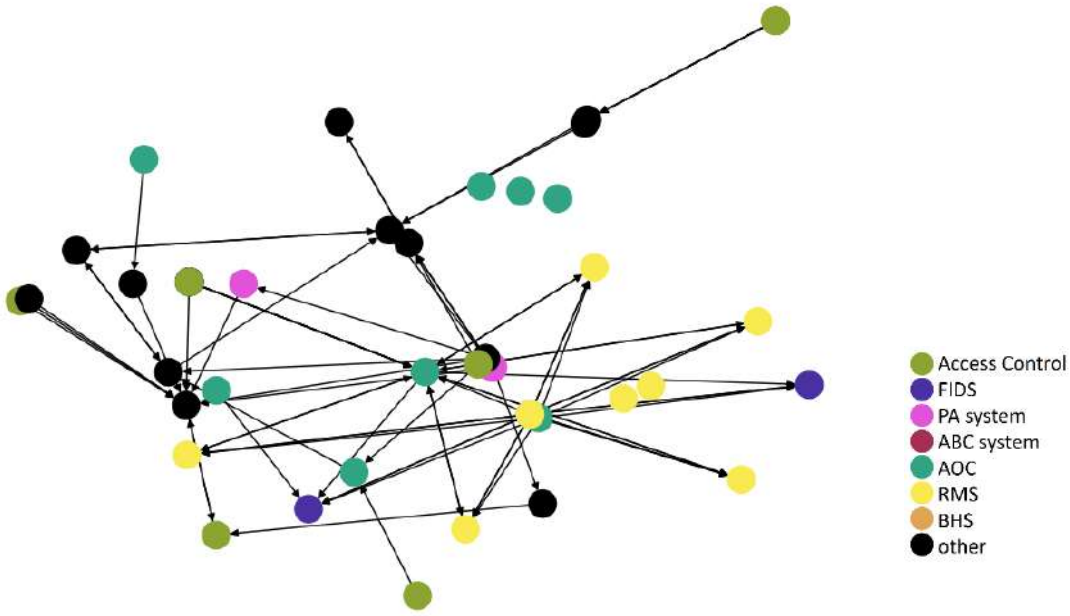

Figure 4.20: Example asset interrelations with each asset represented as a node.

Figure 4.20 was arranged following the same Kamada–Kawai algorithm used in all of these test propagations, clustering related nodes closer together (Kamada & Satoru, 1989). However, the nodes with the same colour are not significantly near each other, indicating that the assets belonging to the same system were not connected to each other closely. Whether this is intentional by the organization's management or not, it will make the propagation results very interesting.

The hypothesized test threat propagation is as follows:

- The threat *theft* occurs to the asset *badge.*
- This then turns into the threat *unauthorized access to physical areas* through the asset *security door*.
- Which eventually becomes *false information insertion* to the asset *AODB – Oracle DB* which is a type of database (DB).

The starting situation of this propagation is shown in the figure below (Figure 4.21) in the upper left corner with the asset *badge* in red, and the objective asset, the *AODB – Oracle DB*, shown in yellow. An interesting thing to note from the beginning is that the initial asset can only impact one other asset, but in the end, the *database* asset (in yellow) has many arrows heading into it, which indicates it can be impacted by many assets. It will be informative to understand which threat through which asset reaches the database the easiest (following the simple rules previously outlined), although it does not mean this would be the only way to negatively impact that asset.

Under these circumstances, the threat would propagate only one step further and then stop. This is because it would reach the asset *finger – badge access to door* but the threats to that asset could not transform into a threat to the only subsequent asset, *aircraft – stand* which is an outdoor location. This indicates a weakness in the validation model, which should be further developed, tailored to the airport environment. The threat *unauthorized access to physical areas* which impacts the "finger" should be applicable to a location, logically. Therefore, here it is worth exploring what would happen if this threat could propagate (assuming further refinement of the validation model).

