BP-IDS - Attack Impact Assessment

Olga Sofia Berens de Carvalho

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Dr. Carlos Nuno da Cruz Ribeiro
Eng. Nelson Nobre Escravana

Examination Committee

Chairperson: Prof. Dra. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. Dr. Carlos Nuno da Cruz Ribeiro
Members of the Committee: Prof. Dr. António Manuel Raminhos Cordeiro Grilo

September 2020

Declaration

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgements

First and foremost, I would like to thank Prof. Dr. Carlos Ribeiro for introducing me to the world of cybersecurity where I finally found my purpose. Thank you for giving me the incredible opportunity to work towards it, by bringing me in contact with interesting problems and the brilliant people at INOV.

Here, my deepest gratitude goes to my co-supervisors, Eng. Nelson Escravana and Eng. Filipe Apolinário. I thank you for all your guidance, boundless patience, and inspiring drive to pursue the best work possible, while making the most out of it. Without your contribution this work would not be possible.

A special thanks to the entire team at INOV for their companionship and friendship, which made the days working on this project so enjoyable.

Finally, I thank my loving family, for the unconditional support that carries me in all aspects of my life. Thank you for all your sacrifice, care and love that forged the person I am today. This accomplishment is as much mine as it is yours.

Last, but definitely not least, I would like to thank my amazing partner for the endless support, tireless brainstorming sessions, and optimistic encouragement that you never seem to run out of. I truly appreciate it.

Abstract

As adversaries continue to develop new attack techniques to undermine organizations' business goals, there is an increase necessity for defenders to understand how a cyber-incident can impact those goals. Since widely implemented security mechanisms generally focus on low-level events and report them independently, system defenders are left with the high specialized and labour-intensive task of filtering those events to analyse what has been compromised (and what could be) to respond in a timely fashion and accurately minimizing the impact of the incident on the organization.

As organizations' resources and goals are growing more dependent on information and communications technology, that analysis becomes even so more complex, which has motivated research in mission impact assessment (MIA) to estimate the impact a cyber-incident can have on the organization's goal (i.e. mission).

Driven by this objective, this work explores relevant contributions in the subject to synthetize the key requirements to perform MIA. As a result, this dissertation proposes the *Business Impact Assessment* (BIA) methodology. BIA was developed to offer a mission-oriented evaluation model to profile the organization, and, upon it, a simulation platform to simulate the impact on the mission of a user-chosen exploited threat. A series of experiments were undergone to test BIA's efficacy of performing MIA under different case-studies developed to mimic expected dynamics in the organization's infrastructure and goals. The results have shown BIA is successful in generating a relevant report on mission impact, that allows the user to identify situations of risk of impact.

Keywords: Impact Assessment, Mission Impact, Evaluation Model, Business Process Modelling, Impact Simulation, Cyber-threats, Threat Impact, Security

Resumo

Com a constante evolução de novas técnicas de ciberataques que visam inviabilizar os objectivos-negócio das organizações, cresce a necessidade por parte dos seus defensores de perceber como ciber-incidentes podem impactar esses objectivos. Como as actuais ferramentas de segurança geralmente focam-se em eventos de baixo nível e reportam-nos independentemente, a tarefa altamente especializada e intensiva de triagem desses eventos, para analisar o que foi comprometido (e o que pode vir a ser), é exclusivamente deixada para o defensor do sistema, para que consiga responder de forma rápida e precisa, de modo a minimizar o impacto.

A crescente dependência dos activos e objectivos-negócio das organizações em tecnologias de informação e comunicação tornam essa análise ainda mais complexa, o que motiva investigação na avaliação do impacto na missão (MIA), que tem como objectivo estimar o impacto que um ciber-incidente pode ter nos objectivos (i.e., missão) da organização.

Conduzido por esse propósito, este trabalho explora contribuições relevantes no tema de forma a sintetizar os principais requisitos que se deve ter em conta quando se avalia o impacto na missão. Como resultado, esta tese propõe a metodologia *Business Impact Assessment* (BIA). BIA foi desenvolvido para oferecer um modelo de avaliação de impacto orientado à missão para representar a organização e, mediante esse modelo, uma plataforma de simulação para simular o impacto na missão de uma ciber-ameaça escolhida pelo utilizador. Uma série de experiências foi realizada para testar a eficácia de BIA sob diferentes casos-de-estudo desenvolvidos para representar dinâmicas realistas na infra-estrutura e objectivos duma organização. Os resultados mostram que BIA produz com sucesso um relatório com informação relevante sobre o impacto na missão, que permite ao utilizador identificar situações de risco de impacto.

Palavras-chave: Avaliação do impacto, Impacto na missão, Modelo de avaliação, Modelação de processos-negócio, Simulação do impacto, Ciber-ameaças, Impacto de ameaças, Segurança

Table of Contents

List of Figures

List of Listings

List of Tables

List of Acronyms

AP	Access Point
API	Application Programming Interface
BIA	Business Impact Assessment
BN	Bayesian Network
BP	Business-process
BP-IDS	Business Process Intrusion Detection System
BYOD	Bring-Your-Own-Device
CB	Circuit Breaker
CIA	Confidentiality, Integrity and Availability
DoS	Denial of Service
HMI	Human-Machine Interface
ICMP	Internet Control Message Protocol
ICS	Industrial and Control System
ICT	Information and Communication Technology
IDS	Intrusion Detection System
IED	Intelligent Electronic Device
IT	Information Technology
MIA	Mission Impact Assessment
MMS	Manufacturing Message Specification
MulVAL	Multihost, Multistage Vulnerability Analysis
PLC	Programmable Logic Controller
SCADA	Supervisory Control And Data Acquisition
SCADA WS	SCADA workstation
SIEM	Security information and Event Management
SW	Switch
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

1. Introduction

1.1. Motivation

The cyberspace is built to access and share information via information and communications technology (ICT) [1], which constant progress opens up the cyberspace to new illicit activities, as adversaries continue to evolve new attack types, tools and techniques to penetrate more complex and well-controlled environments, evading common defences and produce increased damage [2].

Most modern organizations have ICT embedded into the core of their business-processes, as a means to increase their operational efficiency, exploit automation and/or improve decision quality. An attack to the ICT infrastructure of an organization could significantly impact the business-objectives they support. This can be clearly observed when the organization under attack is an essential services provider (such as transportation, energy supply or distribution), and the impact of undermining the security and viability of its business-processes can disrupt the normal functioning of the societies they provide for [3]. The Stuxnet [4], BlackEnergy and Industroyer [5] malware are three notorious examples. In 2011, the worm Stuxnet, targeted Siemens programmable logic controllers (PLCs) by exploiting unpatched Windows vulnerabilities and was able to shut down an Iranian uranium enrichment facility, while the more recent BlackEnergy (2015) and Industroyer (2016) malware targeted the Ukrainian power grid and caused a power outage during its characteristic mid-December cold weather: the former affected roughly 225,000 households for over six hours, whereas the latter, a year later, was able to deprive part of Ukraine's capital of power for an hour.

On the one hand, leaving vulnerabilities unattended may indeed lead to significant damage; on the other hand removing all vulnerabilities of a system is usually impractical [6]. Despite robust defensive measures, inevitably organizations will have to deal with a cyber-incident. Automated detections tools, designed to raise incident-related alerts when suspicious activity is detected, are a widely available and used security solution, such as antivirus software, log analysers and intrusion-detection systems (IDS) [7]. Security's weaknesses may also be discovered through manual means, such as searches on publicly available security information on vulnerabilities, exploits and attacks, and issues reported by the organization's own users.

While these mechanisms aim to improve the security of the organization's resources, with varying levels of detail and accuracy, they generally focus on low-level events and report them independently, which leaves to the system defender the entire decision-making process of determining, in a timely fashion, what has been compromised, what could be compromised and whether the incident has any current or future negative impact on the organization's monitored network and goals, and to respond quickly and accurately to minimize the impact [8]. This analysis is additionally challenging as a result of four factors:

- automated detection tools often overwhelm the system defenders with a large volume of alerts – for instance, in a typical organization, an IDS can raise thousands or even millions incident alerts per day [9] [10] resulting in an almost constant alert situation;

- incidents detected by detection tools, or by manual means, are not guaranteed to be accurate – for example, an IDS may produce false positives or even false negatives [11] or a vulnerability found from publicly available information may not be concretized by the organization's configuration;
- the limited view of the roles the monitored organization's resources play in the overall organization's goal makes it difficult to accurately prioritise and assign resources to perform incident response [12] – for instance, when looking at Stuxnet, detecting an exploit of a Windows vulnerability is not enough to identify the possible impact on the uranium enrichment plant, which in fact was impacted and shut down;
- deep and specialized technical knowledge is necessary for proper and efficient analysis of incident-related data [9] – for example, and IDS alert has little contextual information beyond its identifier and an IDS alert description [3], from which a non-expert user may not be able to grasp its impact scope or urgency.

The aforementioned challenges have motivated recent work from both commercial and government/military sectors [13], in *mission impact assessment* (MIA) [14], which tries to estimate how a cyber-incident can impact the goal of an organization (i.e. mission) to reduce incident responders workload and provide a higher-level of situational awareness. The MIA process may be a part of different cyber practices: from a prevention perspective, when a threat is identified during a *risk analysis*, its impact is a critical analysis item; from a detection viewpoint, during *incident handling*, the relevance of the incident can be assessed by its mission impact; from a response perspective, during *event monitorization*, an event generated by security monitoring systems or a finding obtained in a *vulnerability analysis* should be subject to MIA to proper validate and prioritise its importance.

Independently of the practice, MIA typically requires a great level of detailed knowledge about the organization under assessment, including the organization's mission and the organization's cyber infrastructure, consisting of all organization's ICT resources that carry out the mission, and how they interact, condition and depend on each other, which is often difficult to obtain.

1.2. Contributions

This dissertation proposes a MIA solution that shifts the attention from the reactive incident-oriented security approach employed by detection tools that actively needs to consume real life security configurations and cyber events, which can include false positives, false negatives and thousands of events to process, to a more proactive risk-oriented approach, that can be performed at any point to pre-emptively identify the impact of different compromised entry-points, on different organizational configurations and different threat landscapes.

Grounded in the idea that the needed MIA data does exist in digital format, but in disparate locations and formats, the main contributions of this work can be summarized as follows:

1) Based on a multi-layered information structure, a mission-oriented evaluation model is put forward. To populate this model, existing tools are used to gather knowledge about the organization's infrastructure and mission in a semi-automatic manner, offering an intuitive model easy to configure by a non-expert end-user.

2) Based on the evaluation model, an incident propagation simulation platform was designed accordingly, and a bottom-up computation methodology is proposed to detect the business-processes that are potentially impacted by the simulation of desired threat landscapes.
3) The evaluation model and simulation platform are implemented and combined into a single tool, *Business Impact Assessment* (BIA). BIA was verified by different case-studies and in a real target test environment. The experiments results show that BIA can successfully generate a report on mission impact that gives an overview of situations of risk.

1.3. Structure of the Document

The remainder of this thesis is structured as follows: Section 2 focus on a literature review of research on the subject of MIA. An overview of current contributions for impact modelling is provided, along with current approaches for impact propagation and measurement; Section 3 covers the design of BIA approach while Section 4 describes the implementation process; Section 5 applies BIA to a power supply testbed and covers the evaluation of its functionality and results; Section 6 concludes the thesis, by summarizing up the main contributions and results of the work, as well as future work directions.

2. Literature Review

In a MIA context, as previously stated, the organization’s goal is commonly referenced as the organization’s *mission*, inspired by its literal meaning in MIA research conducted by the military sector. In a broader sense of the word, *mission* is also adopted by the commercial sector where the organization’s goal is related to its business goals.

In essence, the organization’s mission can be decomposed in a collection of explicitly defined business-objectives to be achieved, which, in turn, will depend on one or more of the organization’s resources. To understand how a cyber-incident on the organization’s resources can impact the mission, research in MIA typically follows three main stages (Figure 1): (1) the *modelling* stage that aims to discover and model all the organization’s entities involved in accomplish the organization’s mission, and the dependencies among them ([12], [15]–[29]), (2) the *propagation* stage, to assess how the impact may propagate through those modelled entities and compromise the organization’s mission ([3], [15]–[20], [24]–[27], [28]–[34]) and (3) the *measurement* stage, where metrics are integrated within the model to numerically evaluate the impact on the mission ([3], [15], [17], [19], [22], [23], [29]–[31], [35]–[40]).

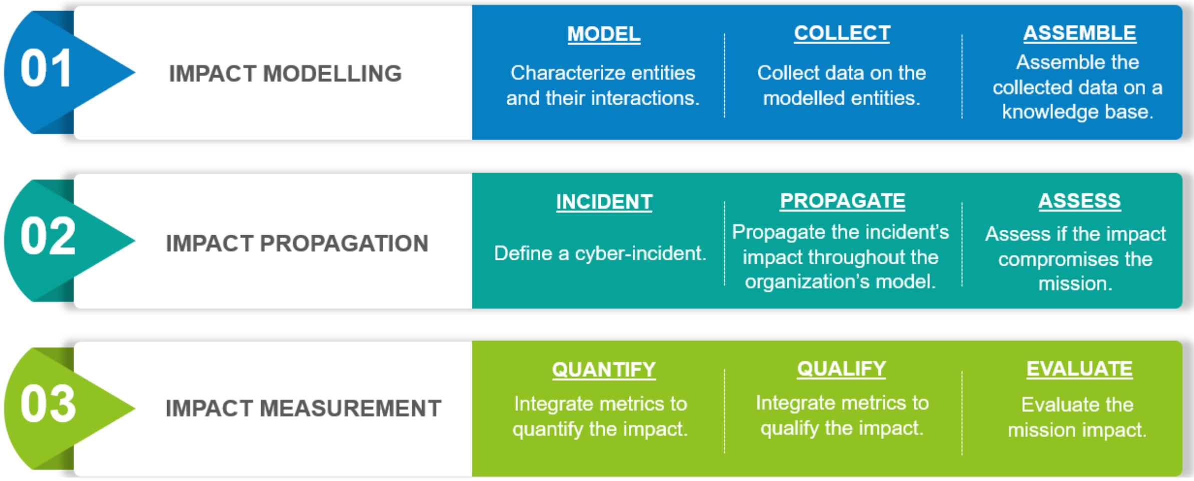

01
IMPACT MODELLING
MODEL
Characterize entities and their interactions.
COLLECT
Collect data on the modelled entities.
ASSEMBLE
Assemble the collected data on a knowledge base.
02
IMPACT PROPAGATION
INCIDENT
Define a cyber-incident.
PROPAGATE
Propagate the incident’s impact throughout the organization’s model.
ASSESS
Assess if the impact compromises the mission.
03
IMPACT MEASUREMENT
QUANTIFY
Integrate metrics to quantify the impact.
QUALIFY
Integrate metrics to qualify the impact.
EVALUATE
Evaluate the mission impact.

Figure 1 - MIA's main stages.

Following the main stages of MIA, this chapter presents a literature review of current approaches and is organized as follows: Section 2.1 explores the conceptual models used in MIA research to represent the organization’s entities and the tools used to discover information to populate those models; Section 2.2 analyses existing approaches for propagating the impact throughout the modelled organization; Section 2.3 presents current metrics to quantify the impact on the organization’s entities. Finally, Section 2.4 presents a summary and comparison of the major related studies in Table 2.

2.1. Impact Modelling

The mission is supported by a number of entities at several abstraction layers that can include, but are not limited to, an asset layer, a service layer, a business layer and a user layer. The identification and definition of all the organization’s entities involved in the organization’s mission is a technically challenging task since organizations may be comprised of networks that include hundreds or more devices, each one used for different applications, used for different goals, by different personnel. The organization's entities involved in the mission are referred by the present work as *mission performers* (or simply *performers*) and should be pro-actively defined:

- A *business-process* is an explicitly defined sequence of activities [3] required to achieve a *business-objective*. The collection of all the business-objectives represents the organization's *mission*.
- An *activity*, defined as the unit of the mission ([13], [22]), indicates the action to be taken, and it can be carried by a service running on an asset, or directly by an asset.
- A *service* is a mechanism that enables access to a set of one or more capabilities [18]. The availability of these capabilities defines which activities can be performed. For instance, services may include (but are not limited to) operating systems, middleware or applications running on assets. It is considered that services are solely dependent on the assets that carry them out [22].
- An *asset* is anything the organization holds [41] that plays a role in the mission and can be divided into several categories. In the present work four categories are distinguished:
    (1) *cyber assets*, to encompasses all the organization's information technology (IT) devices, such as routers, servers, switches, firewalls, hosts (physical or virtual computer systems connected to the organization's network), etc.;
    (2) *physical assets* [41], concerning assets related to the organization's site and physical means of operation, such as employer's badges, building, cablings, among others. Physical assets are specially contemplated by MIA in Industrial and Control Systems (ICS) to consider sensors, actuators and all equipment that interacts with physical processes.
    (3) *users*, to represent any person involved in the organization's mission [15], their accounts and online identity, and their purpose and privilege for accessing assets;
    (4) *information*, to comprise any knowledge or data that has value to the organization [41], such as vital information for the exercice of the mission or strategic information for achieving objectives [42].

Further, before continuing into the subject matter, it is important to review some cybersecurity concepts that will be used frequently. This work adopts the taxonomy provided by the International Standard ISO/IEC 27005 [42] and 27032 [41]:

- A *vulnerability* is a weakness of an asset that can be exploited by a threat [41] to compromise the asset's security policy.
- A *threat* is the potential cause of an unwanted *cyber incident*, which may result in harm to a system, individual or organization [41].
- An *information security risk*, or simply *risk*, is the potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization [42].
- The *impact* is defined as adverse change to the level of business-objectives achieved [42].

This chapter will explore the conceptual models used to represent the organization's mission performers (Section 2.1.1) and tools presently used to identify their characteristics, configurations and dependencies (Section 2.1.2).

2.1.1. MIA Models

To understand how the organization's entities relate to each other, numerous MIA studies ([15]–[17], [22]–[25], [28], [29]) employ an entity dependency graph [3] to model mission performers as abstraction layers, and the interactions and dependencies between mission performers as links between layers and among each individual layer. Some of the most relevant MIA models ([15], [22]–[25]) are reviewed and a brief analysis on the model's strengths and weaknesses is provided.

1. VTAC [15] follows that multi-layered structure by introducing the concept of a *virtual cyber terrain* (VT) (Figure 2), which models a computer network's topology, and combines it with the network's configurations and known vulnerabilities. The proposed VT is modelled as a graph, consisting of nodes that represent hosts (grey ellipses), routers (blue rectangles), or users (blue ellipses), interconnected with directed edges (edges are directed from one node to another). Each user node is connected to their accounts (blue circles), a router node represents connecting devices, such as routers or switches, and a host node represents the root of a tree of the services (orange circles) it provides. This model also incorporates a cybersecurity perspective by including cyber-incident related elements, such as IDS alerts, related to each service (yellow squares, children nodes of the services nodes). VTAC includes a rarely present *user* layer, although useful to keep track of privileges and accountability, it is a complex and time expensive process as information about mapping of users to assets quickly becomes obsolete with wireless environments and Bring-Your-Own-Device (BYOD) infrastructures becoming more popular, so it is rarely attempted [12]. Also, mapping services with related IDS alerts makes this model in need of constant updating since most enterprises can see more than 10,000 IDS alerts per day [10]. Finally, since this model does not integrate a mission abstract layer, MIA is limited to the impact at host, service, user and network-level.

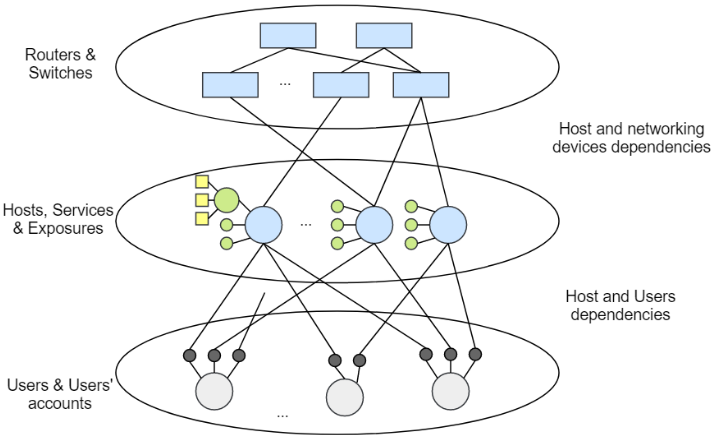

Routers & Switches
...
Host and networking devices dependencies
Hosts, Services & Exposures
...
Host and Users dependencies
Users & Users' accounts
...

Figure 2 - VTAC virtual cyber terrain (VT) model: assets (blue) comprised of network devices (blue rectangles) and hosts (blue circles); services (green); IDS alerts (yellow) and users and their accounts (grey).

2. Sharing VTAC's notion of services connected to hosts nodes, and dependencies between hosts and routers nodes, Jakobson [22] presents a layered structured called *cyber terrain* (CT). It loses the users layer, combines VTAC's hosts and routers into a more inclusive *asset* layer, individualizes services into their own layer, and adds a mission layer (Figure 3) resulting in a CT comprised of

three layers: asset, service and mission; five types of nodes, including assets (blue circles), services (green circles), activities (orange circles), missions (red circles) and logical dependency nodes (grey rectangles); and directed edges representing the interdependencies existing among nodes. In each individual layer, dependencies among same-layer nodes – *horizontal dependencies* – are depicted as edges and may represent connectivity, containment, sequential orders, and other types of relationships. Similarly, across consecutive layers there are edges portraying the dependencies between components of different layers – *vertical dependencies* – representing which component supports, and is supported by, enabling a comprehensive view of the mapping between the lower asset-layer and the higher mission-layer. The logical dependency nodes are basically *AND*-conditions and *OR*-conditions that represent logical dependencies among other nodes. The *AND*-node represents that a parent node depends on all of its children nodes. The *OR*-node denotes that a parent node depends on at least one of its children nodes. For example, a successful activity (orange node T_6), may depend on all of the supporting services being functional (green nodes S_6 and S_7), while a complete mission (red node M_2) could require only one of its activities being fulfilled (orange node T_5, or T_6 or T_7).

The CT provides a high-level reference model [40] of mission performers to model typical IT organizations. Although it does not incorporate a cybersecurity abstraction layer directly in the CT model (as VTAC did with IDS alerts), security vulnerabilities were included in a different model and mapped to the asset layer in the CT. Also, its multi-layer structure and mathematical constraint satisfaction approach (by the logical dependency nodes) can be used to easily integrate with other layers of information.

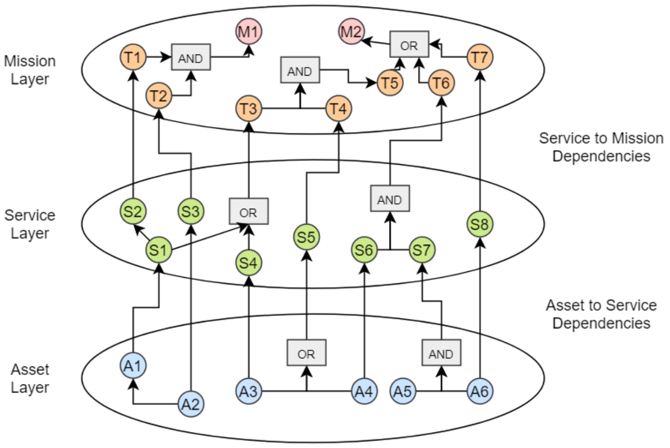


Figure 3 - Jakobson's Cyber Terrain (CT) model: assets (blue), services (green), activities (orange) and mission nodes (red).