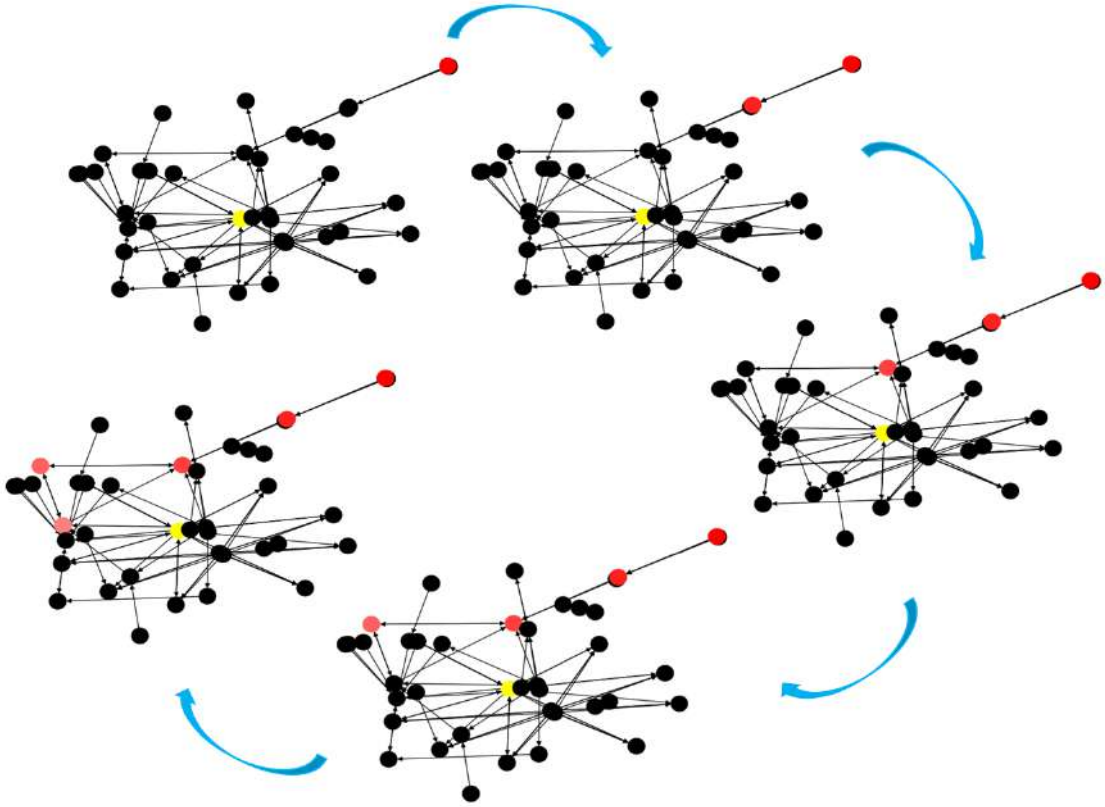

Figure 4.21: The test threat propagation starting on the upper left and continuing clockwise.

The last asset reached with this test was *finger – location* which interestingly can itself be impacted by the database, but not vice versa. This threat stayed among physical assets (*badge, finger – badge access to door, aircraft stand, aircraft, and finger – location*) and never made it to the airport systems. It seems there was a missing relationship between the badge and gaining access to the AOC room.

Overall this test scenario reveals a very distributed set of assets and that with this organization; the threat was not able to propagate into the airport systems with valuable and sensitive information.

4.2.1.5 Definition, test and results of test scenario #4

This scenario is unique in that the whole scenario takes place within one system, the BHS. To test whether a cyber-threat can propagate into a physical threat, step 4.3 of the Scenario #4 will be used: "Cyber-attack on the BHS database that corrupts the baggage destination." This means that the following is the hypothesized threat propagation:

- A *communication infiltration* cyber-attack on the asset *SAC Database* which is a part of the sortation allocation computer (SAC) system.
- This threat works its way through the network, possibly transforming.
- Which in turn results in a threat impacting the asset *baggage*.

Figure 4.22 shows the connections between the assets in this scenario and that most of the assets belong to only two main systems, but the assets are not tightly linked to each other.

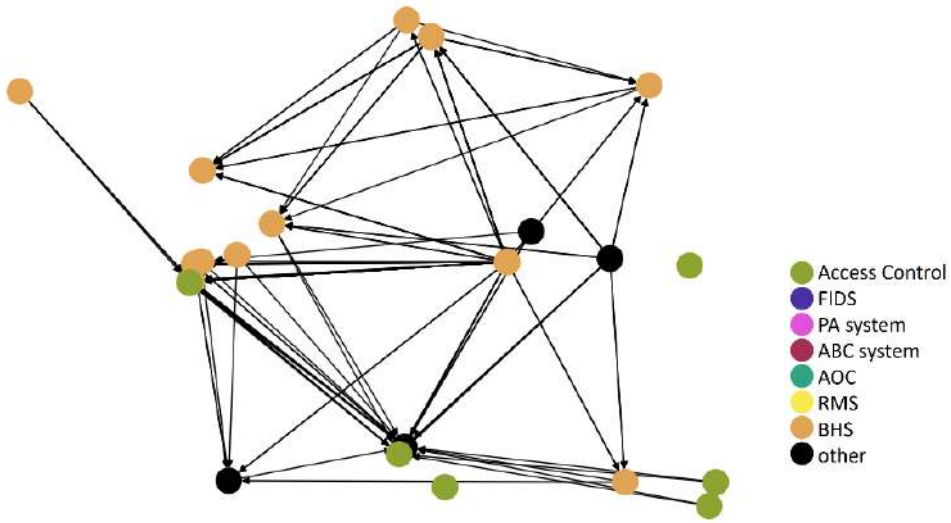

Figure 4.22: Example asset interrelations with each asset represented as a node.

In this case, the first step is the *communication infiltration* to the *SAC database* (in red), which as the diagram below shows, is directly connected to baggage (in yellow). This means that a threat to the database has the potential to directly impact baggage.

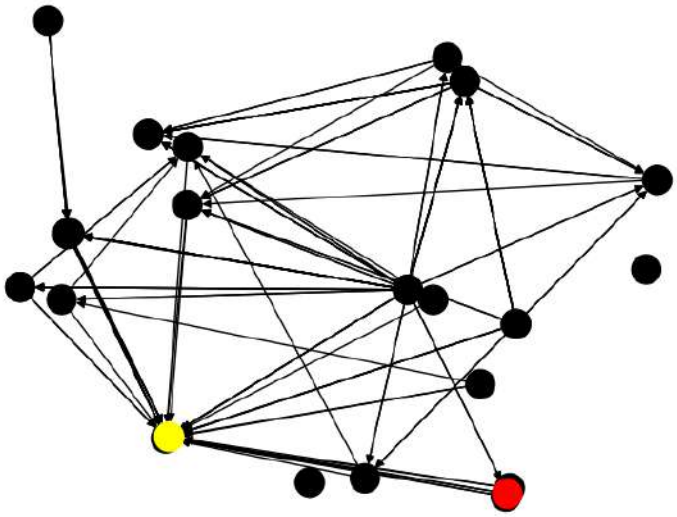

Figure 4.23: The starting condition of this test threat, demonstrating a direct connection between the originating and final assets and no propagation necessary.

Therefore, this does make sense as the final step in the overall scenario description. Two steps before there is a cyber-attack on the SCADA system. Would a cyber-attack to the SCADA database be able to lead to an impact on the baggage? Below (Figure 4.24) is how that threat propagation would play out.

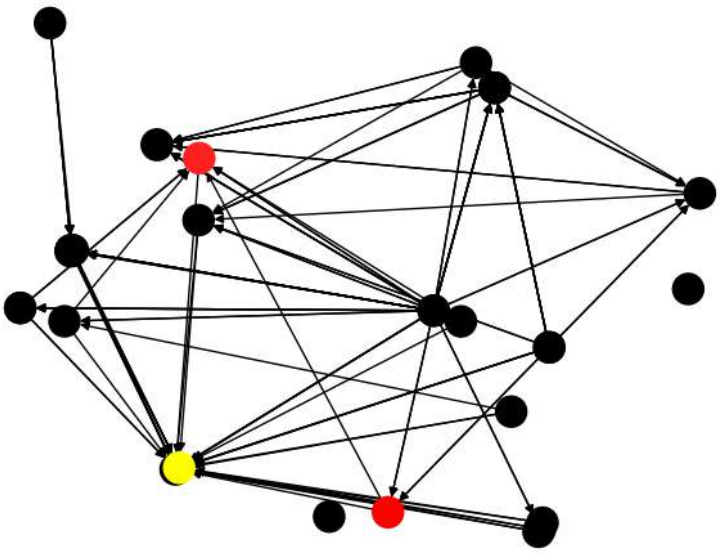

Figure 4.24: Resulting hypothetical threat propagation for test scenario #4.

In fact, it would end very quickly as the propagation would stop at the asset *PDA* and not be able to transform into a threat which could impact the subsequent asset, *employees.*

In conclusion, this test threat indicates that the original threat scenario was thought out well, creating a separate cyber-attack for the *SAC database* because one just to the SCADA database would not be sufficient to reach the baggage and change its destination. Therefore, this threat propagation model, even if simplified, can be useful to understand how systems are separated logically and whether it is enough or too much for the organization's business needs.

4.2.1.6 Definition, test and results of test scenario #5

Scenario #5 takes place in the ATC tower and starts with a physical intrusion, like in Scenario #3, which becomes a cyber-attack to some of the airport systems. Therefore, it would be interesting to use the equivalent steps to test how well, if at all, the threat propagates in this environment. Therefore, the steps to be modelled will be steps 1 & 2: "Physical intrusion in the technical cabinet room of the ATC tower, man-in-the-middle cyber-attack on flight plan services and modified flight plans are shared with pilots. "

The hypothesized propagation is:

- The threat *physical attack and consequent unauthorized access* to the secured zone on the asset *door/barrier*.
- This then leads to a *communication infiltration* (generic term for attacks such as MITM) attack on the asset *flight plans*.
- The threat to the *flight plans* then turns into a threat to the asset *aircraft.*

The asset interrelationships are according to the following Figure 4.25.

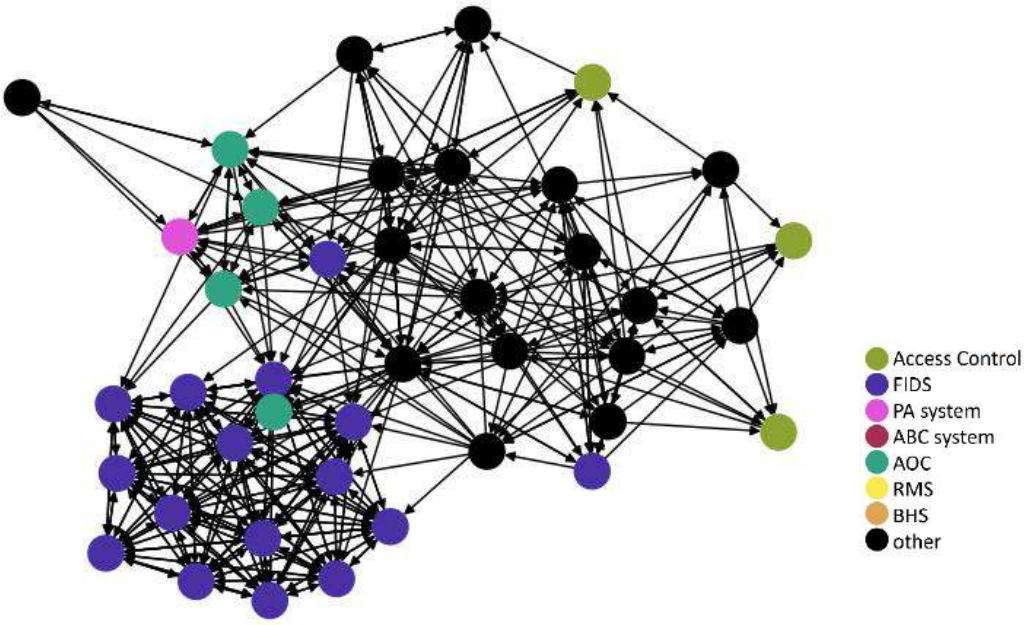

Figure 4.25: Example asset interrelations with each asset represented as a node.

This network (Figure 4.25) is highly interconnected, which makes sense given that most of these assets are related to airport IT systems. Many of the black nodes are network-related assets, but without explicit knowledge as to which networks they belong, they were left in black. Starting with a physical attack on the asset *door/barrier*, and with the ultimate goal (for the attacker) to reach the asset *aircraft*, below (Figure 4.26) is how the threat would propagate through this network (starting in the upper left and going clockwise).