3. The VASM [23] model takes Jakobson’s CT a step forward by integrating a cybersecurity layer *vulnerability* layer, consisting of asset’s vulnerabilities (Figure 4), represented as yellow nodes. This layer connects with the asset layer by vertical dependencies that associate vulnerable assets with the corresponding vulnerabilities: an asset can have multiple vulnerabilities, and the same vulnerability can be associated to multiple assets. Among the vulnerability nodes, VASM ‘s horizontal dependencies depict the sequential order of exploration an attacker has to follow to achieve his goal: an attacker may have to explore first vulnerabilities V_1 and V_2 to reach his target

vulnerability V_3. In this way, the VASM four-layered model is able to map vulnerabilities to the organization's mission activities at mission-level, although they do not leverage the CT's logical conditions (AND-conditions and OR-conditions) in the vulnerability layer, nor among the vertical dependencies with the asset layer. Additionally, the cybersecurity layer unit chosen to integrate in this model (vulnerabilities) may limit its efficiency since, from a risk analysis point of view, an asset can have a vulnerability, but if the same asset has no threat, then it has little to no risk of being impacted. The lack of threat information may lead to several false positive impacts when mapping a vulnerability an asset has, but it is not threatened by, to the mission activities the asset supports.

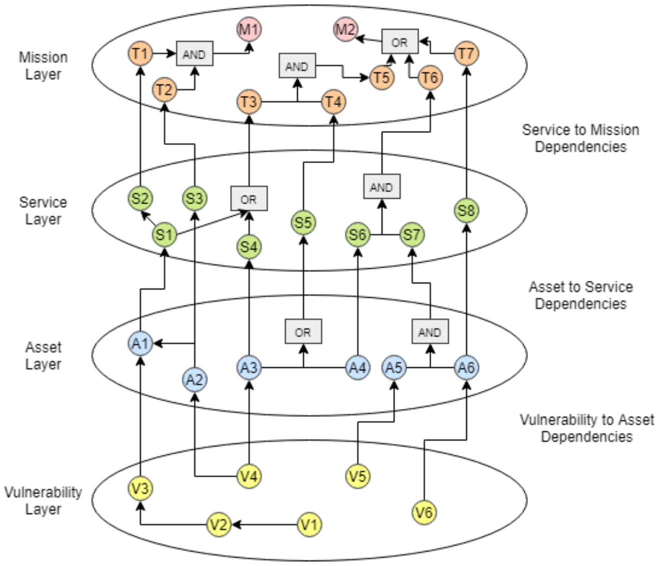


Figure 4 - VASM model: vulnerabilities (yellow), assets (blue), services (green), activities (orange) and mission nodes (red).

4. Finally, MIA research directed for ICS ([24], [25]) provide examples of models that take into account a physical layer (Figure 5), comprised of the organization's physical assets. This layer is vital when performing MIA in cyber–physical systems (computer-controlled systems monitoring and controlling physical processes) [25] considering their architecture is often composed of two primary layers, a cyber layer, consisting of a corporate network, a control network and a demilitarized zone (DMZ), and a physical layer, which consists of sensors, actuators and physical devices. This leads to the two-layer structure: cyber asset layer and a physical asset layer, however mission and security concepts are evidently lacking and should be incorporated for a complete MIA.

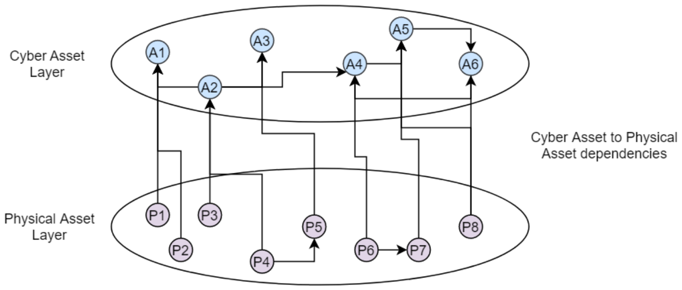


Figure 5 – MIA model for ICS: cyber assets (blue) and physical assets (purple) nodes.

2.1.2. MIA Sources

From the previously reviewed models, one can state that the organization's *cyber infrastructure*, composed by *services*, *assets* and *security data*, combined with the organization's *mission*, forms a suitable four-layered representation of the organization's environment to perform MIA and will be followed by BIA.

Nevertheless, those models only represent a design abstraction of the domain of the mission performers considered by each work. To perform MIA, the organization's environment model requires information about all the performers, how they interact with each other and how they condition each other. In this section it will be further explored how each layer may be populated by different sources, to identify each layer's components, their *horizontal dependencies*, between same-layer components, and their *vertical dependencies*, between different-layer components (both concepts adopted from the reviewed CT by Jakobson [22] (Figure 3)).

2.1.2.1. Mission Layer

Since the early 1990's, as information systems engineers and managers shift their attention from data and objects to the processes that the information system – and the organizational environment in which it operates – is intended to support, enable or enact [43], business-process models (BPMs) have become highly adopted by the organizations as a guiding principle, not only in the design and analysis of information systems, but also as a management discipline in its own right [44].

BPMs can be used to represent business-processes as threads of activities to be taken to complete a business-objective, which, all combined, portrays the organization's mission workflow and, subsequently, the horizontal dependencies at the mission layer. In addition, if there is a software tool that supports the models created by using modelling languages, then the BPM has the property of being executable and interpretable [45], and, as an executable model, they can then be used to predict various mission specific metrics and measure different performance characteristics. Progress has been made in executable standard modelling languages, such as the Business Process Model and Notation (BPMN) [46] and the Business Process Execution Language (BPEL) [47].

Initially and by design, BPMN and BPEL were often used in conjunction [48], seeing that BPMN started as a purely graphical business-process notation, used for the business user-centred perspective, and BPEL is an executable specification language (represents not only a conceptual, but also a behavioural model of the software system to be implemented), used for the technical specification. However, with version 2.0 launched in 2011, BPMN added their own executable environment, making it a standalone product that addresses both business and IT needs.

In a MIA context, the BPMN is one of the most widely used BPM at mission layer [13], with the laudable advantage of offering a graphical and intuitive language, simple to understand, allowing non-technical users to model and simulate process behaviour. This graphical nature also allows the comprehension and confirmation of the model by the organization's managers, perfecting the information flow between the systems admins and the organization's leadership. Several works ([6], [9], [11]–[14], [21], [22]) have taken advantage of BPMN for capturing time-ordered, operational event description that captures discrete, definable interactions among mission activities, while others as CMIA [27] developed a functional subset of BPMN, called Cyber Mission Impact Business Process Modelling tool to obtain a BPM oriented to cyber-processes, cyber-resources and cyber-effects. A notable example of how incorporating the mission layer creates models that are more adapted to the target environments is BP-IDS [49]. BP-IDS is an IDS that also leverages BPMN to model the organization business-processes and monitors their executions to identify non-compliance based on their specification, successfully reducing the false positive and false negative alerts rate.

Recently, researchers have proposed automated discovery of process specification, such as process mining, using event logs ([50], [51], [52], [53], [54], [55]), and hybrid approaches that use clustering techniques to cluster event logs into subsets, and then mining them to construct a simpler and specific model for every cluster [56]. These methods can then be used to find automatically the horizontal dependencies at the mission layer. Nevertheless, when it comes to gather information about the vertical dependencies between the mission and service layer, to map one service, or a combination of multiple service, to the mission's activities, the process is heavily dependent on manual input. BP-IDS attempts to facilitate this process through a graphical view that allows the user to create a BPM and map, manually, each business-process activity to the organization's services (and these to the organization's assets).

2.1.2.2. Service Layer

The service layer consists of all the services running on the organization's assets, and depicts how they depend or enable others (horizontal dependencies) and how they are linked to the assets they run on (vertical dependencies with the asset layer), considering vertical dependencies with the mission layer were already covered in the previous section.

Some tools exist to discover services and their vertical dependencies with the asset layer, by recognizing applications, firmware, or operating system (OS) running on network connected devices (hosts). OS detection's traditional approach is to use fields of the transmission control protocol (TCP) packet headers and compare them to a database of known OSes, a process known as *OS fingerprinting,* for which techniques that are based on other protocols have also been recently developed [57], such as ICMP (Internet Control Message Protocol), UDP (User Datagram Protocol) and DHCP (Dynamic Host Configuration Protocol), among others. For application detection, some techniques have been proposed to identify services and applications used by hosts on a network. For instance, observing that port 22 is accepting connections on a host indicates that an SSH (Secure Shell) service may be running. This is achieved by network scanner tools, such as *NMAP*[1], that can also capture the versions associated to that service [15]. Additionally, statistical analysis of IP packet size, timings and order can be used to classify application-specific traffic [58], as well as analysing application protocol messages syntax can be used to create signatures for the firmware version running on devices.

However, the complexity of populating this layer quickly adds up as different applications and services must interact with each other in order to function properly – the horizontal dependencies. The

[1] https://nmap.org/

two most common horizontal dependencies that can exist between two services include: enabling of one service by another, and the containment of one service within a package of multiple services [29]. For instance, typical applications, such as web, email, instant messaging, file sharing, and audio/video conferencing can rely on many supporting services, such as Active Directory (AD), Domain Name System (DNS), Kerberos, and Windows Internet Name Service (WINS) [59]. Problems at any of these services may lead to failure at a business-level, therefore, discovering the services running, and how they depend on each, becomes an important feature in MIA.

Although several MIA works do consider the service layer ([12], [18], [22], [23], [29], [31], [60]) they rely on manual methods or do not detail the extraction techniques used, supporting the rationale given by Bahşi *et al.* [13] that a prevalent model for the representation of services and a service-layer does not exist. Recently there have been a few attempts to automate dependency discovery. Automatic methods have been applied to network traffic or host-based data for the discovery of dependencies among network services ([59], [61]).

2.1.2.3. Asset Layer

Understanding the network and physical connectivity between assets in a complex system, and identifying the vulnerable ones, is crucial, as asset interconnections (the horizontal dependencies) and their associated vulnerabilities (vertical dependencies with the vulnerability layer) are the main enablers for malware propagation and network-based attacks [6].

To obtain information about assets, there is a plethora of research in MIA ([15], [16], [20], [26]–[29]) that leans on network inventory products, that organizations usually have, to keep up with an inventory of the organization's hardware, that describe how these components interconnect. However, as highlighted earlier, modern networks are continually changing with the popularity of wireless environments and BYOD infrastructures becoming more dominant, prompting new challenges for network admins to keep track of their entire network, making the task of manual book-keeping extremely resource intensive and error prone [20].

To facilitate the situational awareness of the asset layer, network discovery software is used by several MIA studies ([8], [21], [27], [28], [30], [31], [62]) to enable an automatic network topology discovery, and can be divided in two major categories:

- *Network Traffic Analysis* tools that intercept network packets to analyse the network's traffic. Some MIA research ([28] and [21]) specifically uses *Wireshark*[2] to capture data packets to identify traffic patterns that help depict the connectivity among assets.
- *Network Mapping* tools to sweep the network to identify and characterise machines. *NMAP* and *Nessus*[3] are instances of tools used, in addition to service discovery, to create and maintain the network's topology ([8], [27], [30], [31], [40], [62]), by identifying assets in the network, and mapping IP addresses to hostnames.

[2] https://www.wireshark.org/

[3] https:// tenable.com/products/nessus

To analyse the horizontal dependencies between assets, the physical topology of the network is not entirely sufficient: routers and switches allow communications between assets despite them not being physically connected. The communications that take place between assets are mainly subject to the firewalls' configuration of the hosts themselves, as well as the configuration of the routers and switches between the hosts. To further distinguish connectivity, firewall configuration is used with this goal by numerous research in MIA ([8], [14]–[16], [18], [28], [31], [40], [63]). These works do not specify how they integrate firewall's configuration into the MIA modelling, however research on the subject ([64], [65], [66]) suggests there are a few key points to consider:

(1) Firewall filtering rules have to be carefully written and organized in order to correctly implement the security policy [64].

(2) Manual definition of rules often results in a set that contains conflicting, redundant or overshadowed rules, resulting in anomalies in the policy [65].

(3) In distributed firewall environments, firewalls might also have conflicting policy rules when individual firewalls in the same path perform different filtering actions on the same type of traffic [66].

A significant amount of work has been reported in the area of firewall and policy-based security management ([64], [65], [66], [67], [68]) to address key points (1) and (2), while more recent works have been focusing on the analysis and detection of anomalies in firewall policy in distributed firewall environments ([69], [70]) to address point (3). While these works aim to detect and fix anomalies, they can be leveraged to understand the current state of the organization's firewall policy (misconfigurations and all).