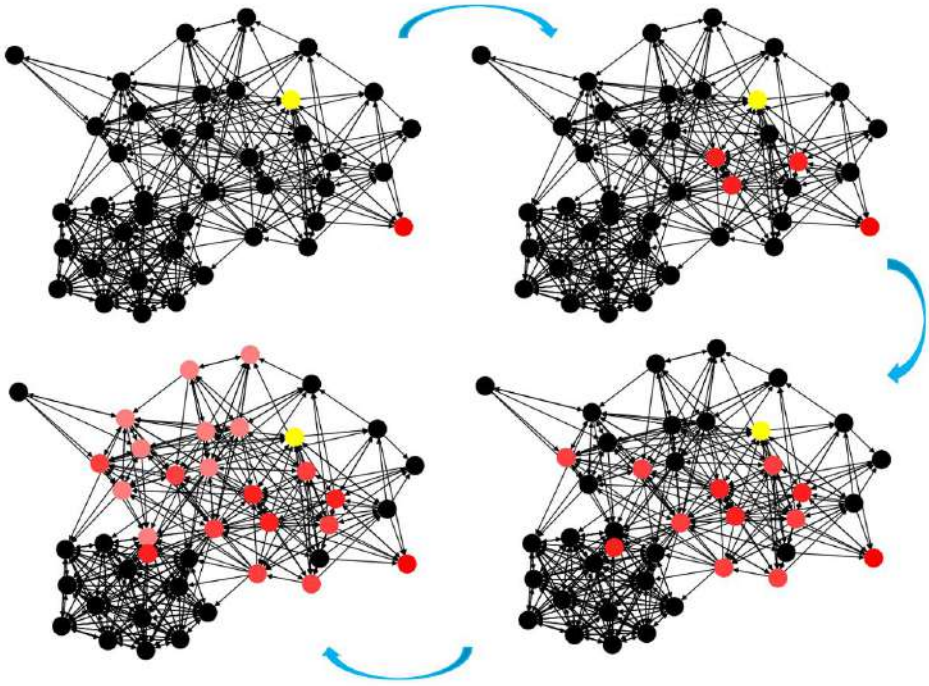

Figure 4.26: The threat propagation test during test scenario #5.

This threat propagated easily through the assets, this was mostly due to having many network and strategic document assets which can all be impacted through communication infiltration or false data insertion. The objective asset, *aircraft*, ends us being almost completely surrounded by potential

threats and yet not impacted. Taking a closer look, this was because, even though the flight plan information was tampered with in the final stage (bottom left diagram), the relevant threats could not transform into a threat applicable to aircraft, despite these assets being connected.

This threat scenario test demonstrates a potential missing threat for the asset class of aircraft, which for now only contains physical threats. Yet, in reality, the aircraft is equipped with numerous digital IT and OT assets, which should be taken into consideration in the potential threats.

4.2.1.7 Validation conclusions about FHG's model

The validation tests of these five scenarios with very different asset interrelationship networks demonstrated a few, quite different, weaknesses. test scenario #1 and test scenario #5 demonstrated the importance of having a complete asset inventory. In both of these situations, there were potential assets missing which would have allowed threats to propagate which can normally propagate. Test scenario #2 demonstrated the importance of having a complete interrelationship graph because in the real world, systems and assets are highly interconnected and representing this accurately is essential for a valid propagation model, to better detect potential issues for the airport. Test scenario #5 revealed the importance for the model to consider the nuances of the particular environment in which the model is being applied. Test scenario #4, on the other hand, exposed another use for a validation model, where not only can one test whether a threat will propagate, but how easily and through how many steps, to get a better understanding of the asset network.

Overall, more information would make these models even more informative, such as the physical distances between the assets to add in an element of time and probability of propagation, along with aspects like more specific access control, whether, even if one has access to a computer and its operating system, does one need digital credentials to log-on to the database, which would halt some of the threats which propagated successfully in these models.

However, without getting excessively detailed (there can be no definitive level of detail which is enough), the above models represent a good starting point for the end-users to understand their systems even better. If the asset inventory and interrelationships match reality, these can be especially helpful in understanding what kind of threats propagate easier than others or how quickly (through how many steps) it can propagate, all in the name of aiding threat prevention and detection.

4.2.2 Refine societal and human impacts in the model

A common view of human behaviour in situations of crisis is that of disorder and chaos; people lose their humanity and resort to their basic instincts. Many studies have shown that this is in fact untrue. A view developed in the 1950s and 1960s by (Ikle, Quarantelli, Rayner, & Withey, 1957) pushed forward that rather than chaos, situations of crisis translate into a-social collective behaviours, where people tend to their own needs. This view changed in the 1980s and 1990s due to Norris Johnson and others (Keating, 1982). They found that people in dire situations continued to act as individuals embedded in social groups, with hierarchies, roles, bonds and concerns. Furthermore, even in normal situations, people tend to move in small groups and stand in semi-circles with people they know. This is something that models should consider, as it effects reactions in cases of crisis. On top of this, a lot of studies demonstrate that people when first faced with danger tend to misunderstand the signs of hazard, thus showing a sort of normalcy bias, which means people do not react instantly to an attack.

These are just some examples of factors that affect crowd movements in hazardous situations. Therefore, for the SATIE agent-based model to be accurate, it needs to consider crowd movements and individual human behaviours, which affects how crowds move both in normal situations and situations of crisis. The following factors, which were selected based on their relevancy for SATIE, thus affect crowd movements:

- Personal traits: Age, sex, physical capacities, relationships and individual thinking processes vary greatly from individual to individual, which affects their capacity to respond to a situation. Indeed, age and health affect speed and breathing rates for instance. In fact, many will apprehend evacuation based on their own assessment of their physical capacity and tasks to be carried out, whilst many also only begin acting when groups begin to form and adapt their behaviour.
- Season: Depending on the season, people were different types of clothing, and heavy winter clothing influences walking speeds and therefore assessments of space.
- Knowledge of the building: Many people present in an airport are not aware of the layout of it. Indeed, airports have many spaces that are separated from one another, and travellers mostly only seldom go through airports. Hence, they do not have a deep knowledge of the building as staff working at the airports do, which affects their capacity to take the shortest route to an exit.
- Affluence and human density: The size of groups and thus the affluence at the airport is an important factor in the capacity to respond to the crisis; bigger groups move more slowly and will take longer to arrive to a consensus on what to do. However, critical mass theory (Marwell & Oliver, 1993) purports that the greater the size and heterogeneity of a crowd, the more likely it is to have individuals with the skills require to survive the crisis.
- Norms and social relations: Relationship between individuals seems to continue in situations of crisis. Hence, individuals do not abandon family members or friends, on the contrary, studies show that in the midst of crisis, evacuees often re-enter hazardous zones to help others (Johnson, Feinberg, & Johnston, 1994). Many also begin helping strangers, thus re-arranging social norms. This falls in line with the Emergent Norm Theory (ENT), which states that collective behaviours and social norms shift in abnormal situations (Scott & Drury, 2000). Power statuses and relationships are re-arranged to enable maximum survival. Following this theory, simulation models should conceptualize the emergence of leadership in situations of crisis.
- Location in the crisis: (Johnson N. , 1987) purports that awareness of an event is not the same depending on where one is located, which will affect the reaction to a situation. Consequently, communication about an event normally occurs from front to back of a gathering, which will also translate into the individuals being closer to an event to be more active in trying to tackle the event. Added to this is the issue that simultaneous conducts occur during an event, which affects group reactions. For instance, some may be seeking to exit the airport after an issue occurred, whilst others who are unaware of the issue seek to enter the airport. Hence, an impact model needs to consider the varying levels of awareness of an event depending on the location of agents.
- Presence of staff: The presence of personnel and especially evacuation management personnel greatly affects responses to crisis, as they are able to guide groups towards shortest or familiar paths to safety, rather than more random paths that individuals with no awareness of the airport might take.

Thus, there are a plethora of socially relevant factors that affect how individuals react to crises. These can be individual factors such as age, physical strength, but also individual levels of alertness and awareness, along with views of social roles, or they can be group factors, such as the re-organization of social bonds, the size of groups, etc. Most models thus lack these psychological and social dimensions that affect responses to attacks. In order to refine its model, SATIE should take the above-mentioned factors into account.

4.2.3 Interaction of the Impact Propagation Simulation with the Incident Management Portal

The Incident Management Portal will receive alerts from different SATIE systems. It is the main tool for the SOC operator to detect a possible threat.

When an alert arrives in the Incident Management Portal, the alert can be automatically classified as incident, or it needs a classification by a SOC operator. The analysis of impact will help the SOC operator to take a decision. The Incident Management Portal will send the alert with the information of the asset involved as soon as it arrives in the system. In response, the Impact Propagation Simulation can update the alert and potentially update the severity. If the alert receives an update from another system or is classified as incident, an update of the impact can be made by resending it to the Impact Propagation Simulation. A SOC operator can also request on demand the Impact Propagation Simulation to have more information about the impacted assets. The report has to be easily understandable and a link to the Impact Propagation Simulation can be provided to allow the SOC operator to go deeper in his analysis.

When an attack happens, different alerts and incidents can be raised, with potentially different affected assets. These relations between the alerts and incidents can have a direct effect on the impact.

When an incident is raised it will be forward to the Crisis Alerting System.

4.2.4 Interaction of the Impact Propagation Simulation with the Crisis Alerting System

The Crisis Alerting System (CAS) will act as a bridge among the airports SOC and the airport's first responders, stakeholders and citizens that are using the airport facilities. The functionalities that will be provided by the CAS are grouped on two main categories:

1. **Generation of the operational picture.** CAS will combine information provided by the security and safety systems of the airport, the Incident Management Portal and the Impact Propagation Simulation in order to generate and provide to the responders the operational picture.
2. **Smart notification and alerting service.** CAS will use smart notification and alerting services in order to enable the information sharing among involved actors at every level of coordination during a crisis.

The main SATIE components that will feed CAS with information are: The Incident Management Portal and the Impact Propagation Simulation. Incident Management Portal will feed CAS with information that is related to the current list of incidents, whereas the Impact Propagation Simulation will feed CAS with information that is related to possible damages produced by the specific list of incidents.

In more details, the Impact Propagation Simulation will create and forward information that estimates the possible damage a specific incident will produce to the list of airport's assets. This information will be received by the CAS and will be combined with the rest information received by the rest components in order to generate the operational picture. This will be depicted on maps informing actors about the evolution of a specific incident and its possible damages. Moreover, the Impact Propagation Simulation will be able to enable specific response and mitigation strategies in order to calculate its effectiveness on a specific incident and the possible variations of the incident's evolution. These results will be available to the CAS, guiding the actors and responders to better organize their response activities. As a result, the information that will be provided by the impact propagation module will be really useful to the actors and responders in order to organize their response activities (that are related to a specific incident) and improve their effectiveness.

5 Conclusion

In this deliverable, the Impact Propagation Model is described. This model is needed for the Impact Propagation Simulation that will be the main goal of Task 5.1. Further, the model was related to other tools and tasks in the project such as the risk assessment in Task 2.2, the Business Impact Assessment, the Incident Management Portal and the Crisis Alerting System. The interrelations between the tools will be further defined in Task 5.1.

The model generation process, the information it is based on and its further employment has been put into context of resilience management. This enabled a conclusive presentation of the steps that were followed to build the model and a comprehensive outlook how the model will be implemented later in Task 5.1. The quantification of airport resilience during and after a crisis event considering impact propagation and cascading effects is the goal the model has been developed towards.

The required information to build the model was collected in an iterative process which means that several rounds of discussions with the end-users were needed to align the inputs with the requests. Even if some information is presented for all scenarios, the focus of the gathered information is on Scenario #1 and #2 which was provided by AIA because the Impact Propagation Simulation is involved in the demonstrations in Athens. Information was partly acquired for asset interrelations, layout maps, recovery times and KPIs. The information that is still missing is the capacities with respect to flow layers and mitigation options. This missing information will be acquired in Task 5.1 when the model is implemented and the requirements are better defined. Specific information about the impact of incidents on the BHS and the ATM were also collected to better understand the impacts and potential cascading effects beyond system boundaries.