Finally, it is important to identify the vulnerable assets that contributes for the spread of cyber-attacks. This can be addressed by vulnerability scanners tools to detect exposures arising from misconfigurations or flawed programming within a network-based asset. Several MIA studies ([8], [15], [27], [40], [62]) incorporate the *Nessus* tool to search services running on assets and examine those services, and their versions, for susceptibilities to known vulnerabilities, that allows to populate the vertical dependencies between the asset and vulnerability layers, where a vulnerability can be linked to only one asset, or shared between multiple assets, and an asset can have one or more vulnerabilities associated.

2.1.2.4. Security Layer

As observed in Section 2.1.1, most of the MIA models reviewed presented some type of the system security information: VTAC used IDS alerts related to the services running on the organization's hosts, and Jakobson and VASM used the organization's asset vulnerabilities, which leads to the conclusion that considering security related information appears to be a key point of MIA's modelling. While there are works that assume a more reactive methodology by consuming live *incident* related alerts from IDS ([15], [19], [34], [71]) or SIEM (Security information and Event Management) [38] systems, there are research that adopts a more active procedure and considers *threat* or *risk* related information ([16], [20], [27], [28], [30], [31]), although it often lacks information about sources and correlations with other layers. Nonetheless, the large body of research on MIA leverages *vulnerability* related information ([3], [17]–[19], [21]–[23], [28]–[31], [35], [36], [38], [40], [71]).

To keep up with information about vulnerabilities, system administrators resort to vulnerabilities databases to search for vulnerabilities that can be linked to current asset's configurations. For instance, if an organization acquired a Cisco router, or a *Linksys* wireless router, or *Solaris* version 9 running *Netscape Enterprise*, or anything that plugs into the organization's network, the brand or service name, or even the type of the device can be searched on a vulnerability database to discover how many, and which vulnerabilities, the network asset can be associated with. This information is then used to populate and update the vulnerability layer, which information is used by MIA to evaluate critical vulnerabilities for due fixing.

The Common Vulnerabilities and Exposure (CVE) dictionary [72] is currently paramount to identify vulnerabilities by assigning a unique identifier for publicly known information-security vulnerabilities in publicly released software packages, which can be found and reported by anyone: a vendor, a researcher or just a user can discover a flaw and bring it to a CNA (CVE Numbering Authority) which will review it and publish an identifier in the CVE database – a CVE-id – with a brief description. CVE-ids, accompanied by a brief and unstructured (although moderated) description are very limited in sophistication and expressivity, which standardized structures try to overcome in establishing consensus on what is being shared, such as CVRF [73], STIX [74] and MAEC (old CME) [75].

Moreover, there are other vulnerability databases, such as the National Vulnerability Database (NVD) [76] and CERT/CC Vulnerability Notes Database [77], and various vulnerabilities mailing lists, such as the Bugtraq [78] and *Symantec* DeepSight [79] databases, maintained by governments or commercial companies, that further develop on CVE's intelligence and provide risk scores, impact ratings, and mitigations strategies. One of the most sought database, integrated in several existing MIA works ([3], [8], [17], [18], [28], [30], [35], [36], [40]) is the NVD, that presents the list of public discovered vulnerabilities from the CVE dictionary, and uses the widely accepted industry scoring standard [36], Common Vulnerability Scoring System (CVSS) [80] to score each vulnerability according to its risk ("Low", "Medium" and "High").

Although there are numerous databases to populate and maintain this layer from, discovering their horizontal dependencies is a much harder procedure, since multiple combinations of vulnerability can be used by an attacker to reach different goals. The sequential order of exploited vulnerabilities that an adversary may follow to reach his/her goal represent a possible horizontal dependency among those vulnerabilities. The modelling and specification of attack scenarios (signatures) implemented by *attack languages*, often used in IDSs, can be used to produce horizontal dependencies among vulnerabilities, however, describing and modelling attack signatures can be complicated and susceptible to errors [81]. Attack signatures databases, such as BIG-IP ASM [82], can be searched by CVEs and used to infer horizontal dependencies among existing vulnerabilities.

2.2. Impact Propagation

As a single act, a cyber-attack is often not sufficient for an attacker to reach his/her ultimate goal, whereas multiple stages from attack preparation and network penetration to the final attacks often occur. In a typical cyberattack scenario, the attacker infiltrates the network from an entry-point and moves within the network until finding his/her way to the ultimate goal, whose damage can lead to a cascading effect that will disturb the correct behaviour at the higher mission level. This effect is called impact propagation.

Considering a multi-layered MIA model, impact propagation can be divided and identified as (1) impact *vertical propagation*, from a lower layer to a higher layer exploring vertical dependencies, and as (2) impact *horizontal propagation*, among elements from the same layer, respectively exploring horizontal dependencies between mission performers.

Generally, to explore vertical and horizontal dependencies among layers, a model-based analysis is used and can be categorized into *logic-based models* ([3], [15]–[19], [28]–[30]), *probabilistic-based models* ([18], [20], [31], [32]) and *sensitivity-based models* ([24]–[27], [33], [34]). Regardless of the modelling formalism, such models share a common objective: to propagate an initial condition throughout the system under evaluation; and a common feature: by themselves they do not provide a quantitative evaluation of the impact of cyber-attacks but can be combined with various metrics that do so, reviewed in the next section (Section 2.3).

2.2.1. Logic-based Propagation

Logic-based analysis approaches resort to an attack graph model ([3], [15]–[19], [28]–[30]), that uses a sequential and explorative process to gradually identify and assess security-related conditions of the system under evaluation, by representing the way vulnerabilities can be combined and exploited in a network to compromise the network's security policy [83].

There are essentially two types of attack graphs [35]. The *state enumeration attack graph*, used in early formulations [84], represents transitions of a state machine, where each node represents the entire network state and the edges represent state transitions caused by the adversary's actions, resulting in graphs that enumerate transition paths through state space. Currently it has been recognized it is not necessary to explicitly enumerate attack states, which have serious scalability problems [85], but rather it is sufficient to form a graph of dependencies among exploits and security conditions, which leads to the second type of attack graphs – *dependency attack graph* [35]. A dependency attack graph, commonly abbreviated to *attack graph*, represents the overall state of the system, where a node in the graph depicts a system's condition and edges between nodes represent the causality between conditions (Figure 6).

Although some MIA studies focus on generating their own models of attack graphs ([1], [14], [19], [32], [37]), there are several that use existing attack graph tools like Cauldron [8] ([10], [11]), TVA [62] ([16]) and MulVAL [86] ([3], [17], [35]). All of these attack graph generation tools essentially provide a snapshot in time analysis. Being a commercial (and paid) tool, Cauldron focus on providing better visualization and user experience, however, academic projects such as TVA and MulVAL can produce clear and relevant results. Furthermore, MulVAL being open-source, contrary to TVA, and having algorithm complexity close to that of the commercial tool Cauldron [87], makes it a viable and helpful tool among attack graph techniques.

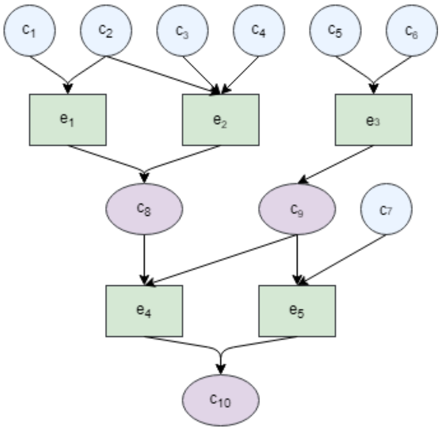


Figure 6 - Example of a dependency attack graph, where e_i represents exploits and c_i represent preconditions and consequences of the exploits.

MulVAL (Multihost, Multistage Vulnerability Analysis) generates an attack graph based on (1) data log, that combines a variety of information on vulnerability of servers and clients, (2) information on machine and network configuration, (3) rules specifying the interactions by different parts in the network; and (4) security policy on legal data access by system users, all described by logic programming language Datalog. The use of Datalog to define rules and network's configuration provides a great deal of flexibility for the creation of new assertions to capture the network and desired propagation configuration, resulting in extensive research in extending MulVAL ([3], [17], [88]–[92]), including the MIA approach proposed in this thesis. MulVAL also incorporates vulnerability configuration (default NVD), OVAL (Open Vulnerability and Assessment Language) and Nessus vulnerability reports file. It uses this data log to produce an attack graph, with worst-case complexity of $O(n^2)\sim O(n^3)$ [89], in textual and graphical format, which can then be used to analyse how an attack occurs in a network.

2.2.2. Probabilistic-based Propagation

There is research ([18], [20], [31], [32]) that opt for a probabilistic-based approach to impact propagation, mostly based on Bayesian Networks (BN), to represent cause-and-effect relationships based on the assumption that all data (domain knowledge and accumulated evidence) can be conveniently represented by probability functions.

Formally, a BN is a probabilistic graphical model characterised by a directed acyclic graph (a type of graph that have directed edges between nodes and is without cycles connecting the nodes). Nodes represent the variables of interest, directed edges between pairs of nodes represent the causal relationship among the nodes, and such causality relationship is specified with conditional probability distribution functions [93]. The model considers these conditional probability distribution functions to propagate the belief in a hypothesis at the root node to related nodes and provides the most probable path to reach a desired node and directly observes evidence at that node.

Some MIA research ([32]) resorts to construct a BN by leveraging the collected intrusion evidence from various security sensors, and infer the probabilities of interested security events, such as a system object or a mission's activity being tainted. ARGUS [18] uses BNs to propagate the impact factor (reviewed next in Section 2.3) through the mission performers, to measure the likelihood of achieving the mission goals. M-Correlator [31] combines the relevance factor of alerts (reviewed in Section 2.3), the vulnerabilities classification given by the user and the probability of the intrusion to success, to create a BN to classify the impact of an incident, although it does not propagate that impact to the mission layer.

While BNs allow to propagate the impact throughout the organization's infrastructure and reach the mission layer, most often, the problem of determining the impact propagation using a BN approach is the lack of information about the domain required to fully specify the conditional dependencies between random variables. If available, calculating the full conditional probability for an event can introduce a high modelling overhead. Also, a network graph cannot be assumed to have an acyclicity constraint, and a joint probability distribution is not defined for cyclic graphs [20], which means the network's dependencies cannot be fully considered when propagating the impact throughout the network's infrastructure using a BN.

2.2.3. Sensitivity-based Propagation

Other MIA studies ([24]–[27], [33], [34]) implement a sensitivity-based approach to impact propagation, where they use active perturbation to measure how sensitive a model is to changes in its control parameter values and infer consequences in the system [24]. Studying the system's state before and after the perturbation allows the quantification and classification that the perturbation's impact (from a cyber-attack, a malfunction or intended interference) had on the system.

A sensitivity analysis usually begins by modelling the system under evaluation to calculate measures of effectiveness (MoEs) to be used as a reference to reflect the normal behaviour of the system. Then, the simulation is rerun under different initial conditions, often by manual intervention, to reflect changes in system capabilities caused by an attack to control parameters. New MoEs are calculated and the impact of the attack can then be determined by comparing the two MoE values (before and as a result of the incident).

Since there are no logical or probabilistic models to follow, this methodology can be easily understood and implemented, however, its implementation requires a great level of *a priori* information to either actively perturb all mission performers to understand the cascading effects throughout the infrastructure, which results are inherently difficult to obtain in large enterprise networks, or to learn a list of candidate control parameters to perturb, which results in an incomplete understanding of how the impact can propagate in the entire infrastructure.

2.3. Impact Measurement

As reviewed in the previous section, impact propagation methods are able to provide an overview of potential attacks and their evolution, and, combined with a MIA model in place, are capable of giving information about which mission performer may be impacted and its role at mission-level, which supports the decision-making process when addressing a cyber incident. To further evaluate the consequences of an attack's impact, impact metrics should be integrated within the MIA model [30] and propagated throughout the model to evaluate how the impact reaches the organization's mission.