The Impact Propagation Model is a hybrid model consisting of a layer model, a network model and an agent-based model to represent the airport processes in an abstract way. The goal of this model and the corresponding implementation is to establish a simulation environment that enables to calculate predictions about the impact propagation and potential cascading effects between the systems of the airport infrastructure. When an incident is reported, the impact on other assets belonging to various systems and flow layers is calculated using the information collected about asset properties such as asset type, location and capacities. The tool will enable to compare mitigation options that still need to be defined with the end-users based on the scenarios.

The business impact assessment has a specific focus on cyber threats. How and if the tools (Impact Propagation Simulation and Business Impact Assessment) exchange information during threat simulations still needs to be defined.

Finally, the model was reviewed and the interfaces of the planned simulation environment with the other adjacent tools of the simulation platform were discussed. Additional suggestions were provided for the agent-based model to refine societal and human aspects. The suggested additions and modifications will be implemented in Task 5.1.

6 References

2018 Gartner Magic Quadrant for Intrusion Detection and Prevention Systems (IDPS) | Alert Logic. (2019, February 26). Retrieved from https://www.alertlogic.com/resources/industry-reports/intrusion-detection-and-prevention-systems-providers/

(2019, November 26). Retrieved from Moloch: http://molo.ch

A. de Barros Barreto, P. C. (2014). Cyber-Argus: Modeling C2 Impacts of Cyber Attacks. *GEORGE MASON UNIV FAIRFAX VA CENTER FOR EXCELLENCE IN COMMAND CONTROL*.

A. Kott, J. L. (2017). Assessing Mission Impact of Cyberattacks: Toward a Model-Driven Paradigm. *IEEE Secur. Priv., vol. 15, no. 5*, 65–74.

A. Motzek, R. M. (2015). Probabilistic mission impact assessment based on widespread local events. *Assess. Mission Impact Cyberattacks*, 1.

Aalst, W. V. (2011). Process mining: discovery, conformance and enhancement of business processes. *vol. 2. Springer*.

al., H. C. (2018). Detecting Attacks Against Robotic Vehicles: A Control Invariant Approach. *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 801–816.

al., H. C. (2018). Detecting Attacks Against Robotic Vehicles: A Control Invariant Approach. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 801–816.

Athens International Airport. (n.d.). *ATHENS CITY MAP - ATHENS METRO MAP - ATHENS AIRPORT MAP*. Retrieved from http://www.athensflights.gr/index_maps.htm

Balmas, F. (2004). Displaying dependence graphs: a hierarchical approach. *J. Softw. Maint. Evol. Res. Pract., vol. 16, no. 3*, 151–185.

Broumi, S., Bakal, A., Talea, M., Smarandache, F., & Vladreanu, L. (2016). Applying Dijkstra algorithm for solving neutrosophic shortest path problems. *IEEE*, pp. 412-416.

C. Liu, A. S. (2017). A layered graphical model for mission attack impact analysis. *2017 IEEE Conference on Communications and Network Security (CNS)*, 602–609.

C. M. Ahmed, J. Z. (2018). Noise Matters: Using Sensor and Process Noise Fingerprint to Detect Stealthy Cyber Attacks and Authenticate sensors in CPS. *Proceedings of the 34th Annual Computer Security Applications Conference*, 566–581.

C. Ten, C. L. (2007). Vulnerability Assessment of Cybersecurity for SCADA Systems Using Attack Trees. *2007 IEEE Power Engineering Society General Meeting*, 1–8.

C. Ten, C. L. (2007). Vulnerability Assessment of Cybersecurity for SCADA Systems Using Attack Trees. *IEEE Power Engineering Society General Meeting*, 1–8.

C.-W. Ten, C.-C. L. (2008). Vulnerability assessment of cybersecurity for SCADA systems. *IEEE Trans. Power Syst., vol. 23, no. 4*, 1836–1846.

CBS News. (2014, September 27). Retrieved from https://www.cbsnews.com/news/chicago-air-traffic-halted-over-fire-at-faa-facility

Chakraborty, M., Chowdhury, S., Chakraborty, J., Mehera, R., & Pal, R. K. (n.d.). Algorithms for generating all possible spanning trees of a simple undirected connected graph: an extensive review. *Complex & Intelligent Systems*, pp. 265-281.

CVE - Common Vulnerabilities and Exposures (CVE). (2019, November 26). Retrieved from https://cve.mitre.org/

D. Formby, P. S. (2016). Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems. *NDSS*.

Dam, K. H., Nikolic, I., & Lukszo, Z. (2013). *Agent-based Modelling of Socio-Technical Systems.* Springer.

DefectDojo | CI/CD and DevSecOps Automation. (2019, November 26). Retrieved from https://www.defectdojo.org/

Demsar, U., Spatenkova, O., & Virrantaus, K. (2008). Indentifying critical locations in a spatial network with graph theory. *Transactions in GIS* , pp. 61-82.

Domenico, M. D., Granell, C., Porter, M. A., & Arenas, A. (2016). The physics of spreading processes in multilayer networks. *Nature Physics 12.10*.

Dorri, A., Kanhere, S. S., & Jurdak, R. (2016). Multi-Agent System: A survey. *IEEE Access*.

Download Nessus Vulnerability Assessment | Tenable®. (2019, November 26). Retrieved from https://www.tenable.com/products/nessus

DW. (2018, December 29). Retrieved from https://www.dw.com/en/hanover-airport-shut-down-over-drug-fueled-security-incident/a-46895744

Eldosouky, A. R., Saad, W., & Mandayam, N. (2017). Resilient Critical Infrastructure: Bayesian Network Analysis and Contract-Based Optimization. *arXiv preprint*, p. arXiv:1709.0030.