The prime focus of the present work is on the impact modelling and propagation aspects of MIA, nevertheless the importance of the impact quantification aspect is recognized, and, to that effect, the metrics used by the reviewed MIA research are briefly analysed in this section. Kotenko *et al.* [38] provides a comprehensive literature review on current security metrics taxonomies and carries out a technique to define and calculate numerous metrics for security assessment. Nonetheless, from a MIA standpoint, their proposed classification of security metrics is not inclusive considering that metrics at a mission-level were not contemplated.

This section outlines the reviewed impact metrics following the MIA layered model – *mission*, *service*, *asset* and *security* level metrics – and presents them in Table 1. It further distinguishes them by two categories: *qualitative metrics*, that focus on the weight of the impact according to the impacted performer's characteristics, which remain constant independently of a cybersecurity event. These metrics help system defenders to prioritise the impact; and *quantitative* metrics, to bring together metrics to assess the impact of a security incident and the performance capabilities of a performer, which loss will directly reflect restrictions imposed by the impact. Additionally, it is to be noted that some reviewed metrics' original designations were slightly adapted, to distinguish and contextualize their purpose among all of the presented metrics in this section.

Mission-level qualitative metrics define the value an activity has in respect to the mission they are a part of – for instance, an *Activity Value* factor ([17], [31]) may be used to measure the user's preferred activities, or thread of activities, when redundancy exist in accomplishing a business-process. On the other hand, the quantitative impact measurement at this level can be seen as the mission capability, by the *Mission Operationality* ([22], [29], [23]), combining an *Activity Efficiency* factor [19] for each activity pertaining its efficiency on performing, which will decrease by the impact. The confidentiality, integrity and availability (CIA) security classification triad is also suggested [40] to qualify, and quantify, the impact on mission-layer, by representing the security requirements a business-objective requires to perform its purpose.

Qualitative metrics on a ***service-level*** can be defined by application characteristics and features of services dependencies to measure the relevance of the service to the overall mission. However, most often, service-level metrics are not being used to stand on their own, but to qualify the importance a service contributes for the operationality of the assets it runs on. Two instances of this is the *Service Criticality* [40] and the *Business Value* [38] metrics. To quantify the impact on a service, some research proposes to use a *Service Operationality* ([22], [29], [23]) metric to numerically characterize the loss of operational capability the service has to perform its intent.

Asset-level metrics can be defined from the organization's network topology and asset configurations and represent the majority of impact metrics in MIA research. Qualitative metrics involve attributes to take into account how hard is to access the asset (*Access Complexity* [35]) and therefore how hard is to compromised it; the frequency the asset communicates with other assets (*Access Frequency* [36]) that subsequently facilitates the propagation of the impact; the *Asset Criticality* ([15], [40], [37]) to value how crucial the asset is to the success of the mission; the *Asset Susceptibility* [40] to distinguish the priority an asset on a confidential network has from one in a public network; the *Asset Sensibility* [40] to raise the degree of the asset's importance according to the sensibility of the data stored in it, for instance personally identifiable information (PII); a *Hot Asset* [37] attribute to indicate the assets that are effectively being used, in real-time, by the missions; an *Asset Placement* [40] value to further evaluate its importance according to the organizational department it belongs to. From a quantitative viewpoint, numerical values are assign to assess the asset's normal requirements, and changed according to the impact and characteristics of the cyber event: an *Asset Operationality* ([22], [23], [29]) score is devalued to represent the effect of an attack on the operationally state of the asset, with a range of scores [0-1] where 0 is assign when the asset is completely destroyed; the *Asset Exposure* [40] aims to value the likelihood of a host being the target according to the asset's and attack's characteristics; the CIA triad is also used ([38] [40] [26] [14] [30] [9] [37] [94]) to represent the violation on each one of those requirements as the impact; finally, an *Asset Efficiency* [19] score is assigned to assess how efficient an asset operates under particular conditions, and the loss of efficiency reflects the impact on the asset. Differently from the previous *Asset Operationality*, the *Asset Efficiency* score helps distinguish the most efficient assets in supporting the business-processes when redundancy exists.

At ***security-level***, metrics pertain different types of security data, such as alerts, vulnerabilities, attacks and the skills of the attacker. Qualitative metrics related to security alerts raised by network monitoring sensors include an *Alert Priority* [31] calculation to indicate the degree to which an alert is targeting a critical asset and the amount of interest the user has registered for this type of security alert (given by an *Alert Interest* [31] score); an *Alert Relevance* [31] calculated through a comparison of the alert target's known characteristics against the known vulnerability requirements of the incident to determine the likelihood of exploitation; from a vulnerability viewpoint, the impact is often qualified by *CVSS* scores ([3], [17], [23], [29], [30], [35], [36], [38], [40]) (reviewed in Section 2.1.2.4) and quantified in terms of CIA [40]; regarding an attack, the *Likelihood* ([35], [36]) score evaluates if the attack path can lead to a successful exploit. Furthermore, an attack can be qualified by the attacker perspective by an *Attacker Skill* score ([38], [14]) to determine its expertise and resources in performing the attack. The CIA triad can also be used to represent the attack's goals of impact. To quantify the attack impact, several studies employ the *Attack Impact Factor* ([15], [23], [29], [22], [36], [17], [3]) to indicate to what degree the attack is capable to compromise the attacked performer.

Observing the grouped metrics presented in Table 1 some conclusions can be taken:

- Some metrics are related to multiple layers. This is the case for performer's *CIA requirements*, *operationality capability* and *criticality*. This cross-layer applicability offers an advantage when integrating metrics in current evaluation models that consider multiple layers.
- CIA security triad metric can be used to both qualify and quantify the impact.
- Also, any of these metrics can be used to determine the impact of a cyber-incident, but it is important to emphasize that, by itself, a unique metric may not be sufficient to qualify and quantify the impact. However, taken together, the resulting value may give a good representation of the impact a cyber-incident may have.

Table 1 - Classification of security metrics

Classification Level	Qualitative Metrics	Quantitative Metrics
Mission	- *Activity value* - *CIA Mission Requirement*	- *Activity efficiency* - *Mission Operationality* - *CIA Mission Requirement*
Service	- *Service Criticality* - *Business Value*	- *Service Operationality*
Asset	- *Access Complexity* - *Access Frequency* - *Asset Criticality* - *Asset Susceptibility* - *Asset Sensibility* - *Hot Asset* - *Asset Placement* - *CIA Asset Requirement*	- *Asset Operationality* - *Asset Exposure* - *Asset CIA* - *Asset Efficiency* - *CIA Asset Requirement*
Security	- *Alert Priority* - *Alert Interest* - *Alert Relevance* - *CVSS Vulnerability Exploitability* - *Likelihood* - *Attacker skill* - *CIA Attack Impact Goals*	- *Attack Impact Factor*

2.4. Summary and Discussion

This chapter has presented relevant studies in approaching MIA. The summarized literature review is depicted in Table 2 and was narrowed down to include the approaches that address the two topics most pertinent to the context of this dissertation – the modelling and propagation aspects of MIA. The studies are distinguished by the domain of study they were conducted in (military, business or essential services provider infrastructures), the assessment layers they address (mission, service, assets and security), the propagation methodology employed (logic, probabilistic or sensitivity) and the impact metrics used according to the abstraction layers they are related to. As noticed during this chapter, not all presented works provide all the sources required to populate the proposed conceptual models, or the calculation techniques for the contemplated metrics, but they do consider them at different levels of abstraction, therefore are included here.

Table 2 - Summary of the literature review.

Paper [Reference]		[18]	[28]	[15]	[16]	[17]	[22]/[29]	[23]	[24]	[25]	[26]/[27]	[30]	[20]	[3]
Domain	Military	✓			✓						✓			
	Business		✓	✓		✓	✓	✓				✓	✓	✓
	Infrastruct.								✓	✓				
Assessment Layers	Mission	✓	✓		✓	✓	✓	✓			✓		✓	✓
	Service	✓		✓		✓	✓	✓				✓		✓
	Asset	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
	Physical Asset								✓	✓				
	Users			✓										
	Security	✓	✓	✓	✓	✓	✓	✓			✓	✓		✓
Propagation	Logic	✓	✓	✓	✓	✓	✓	✓				✓		✓
	Probabilistic	✓			✓								✓	
	Sensitivity								✓	✓	✓			
Metrics	Mission		X		✓	✓	✓	✓			✓		X	
	Service						✓	✓						
	Asset	✓		✓		✓	✓	✓	✓	✓	✓	✓		✓
	Security			✓			✓	✓						✓

After analysing the current methods and examined their applicability, some conclusions can be taken on the most used approaches:

- Most of the studies are under the business domain, which supports the motivation behind this work, based on the increasing interest by organizations to protect their businesses by taking advantage of a MIA solution. To prove its cross-applicability, this work is envisioned in a business domain but demonstrated under an infrastructure domain.
- The assessment layer most used is the asset layer. This may be explained due to the plethora of research, approaches and tools that can be leveraged to concretize this layer. Followed is the security layer, which may be similarly reasoned due to the large number of community-contributing databases with security-related information. This security-related information includes sources such as IDS and SIEM's alerts, vulnerabilities, attack references and risk/threat information, however, a description of how the latter is implemented could not be located. On that account, this work includes a threat layer in its assessment model and demonstrates how cyber-threats are considered and how they affect mission impact.
- The logic-based propagation appears to be the most used, however, existing propagation tools require a great level of operating knowledge. This works allows the user to bypass the direct

operation with MulVAL [86] propagation tool by automatically preparing the tool required input model from the knowledge database, and automatically interpreting the tool output to a well-known format. Additionally, from the literature review, it is possible to conclude that using hybrid methods by combining logic-based and probabilistic-based appears to be getting attention as a worthwhile approach.

- Lastly, impact measurement is mostly done on an asset-layer but, as an increasing subject of study, metrics that can be implemented in different assessment layers are starting to draw interest.

Finally, the approach most related to the MIA solution proposed in this dissertation is the solution put forward by C.Cao *et al.* [3] that, likewise, adopts a logic-based propagation by altering MulVAL [86] attack graph tool basic configuration to address the security, asset, service and mission layer. However, some key features distinguish the scope of this work:

- The assessment of the security layer of the present work is done at a cyber-threat level, instead at the vulnerability layer. This allows the user to perform MIA of any desired threat landscapes, instead of considering only the scanned vulnerabilities.
- The proposed assessment model integrates with multiple data sources (packet captures, firewall configurations, IDSs), existing tools (network analysers and IDSs), and previous studies (firewall's anomalies detection algorithms and threat's classification standards), as opposed to only the vulnerability-related data sources proposed by MulVAL.
- Furthermore, contrarily to the work proposed by C.Cao *et al.*, it is not required that the user have knowledge about the intricacies of operating MulVAL, since, as stated before, the work of populating the evaluation model and creating the required input to MulVAL is automatically done behind scenes, allowing the user to use the tool at a much higher-level.

3. Proposed Approach

The main goal of this dissertation is to provide a MIA solution to better understand how security threats can be leveraged to impact the organization's business. This chapter describes BIA[4] (Business Impact Assessment), an integrated approach for understanding mission impact of cyber threats. The proposed approach for the design of BIA is two-fold: (1) to create a multi-layered evaluation model for MIA that can be easily integrated with current information sources and (2) to put forward a simulation platform that allows to reproduce how the impact of exploited cyber-threats propagate throughout the organization's infrastructure and to assess the impact on the organization's mission.

The principles of BIA are described in the following sections.

3.1. System Architecture

This work proposes a two-stage approach for MIA and is architected as illustrated in Figure 7. BIA's general idea is to first create a knowledge database with the organization's cyber infrastructure and business profile – the Setup stage – to then be used to simulate the impact of a user-chosen compromised entry-point on the organization's mission – the Simulation stage.

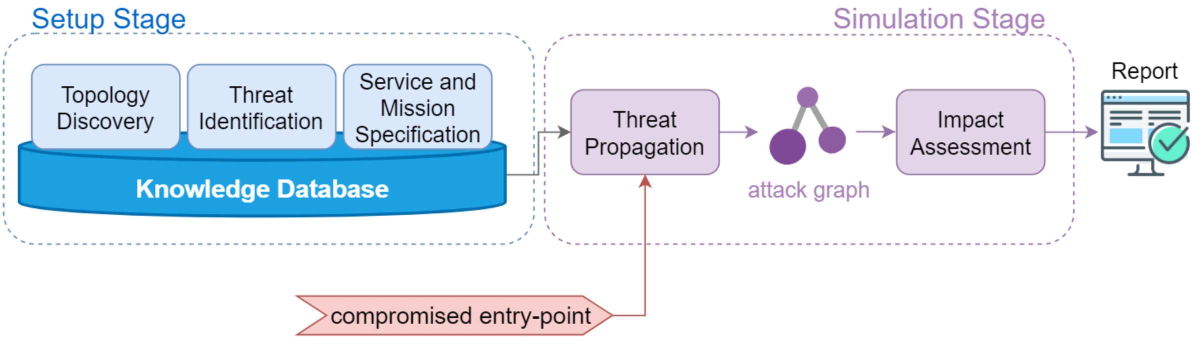


Figure 7 – BIA's architecture.

The approach takes a set of three knowledge units as input during the Setup stage and a compromised entry-point during the Simulation stage to generate a MIA report as the output.

Prior to know the extent to which the organization's mission under test can be impacted if a given threat is exploited, one needs to understand how the organization's cyber infrastructure and business is configured: what are its assets, what threats they possess, how are they interconnected and how are they related to the organization's mission. This is addressed during the Setup stage (Section 3.2) of the proposed approach to construct a knowledge database to build the evaluation model. The Simulation stage (Section 3.3) describes how the simulation platform was devised to propagate the impact of an exploited threat to the mission, using the evaluation model built before.

[4] BIA is the result of this thesis and its design was published in CPS4CIP 2020 conference (https://sites.google.com/fbk.eu/cps4cip20) which proceedings will be published in SPRINGER Lecture Notes in Computer Science.

3.2. Setup Stage

As outlined in Figure 7, the central idea of this stage is to capture the cyber infrastructure and business information, and consolidate it in an integrated data representation to be interpretable by the simulation. The data representation proposed to map the organization's cyber infrastructure onto the business-objectives is based on a four-layer evaluation model, as illustrated by Figure 8, to represent the entities considered for MIA and the relationships between them.

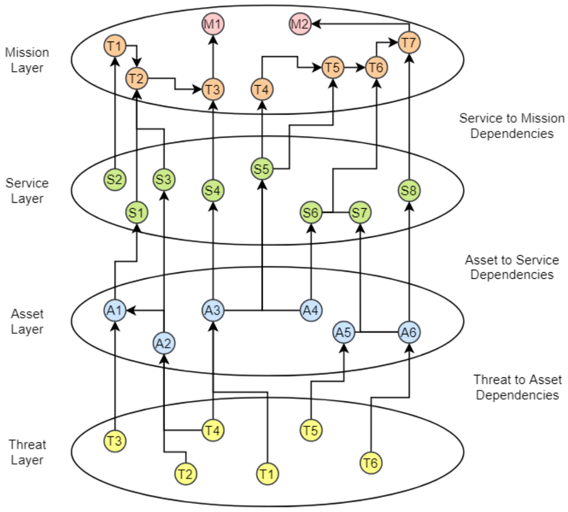


Figure 8 - Architecture of BIA's assessment model.

Figure 8 shows the abstraction layers to represent MIA related entities considered by this work: threat, asset, service and mission, which are bridged by four types of many-to-many relationships: an asset *has* (is exposed to) a threat and *runs* a service; the service *provides* an activity which in turn *supports* a business-objective.

To populate the evaluation model based, BIA's Setup stage comprises three knowledge units that mine different data sources to extract the required information, which conceptual design is further described in this section: a *Topology Discovery* unit (Section 3.2.1) to populate the asset layer, a *Threat Identification* unit (Section 3.2.2) to populate the threat layer and the *asset-has-threat* relationship, and finally, a *Service and Mission Specification* unit (Section 3.2.3) to populate the service and mission layers and the relationships between them and the asset layer.

3.2.1. Topology Discovery

The first step in populating BIA's multi-layered model to represent the organization's profile is performed by the Topology Discovery unit, that aims to gather information about the asset layer. To achieve this purpose, this unit receives two types of input, as illustrated in Figure 9: (1) network packet captures and (2) firewall configuration, which are handled using two different components: the *Network Discovery* (Section 3.2.1.1) and the *Connectivity Discovery* (Section 3.2.1.2), that address each data source, and upload the refined and formatted data to the knowledge database.

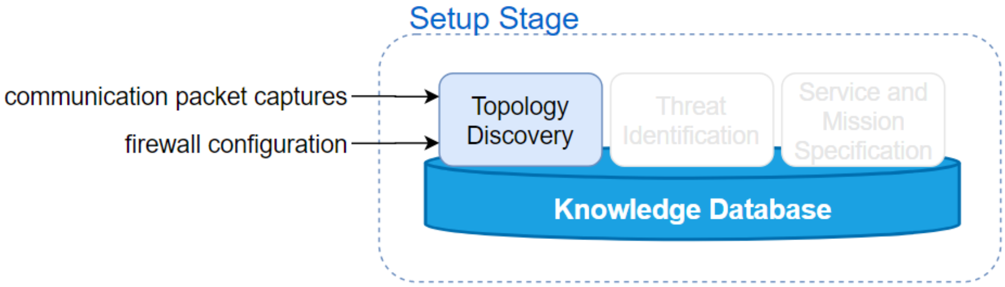


Figure 9 - The Topology Discovery unit. The first knowledge unit of the Setup stage

At the end, this knowledge unit stores its findings about discovered assets and reasoned connectivity between assets, in the knowledge database to be used by the next knowledge units.

3.2.1.1. Network Discovery

This component aims to gather information about the organization's assets. Information about assets may include their IP (Internet Protocol) addresses or MAC (media access control) addresses, while communications could be identified using network protocols that use network ports (for instance TCP and UDP protocols), or protocols that do not use network ports (e.g., ICMP protocol). To obtain this information, the component resorts to a network analyser tool that receives packet captures containing network communications exchanged between the IT components of the infrastructure under evaluation. Using basic dissection techniques[5], those packet captures are parsed to extract information about the infrastructure assets and their communications.

Additionally, the asset layer can also be fed with other sources of information, such as asset management systems that keep track of the equipment and inventory vital to the operation of the organization's business-processes, or process identifiers (PID) within a host to identify asset dependencies (for instance, virtual machines running on host machines).

The discovered assets and dependencies among them are then consolidated in the knowledge database as the ground to model the asset layer.

3.2.1.2. Connectivity Discovery

Even though network captures provide a wide perspective of the network topology, non-frequent communications may be missing from packet captures. To complement connectivity information previously gathered using packet captures, this component inspects firewall configuration to infer missing allowed communications. These firewalls can be either asset-based firewall software, or firewall functionality provided by hardware devices, such as routers or firewall appliances, and their place in the infrastructure is obtained from network infrastructure documentation, given as input, to generate firewall *domains* (the group of assets protected by each firewall). It is important to note that, even if a firewall allows a type of communication, it does not mean this communication is not filtered along the way to its destination, or even completely stopped, by other firewalls, as firewall hierarchies are often in place. Figure 10 - Example of firewall hierarchy filtering.Figure 10 illustrates an example of the filtering effect a hierarchical policy can have.

[5] https://www.wireshark.org/docs/wsdg_html_chunked/ChapterDissection.html

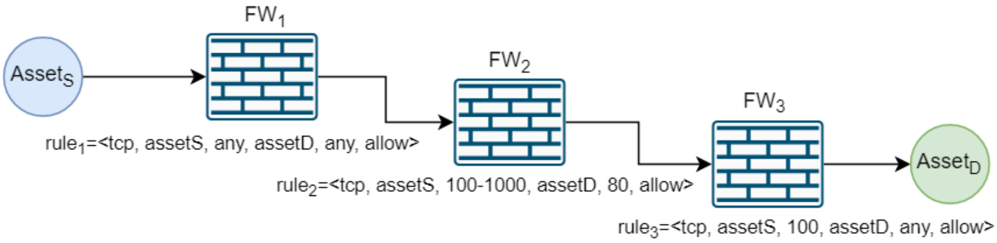

FW1
FW2
FW3
AssetS
AssetD
rule1=<tcp, assetS, any, assetD, any, allow>
rule2=<tcp, assetS, 100-1000, assetD, 80, allow>
rule3=<tcp, assetS, 100, assetD, any, allow>

Figure 10 - Example of firewall hierarchy filtering.

In Figure 10, firewall FW_1 has a rule, $rule_1$, that allows $Asset_S$ to communicate with $Asset_D$ by TCP protocol, from any port to any destination port. Along the way, FW_2 filters the source port of the communication to a range of source ports (from port 100 to port 1000) to the destination port 80 with $rule_2$, and finally, FW_3 further filters the communication allowed with $rule_3$ only consenting traffic to $Asset_D$ from $Asset_S$ if originated on port 100. The rule that reflects the connectivity that is effectively allowed by the hierarchical policy would result as:

$$rule_{allowed} = < tcp, asset_S, 100, asset_D, 80, allow >$$

To assess the communications that are effectively allowed by the firewall policy environment, this component comprises two algorithms, a ***Comparing Algorithm*** to first assess allowed communications by each individual firewall, and a ***Filtering Algorithm*** to address firewall hierarchy and assess which rules survive the filtering action.

3.2.1.2.1. Comparing Algorithm

The first step is to determine all the communications that are effectively allowed by each firewall policy (i.e. the rules that define what kind of traffic is allowed or denied). When a packet arrives at a firewall it is tested against each rule sequentially, meaning the firewall rules are order sensitive and the sequence of the firewall rule's list is to be taken into consideration when trying to understand which communication packets are effectively allowed. The proposed algorithm is designed to work as follows:

(1) Take each $deny$-rule and compare it to the $allow$-rules that come next. Considering d and a as a $deny$-rule and $allow$-rule, respectively, and f as the field to compare, it is possible to arrive to four possibilities, as suggested by previous work [65] and illustrated by Figure 11:

- The $allow$-rule is a *subset* of the $deny$-rule, or they are exactly equal, when all the packets matched by the $allow$-rule are completely matched by the $deny$-rule that appears first on the sequence. This is depicted by Figure 11-(a).
- The $allow$-rule is a *superset* of the $deny$-rule when a portion of the packets matched by the $allow$-rule are first matched by the $deny$-rule. This means there are two disjoint portions of the $allow$-rule that are effectively allowed, and one portion that will be matched by a $deny$-rule that comes before. See Figure 11-(b).
- The $allow$-rule is *correlated* with the $deny$-rule, when a portion of the packets are matched by the $deny$-rule but it does not constitute a subset or superset. This is illustrated by Figure 11-(c) and Figure 11-(d).

- Finally, the $allow$-rule and $deny$-rule are disjoint when they have a field for which they have completely disjoint values. This is depicted in Figure 11-(e).

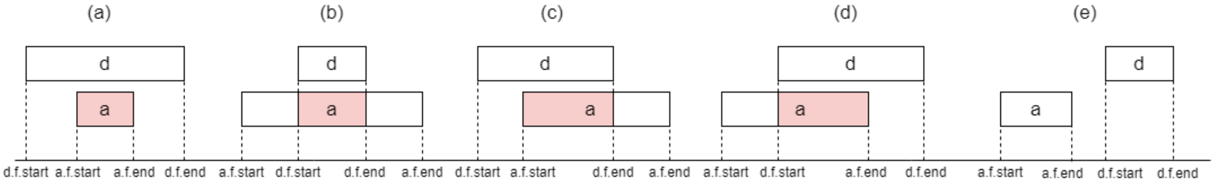

(a)
(b)
(c)
(d)
(e)
d
a
d
a
d
a
d
a
d
a
d.f.start a.f.start a.f.end d.f.end
a.f.start d.f.start d.f.end a.f.end
d.f.start a.f.start d.f.end a.f.end
a.f.start d.f.start a.f.end d.f.end
a.f.start a.f.end d.f.start d.f.end

Figure 11 - Comparison possibilities between one field of two rules.

(2) Remove from the $allow$-rules the parts in common with the $deny$-rule (red zones in Figure 11), creating new $allow$-rules to represent only what is effectively allowed by each $allow$-rule.