Enterprise Intrusion Prevention (IPS) Software & Solutions. (2019, December 8). Retrieved from https://www.trendmicro.com/en_us/business/products/network/intrusion-prevention.html

Fingerprinting electronic control units for vehicle intrusion detection. (2016). *USENIX Security Symposium* , 911–927.

Furno, A., Faouzi, N.-E. E., Sharma, R., Cammarota, V., & Zimeo, E. (2018). A Graph-Based Framework for Real-Time Vulnerability Assessment of Road Networks. *International Conference on Smart Computing (SMARTCOMP)* (pp. 234-241). IEEE.

Gao, X., Xiao, B., Tao, D., & Li, X. (2010). A survey of graph edit distances. *Pattern analysis and applications*, pp. 113-129.

Green, O., & Bader, D. A. (2013). Faster clustering coefficient using vertex covers. *IEEE International Conference on Social Computing*, pp. 321-330.

Häring, I., Sansavini, G., Bellini, E., Martyn, N., Kovalenko, T., Kitsak, M., . . . Linkov, I. (2017). Towards a Generic Resilience Management Quantification and Development Process: General Definitions, Requirements, Methods, Techniques and Measures, and Case Studies. In *Resilience and risk* (pp. 21-80). Springer Dordrecht.

Hasan, S., & Foliente, G. (2015). Modeling infrastructure system interdependencies and socioeconomic impacts of failire in extreme events: emerging R&D challenges. *Nat Hazards*, pp. 78:2143-2168.

Hellwig, A., & Volkmann, L. (2008). Maximally edge-connected and vertex-connected graphs and digraphs: A survey. *Discrete Mathematics*, pp. 3265-3296.

Hernandez, J., & Mieghem, P. V. (2015). Classification of graph metrics.

Hiermaier, S., Hasenstein, S., & Faist, K. (2017). Resilience Engineering - how to handle the unexpected. *7th REA Symposium*.

History. (2019, September 11). Retrieved from https://www.history.com/topics/21st-century/9-11-attacks

Holmgren, A. J. (2006). Using graph models to analyze the vulnerabilty of electric power networks. *Risk analysis*, pp. 955-969.

Huth, M. K. (2018). Scission: Signal Characteristic-Based Sender Identification and Intrusion Detection in Automotive Networks. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 787–800.

IBM QRadar Risk Manager. (2019, November 26). Retrieved from www.ibm.com/support/knowledgecenter/en/ss42vs_7.3.2/com.ibm.qradar.doc/c_qrm_ug_overview.html

Ikle, F. C., Quarantelli, E. L., Rayner, J. F., & Withey, S. B. (1957). Withdrawal Behavior in Disasters: Escape, Flight and Evacuation Movements. . *Washington, D.C. National Academy of Sciences - national Research Counclin, Division of Anthropology and Psychology Committee on Disaster Studies*.

INOV / BP-IDS – Business Process Intrusion Detection System. (2019, February 24). Retrieved from http://www.bp-ids.com/

International Organization for Standardization. (2018). ISO 31000 Risk management. Genf.

J. R. Goodall, A. D. (2019). Camus: Automatically mapping Cyber Assets to Missions and Users. *MILCOM 2009 - 2009 IEEE Military Communications Conference*, 1-7.

Jakobson, G. (2011). Mission cyber security situation assessment using impact dependency graphs. *14th International Conference on Information Fusion*, 1–8.

Johnson, N. (1987). Panic at 'The Who' concert 'stampede': an empirical assessment. *Social Problems*, 362-373.

Johnson, N. R., Feinberg, W. E., & Johnston, D. M. (1994). Microstructure and Panic: The impact of Social Bonds on Individual Action in Collective Flight from the Beverly Hills Supper Club Fire. In R. Dynes, & K. Tierney, *Disasters, Collective Behavior and Social Organization* (pp. 168-189). Newark, Delaware: University of Delaware Press.

Joint Task Force Transformation Initiative. (2012). *Guide for conducting risk assessments.* National Institute of Standards and Technology.

Kamada, T., & Satoru, K. (1989, April 12). An algorithm for drawign general undirected graphs. *Information Processing Letters, 31*(1), 7-15.

Keating, J. P. (1982). The Myth of Panic. *Fire Journal*, pp. 56-61.

Kroese, D. P., Brereton, T., Tamire, T., & Botev, Z. I. (2014). Why the Monte Carlo method is so important today. *Wiley Interdisceplinary Reviews: Computational Statistics*, 386-392.

L. Cheng, K. T. (2017). Orpheus: Enforcing cyber-physical execution semantics to defend against data-oriented attacks. *Proceedings of the 33rd Annual Computer Security Applications Conference*, 315–326.

LTBA-ISTANBUL. (2019, November 22). *25-MAY-2011 FAKE ATC IN ACTION*. Retrieved from https://www.liveatc.net/recordings.php

Mamun, A. A., & Rajasekaran, S. (2016). An efficient minimum spanning tree algorithm. *IEEE Sympsium on Computers and Communication (ISCC)*, pp. 1047-1052.

Mamun, A.-A., & Rajasekaran, S. (2016). An efficient minimum spanning tree algorithm. *IEEE Symposium on Computers and Communication (ISCC)*, pp. 1047-1052.

Marwell, G., & Oliver, P. (1993). *The Critical Mass in Collective Action. A Micro-Social Theory.* London: Cambridge University Press.

N. Friedman, D. G. (1997). Bayesian network classifiers. *Mach. Learn., vol. 29, no. 2–3*, 131–163.

Newman, M. E. (2002). Assoritve mixing in networks. *Physical review letters*, p. 208701.