Applying this method to all $deny$-rules in a firewall's configuration results in a list with only $allow$-rules ($allow$-list) that represent all the possible communications that may pass through the firewall.

3.2.1.2.2. Filtering Algorithm

The second algorithm was designed to inspect each individual firewall $allow$-configuration (obtained by the previous algorithm) to assess what communications are effectively allowed between different firewall domains. To assess which rules survive the filtering hierarchy, and how, while traversing the network and the firewall infrastructure that constitute it, **three** distinct actions are proposed to be taken:

(1) First, classify each $allow$-rule from the firewall $allow$-list according to its source and destination to understand which rules should be submitted to the other firewall's policies. It is proposed four possible classifications:

- A rule is classified as **INTER** if both its source and destination are outside of the firewall's domain, meaning the firewall acts as an intermediator for communications related to this rule.
- On the other hand, a rule is classified as **INSIDE** if both its source and destination are inside the firewall's domain. This implies the firewall is the sole decider upon communications related to the rule.
- A rule is classified as **OUTBOUND** if its source belongs to the firewall's domain, but its destination is outside. The firewall will be the first firewall deciding the effect of the rule, but communications related to the rule will be further filtered by other firewalls in the way to their destination.
- Lastly, a rule is classified as **INBOUND**, if its destination is inside the firewall's domain but its source comes from outside the domain. Therefore, this firewall will be the last one deciding upon communications related to the rule.

(2) Next, **OUTBOUND**-rules are propagated to adjacent firewall's to be compared with their configurations and filtered accordingly by their **INTER**-rules and **INBOUND**-rules, following the same reasoning proposed by the previous algorithm. Since **INSIDE**-rules' source and destination are protected by the same firewall, the rule does not need to be further filtered and communications related to that rule are considered allowed.

(3) Repeating this process to every firewall's rule list, results in a list of all allowed communications in the infrastructure.

From the resulting list it is possible to infer connectivity that was maybe missing from the Network Discovery component when processing communication packet captures. The inclusion of the missing connectivity reflects better how the current firewall policy allows connectivity that may be leveraged by an attack to move within the network. This $allow$-list is translated to possible connectivity and can be used to populate horizontal dependencies at asset layer.

3.2.2. Threat Identification

The next knowledge unit, the Threat Identification, imports information about the asset layer (already stored in the knowledge database) and, from user input, creates a threat layer and maps it to assets.

One way to approach this would be mapping each asset to a set of threats the asset is exposed to. However, as threat landscape is in constant changing, this information would need to be constantly updated, which could become impractical. Therefore, this work proposes mapping threats to assets according to their type (chosen by the user).

To do this, the user's input must be threefold (Figure 12): (1) assign types to assets, (2) assign threats to types of assets, (3) use a classification system to group threats.

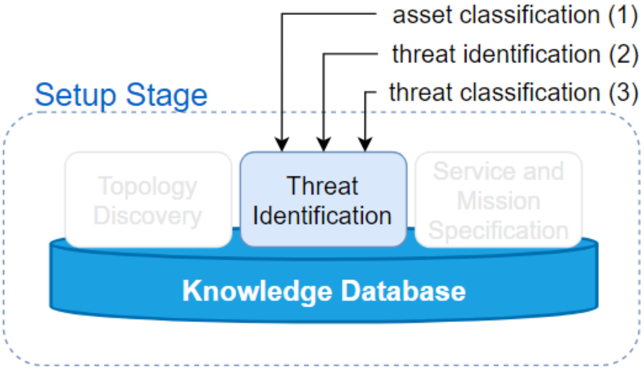


Figure 12 - The Threat Identification unit. The second knowledge unit of the Setup stage.

The input can be given with any desired semantical description to classify types of assets and threats, from which the Threat Identification unit then proceeds to map threats with the corresponding assets and stores that information in the knowledge database.

3.2.3. Service and Mission Specification

The third step of the Setup stage aims to bridge the assets found by the Network Discovery unit, to the organization's business goals, specifically the organization's business-processes. This is achieved by the Service and Mission Specification unit (Figure 13).

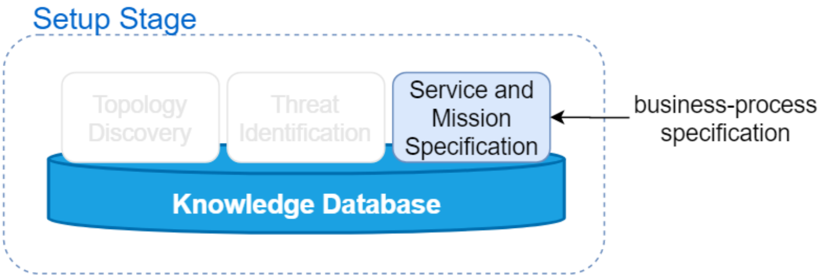


Figure 13 - The Service and Mission Specification unit. The third knowledge unit of the Setup stage.

As stated before, a business-process is represented by a collection of activities to be accomplished, which are provided by services running on assets. Figure 14 illustrates how a business-process, composed of three sequentially tasks, can be modelled.

This unit is envisioned to receive business-processes specification, consisting of the sequence of activities it takes to accomplish one process, the applicational services supporting those activities and the assets they are running on and maps this information to the assets already stored in the knowledge database.

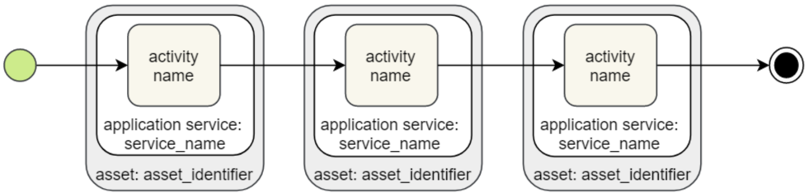


Figure 14 - Business-process general specification, composed of three activities.

3.3. Simulation Stage

Following the Setup stage, which results in a fully populated knowledge database based on the proposed layered model, the *Simulation* stage proceeds to simulate the impact of a user-chosen entry-point to the system and perform MIA. This is proposed to be achieved by two modules as outlined in Figure 15, a *Threat Propagation* module (Section 3.3.1) that aims to propagate the threat at the entry-point, throughout the organization's cyber infrastructure to reach the mission; and an *Impact Assessment* module (Section 3.3.2) to interpret the simulation's outcome and produce a report of MIA relevant information.

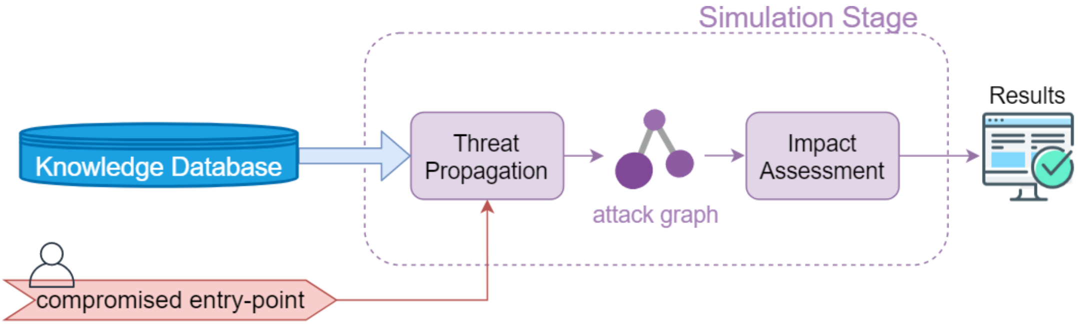


Figure 15 – BIA's Simulation stage.

3.3.1. Threat Propagation

The Threat Propagation module takes the main stage for the impact propagation simulation, based on an attack graph model, where the goal is to determine whether a compromised asset is likely to deleteriously affect any of the business-objectives of the organization. To this end, this module is designed as a simulation platform which is configured with the organization's infrastructure and mission identified and modelled by the Setup phase.

The simulation begins with a user-chosen entry-point (a specific asset and exploited threat) and ultimately tries to determine which organizational business-objectives would be affected if that asset became unreliable or unavailable. Starting from that entry-point, the simulation performs a bottom-up analysis, searching for attack paths by leveraging the organization’s model interdependencies (vertical and horizontal) to propagate the initial threat. If an asset is accessible and has a threat, then is exploitable and the simulation advances to that asset. Additionally, if an asset runs a service that has a role in the mission, then the threat’s impact is propagated towards the mission’s activity (or activities) the service supports, and, from there, to the business-process(es) that rely on those impacted activities.

This threat propagation is achieved by resorting to logic programming to express how the propagation advances with a set of series of Horn clauses, a logical formula that takes a particular rule-like form: $L_0 \leftarrow L_1, \ldots, L_n$, where $L_i \; \forall i \in N$ are literals, and if $L_1, \ldots, L_n$ are true then L_0 is also true. Semantically, this type of clauses represents the *preconditions* required to reach a goal (or *postcondition*). This feature gives it valuable properties to create a set of desired rules, and the preconditions that must be met, for the propagation simulation to advance.

Any verified rule by the current simulation’s environment configuration can then be represented by a node in an attack graph to depict its postcondition, while edges represent a new propagation step between two adjacent nodes. In this way, when the simulation finishes exploring all possible paths that reach the mission, an attack graph is created, depicting all the possible threat propagation paths found from the simulated entry-point to the organization’s business-processes.

In the design of this module, the threat propagation was defined using the following Horn clauses presented in Table 3.

Table 3 - Horn clauses used to define threat propagation.

Clause Number	Clause Description	Clause’s Postcondition (L_0)	Clause’s Preconditions ($L_1, \ldots, L_n$)
1	Entry-point compromised	$compromisedAsset(Asset)$	$attackerLocated(Asset)$, $threatExists(Asset, Threat)$
2	Attack propagated to another asset	$compromisedAsset(Asset2)$	$compromisedAsset(Asset1)$, $connectivity(Asset1, Asset2)$, $threatExists(Asset2, Threat)$
3	Attack propagated to the applicational service	$compromisedService(Service)$	$compromisedAsset(Asset)$, $runsService(Asset, Service)$

4	Attack propagated to the business-process	$compromisedProcess(Process)$	$compromisedService(Service),$ $runsActivity(Service, Activity),$ $runsProcess(Activity, Process)$

Observing *Clause 1*, it is possible to recognize how it can be used to define how an asset becomes the system's entry-point: if the attacker has control of $Asset$, represented by the literal $attackerLocated(Asset)$, and if $Asset$ has a threat, $threatExists(Asset, Threat)$, then the attacker can exploit it and compromise the asset – given by $compromisedAsset(Asset)$, since all preconditions were met. Following the same reasoning, the next clauses define how an attack propagates to another (*Clause 2*) by leveraging connectivity between assets; how a service may be compromised (*Clause 3*) and how a business-process may be compromised (*Clause 4*).

To verify the defined clauses, the organization's multi-layered modelled entities and their dependencies are transformed to *primitive* literals such as $threatExists$ to identify a threat at asset-level, $connectivity$ to represent connectivity between assets, $runsService$ to associate services to assets, $runsActivity$ to map a service to a mission activity, and finally $runsProcess$ to identify the business-process an activity belongs to. Also, it is important to point out that a clause's postcondition (*derived literal*) can be another's precondition. For instance, *Clause 3* has a precondition given by the literal $compromisedAsset(Asset)$, which can be derived by the *Clauses 1* and/or *2*. This cascading effect resulting of iteratively validate each clause with primitive and derived literals produces an attack graph step by step.

Example. To illustrate the approach, let us consider a small network consisting of four types of assets typically found on electric power infrastructures to control physical processes on a SCADA (Supervisory control and data acquisition) system (Figure 16): an HMI (Human Machine Interface), a control PLC (CPLC) to control other PLCs, a PLC and a IED (Intelligent Electronic Device) to control the closing and opening of a circuit breaker. Additionally, the *STRIDE* threat landscape being considered is also depicted in the figure, as well as how a simple business-process, "Power Supply", is being carried out by the PLC and IED.