Nikto2 | CIRT.net. (2019, November 26). Retrieved from https://cirt.net/Nikto2

NVD - Vulnerabilities. (2019, November 26). Retrieved from https://nvd.nist.gov/vuln

Oliva, G., Panieri, S., & Setola, R. (2012). Modeling and simulation of critical infrastructures. *WIT Transactions on State-of-the-art in Science and Engineering 54*.

OpenVAS - OpenVAS - Open Vulnerability Assessment Scanner. (2019, November 26). Retrieved from http://openvas.org/

Ouyang, M. (2014). Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability Engineering and System Safety*, pp. 121:43-60.

OWASP Threat Dragon - OWASP. (2019, November 26). Retrieved from https://www.owasp.org/index.php/OWASP_Threat_Dragon

P. A. Porras, M. W. (2002). A mission-impact-based approach to INFOSEC alarm correlation. *International Workshop on Recent Advances in Intrusion Detection*, 95–114.

R. Bray, D. C. (2008). *OSSEC host-based intrusion detection guide.* Syngress.

Rinderle-Ma, K. B. (2016). Automatic Signature Generation for Anomaly Detection in Business Process Instance Data. *Enterprise, Business-Process and Information Systems Modeling, Cham*, 196–211.

Rinderle-Ma, K. B. (2016). Automatic Signature Generation for Anomaly Detection in Business Process Instance Data. *Enterprise, Business-Process and Information Systems Modeling, Cham*, 196–211.

Roesch, M. (1999). *Snort: Lightweight intrusion detection for networks.* Lisa.

S. Jajodia, S. N. (2011). Cauldron mission-centric cyber situational awareness with defense in depth. *2011 - MILCOM 2011 Military Communications Conference*, 1339–1344.

S. Noel, E. H. (2016). 'Chapter 4 - CyGraph: Graph-Based Analytics and Visualization for Cybersecurity'. In V. V. V. N. Gudivada, *Handbook of Statistics* (pp. 117–167). Elsevier.

SATIE project. (2019). *DX.X - name of the SATIE deliverable you want to reference.*

Scott, C., & Drury, J. (2000). Crowds, context and identity: Dynamic categorization processes in the 'poll tay riot'. *Human relations*, 247-273.

Shin, K.-T. C. (2016). Fingerprinting electronic control units for vehicle intrusion detection. *25th USENIX Security Symposium USENIX Security 16*, 911–927.

SiLK. (2019, November 26). Retrieved from https://tools.netsa.cert.org/silk/

Solomon, J. D., & Sharpe, A. (2016). Infrastructure systems renewal and replacement model using probabilistic forecasting. In *2016 Annual Reliability and Maintainability Symposium (RAMS)* (pp. 1-6). Tucson, AZ. doi:10.1109/RAMS.2016.7448032

Standish, R. K. (2008). Concept and definition of complexity. *Intelligent complex adaptive systems*, pp. 105-124.

Stroeve, S., Blom, H. A., & Bakker, G. B. (2009). Systemic accident risk assessment in air traffic by Monte Carlo simulation. *Safety science*, 238-249.

T., J. T. (2012). *Guide for conducting risk assessments.* National Institute of Standards and Technology.

Temin, S. M. (2015). A Cyber Mission Impact assessment tool. *IEEE International Symposium on Technologies for Homeland Security (HST)*, 1-7.

Temin, S. M. (2015). A Cyber Mission Impact assessment tool. *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, 1–7.

The Telegraph. (2016, November 8). Retrieved from https://www.telegraph.co.uk/travel/destinations/oceania/australia/articles/hoax-caller-impersonating-air-traffic-control-forces-pilot-to-abort-landing/

ThreadFix Vulnerability Management Platform. (2019, November 26). Retrieved from https://threadfix.it/

Tiwari, A. (2018, September 20). *Archery - Vulnerability Assessment and Management Tool*. Retrieved from Medium: https://medium.com/archerysec/archery-vulnerability-assessment-and-management-tool-ecbf5e92f717

Urbina, D. I. (2016). Limiting the impact of stealthy attacks on industrial control systems. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 1092–1105.

V. Paxson, S. C. (2006). *Bro intrusion detection system.* Lawrence Berkeley National Laboratory.

W. Aoudi, M. I. (2018). Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 817–831.

What's New with Microsoft Threat Modeling Tool 2016. (2015, October 8). Retrieved from Microsoft Security: https://www.microsoft.com/security/blog/2015/10/07/whats-new-with-microsoft-threat-modeling-tool-2016/

Wolfgang, M. (2002). Host Discovery with nmap. *Explor. Nmaps Default Behav.*, 16.

Wooldridge, M. (2009). *An introduction to multiagent systems.* John Wiley & Sons.

Y. Chen, C. M. (2018). Learning from mutants: using code mutation to learn and monitor invariants of a cyber-physical system. *IEEE Symposium on Security and Privacy (SP)*, 648–660.

Y. Yang, Z. C. (2018). Probabilistically Inferring Attack Ramifications Using Temporal Dependence Network. *IEEE Trans. Inf. Forensics Secur., vol. 13, no. 11*, 2913–2928.

Yan, E., & Ding, Y. (2009). Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the American Society for Information Science and Technology*, pp. 2107-2118.

Yang, B. J. (2008). VTAC: Virtual terrain assisted impact assessment for cyber attacks. *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, vol. 6973*.

Zager, L. A., & Verghese, G. C. (2008). Graph similarity scoring and matching. *Applied mathematics letter*, pp. 86-94.

ZAP. (2019, November 26). Retrieved from https://www.zaproxy.org/

Zhang, P., Peeta, S., & Friesz, T. (2005). Dynamic game theoretic model of multi-layer infrastructure networks. *Networks and Spatial Economics 5(2)*, 147-178.