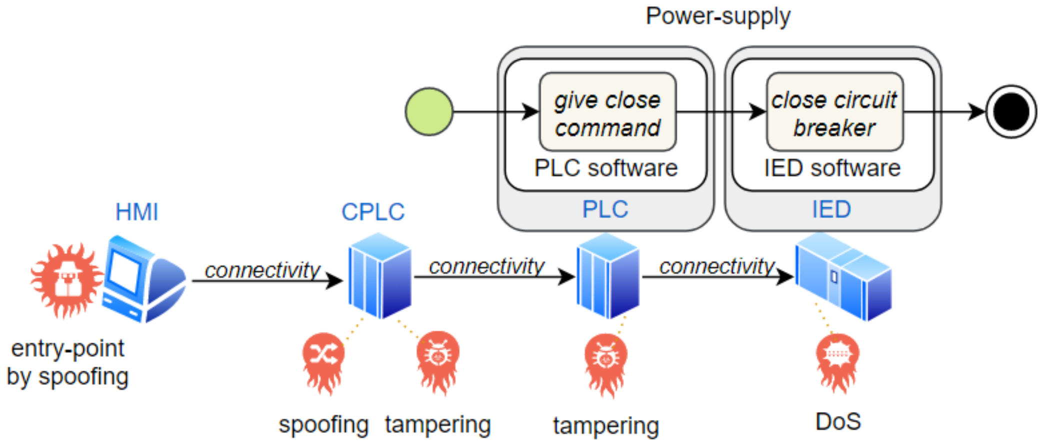


Figure 16 - Network example consisting of a SCADA, a CPLC, a PLC and a PLC.

The business-process becomes the goal of the propagation simulation and, together with the simulation's environment configuration, it can be represented by the primitive literals given in Table 4.

Table 4 - Primitive literals to define example's configuration.

Primitive Literal	Literal's Description
$threatExists(CPLC, Spoofing)$	CPLC has spoofing threat
$threatExists(CPLC, Tampering)$	CPLC has tampering threat
$threatExists(PLC, Tampering)$	PLC has a tampering threat
$threatExists(IED, DoS)$	IED has a DoS threat
$connectivity(HMI, CPLC)$	HMI communicates freely with CPLC
$connectivity(CPLC, PLC)$	CPLC communicates freely with PLC
$connectivity(PLC, IED)$	PLC communicates with IED
$runsService(PLC, PLC\ software)$	PLC runs *PLC software* service
$runsService(IED, IED\ software)$	IED runs *IED software* service
$runsActivity(PLC\ software, "Give\ close\ command")$	"Power Supply" service supports "Give close command" mission activity
$runsActivity(IED\ software, "Close\ circuit\ breaker")$	"Power Supply" service supports "Close circuit breaker" mission activity
$runsProcess("Give\ close\ command", "Power\ supply")$	"Give close command" activity provides "Power Supply" business-process
$runsProcess("Close\ circuit\ breaker", "Power\ supply")$	"Close circuit breaker" activity provides "Power Supply" business-process
$attackerGoal("Power\ supply")$	Defining the business-process "Power Supply" as the goal to reach in threat propagation

Starting the simulation from the entry-point defined as the HMI asset, associated with a spoofing threat, will add two more primitive literals (Table 5).

Table 5 – Conditions to define entry-point to be simulated.

Primitive Literal	Literal's Description
$threatExists(SCADA, Spoofing)$	SCADA has spoofing threat
$attackerLocated(SCADA)$	Active threat for propagation to be evaluated

To perform impact assessment of the spoofing threat affecting the HMI asset, the Threat Propagation module takes the initial literals representing the example's configuration under test (Table 4 and Table 5) and compares them to the preconditions of the propagation clauses previously defined in Table 3.

From the entry-point conditions, the preconditions of the first clause $compromisedAsset(Asset)$ are satisfied, deriving a new literal: $compromisedAsset(SCADA)$. Next, the simulation continues to explore the environment literals (primitive and derived) and, validating the clauses, tries to advance throughout the simulation environment until it reaches the mission. The resulting attack graph for this example is given by Figure 17.

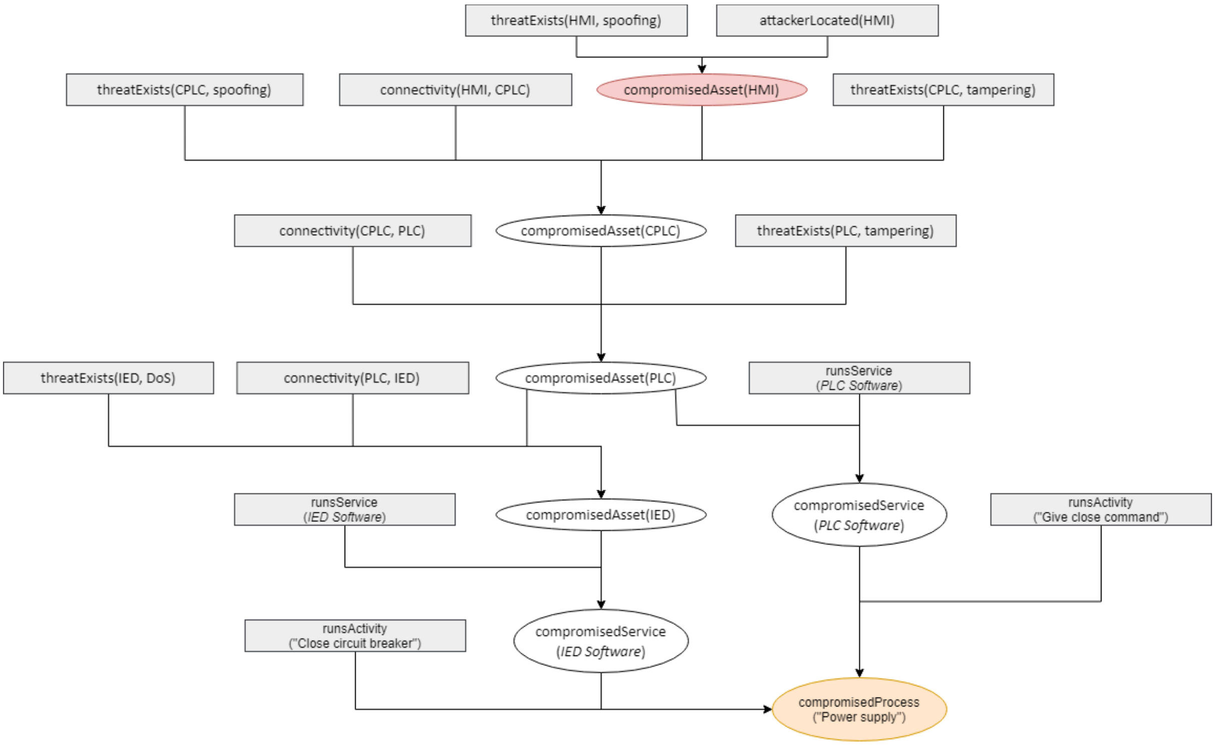


Figure 17 – Simulation attack graph example: primitive conditions (rectangle nodes) and derived conditions (ellipse nodes). Red node depicts the entry-point to the system and orange node represents the business-process reached.

The attack graph in Figure 17 illustrates how the simulation on the chosen entry-point reached the mission, the business-process “Power Supply” (orange node). Beginning with the entry-point conditions defined by the user, the simulation derives the first compromised asset as the HMI (red node) by the $compromisedAsset(Asset)$ clause. Combining this newly derived condition with the $connectivity(HMI, CPLC)$ condition, the $compromisedAsset(Asset2)$ clause is satisfied and a new condition is derived, which lets the simulation advance to the CPLC and identify it as compromised ($compromisedAsset(CPLC)$). This procedure of combining primitive conditions (rectangle nodes) and iteratively validating the clauses defined to produce derived conditions (ellipse nodes) is repeated until the simulation reaches the mission (orange node).

The resulting graph is then output by this module.

3.3.2. Impact Assessment

After the attack graph is constructed, a careful reading of the graph is necessary to understand relevant MIA information. The attack graph illustrated in Figure 17 is a readable example, however, simulating a more-realistic environment involving (several) more than the four assets and three allowed communications considered in the example, can quickly become difficult for a human to digest and identify key aspects of MIA. To address this issue, the attack graph should be processed by an *Impact Assessment* module to interpret the simulation's outcome and assemble it to produce a report on MIA, as outlined in Figure 15.

Hence, the attack graph produced by the Threat Propagation module is traversed by this module to identify the compromised assets and exploited threats, the explored connectivity among assets, and the business-processes compromised, and the propagation steps the simulation followed to advance throughout threatened assets towards the mission. This final analysis highlights relevant information and is presented in a report for impact assessment.

3.4. Summary

This chapter presented the structure and the basic concepts of the various functioning stages of the proposed solution, named BIA. In short, BIA tackles two important aspects of MIA by combining two stages in one solution.

The Setup stage addresses the *Impact Modelling* aspect of MIA, by proposing a four-layered evaluation model of four different abstraction levels: threat, asset, service and mission. To populate this model, BIA proposes to use packet captures to discover assets and firewall configurations to infer allowed connectivity by individual and hierarchical firewall policy to build the asset layer, threat's classification standards to identify the organization's threat landscape, and a business-process model combined with services and assets, from which to map the built asset layer to the service and mission layer.

The Simulation stage undertakes the *Impact Propagation* aspect of MIA. It defines Horn Clauses to represent how threat propagation should be tackled by a logic-based propagation methodology to create an attack graph. This stage takes the organization's environment to simulate, and, upon choosing an entry-point and exploited threat, starts the propagation simulation of the chosen threat impact throughout the organization's environment to impact the business-processes. A report is issued with relevant information pertaining the compromised assets, services, activities and processes, and the connectivity exploited to reach those.

4. Implementation

The solution outlined in Section 3 has been implemented into a single tool so that BIA's approach effectiveness could be tested in practice. This chapter discusses the implemented tool and which existing tools and previous work can be leveraged for certain components of the integrated overall approach to MIA, and the envisioned assumptions the proposed solution works under.

Following the two-stage architecture described in the previous chapter (review Figure 7), the technical implementation of BIA is also described in two sections: the Setup stage and the Simulation stage.

4.1. Setup Stage

The Setup stage's architecture was defined as a set of three knowledge units – *Topology Discovery*, *Threat Identification* and *Service and Mission Specification* – meant to mine data sources for cyber-infrastructure and mission profiling, and a knowledge database to collect this information using the layered-model, as described in Section 3.2. To consolidate the proposed assessment model in a data representation BIA employs *Neo4j*[6] database, a graph database that offers a data model optimized for graph operations to address the adopted multi-layer architecture and its vertical and horizontal dependencies.

Regarding the knowledge units used to populate the database, each unit leverages different data sources, which, evidently, will be processed differently. Therefore, according to the level of data granularity needed for each particular data source, the three knowledge units were implemented by five components combined: *Network Discovery* to process packet captures, *Connectivity Discovery* to infer missing connectivity, *Asset Classification* to give a user-chosen classification to each asset, *Threat Identification* to combine asset's classifications and identify asset's threats with a user-chosen threat classification and, finally, a *Service and Mission Specification* component to interact with BP-IDS [49] to receive business-process and service specification and map it to the discovered assets.

The components were implemented using a combination of *Python*[7] programming language due to its versatility, and *shell scripting* for its simplicity. To help the implementation of this components go smoothly, the following assumptions were made, and their limitations highlighted.

Assumptions.

1. The most relevant layer of the open systems interconnection model (OSI Model) to this work is the network layer (third layer) and it was the one considered to identify assets, meaning *Assets* encompass *Network Assets*, such as hosts and routers, characterised by their IP address(es). However, to concretize an inclusive model for future extensions, the evaluation model also identifies an *Asset* by a unique numerical identifier and associates a *Network Asset* entity to it.

[6] https://neo4j.com/

[7] https://python.org/

2. *Subnets* refers to IPv4 subnetworks and they have to be explicitly given to BIA's input. A subnet can have many assets belonging to it in which they can communicate freely.
3. A *router* connects subnets and acts as a firewall to restrict connectivity among the subnets it connects. Firewall policy must be fully defined as input. Routers can have multiple subnets and/or other routers as neighbours.
4. Connections between assets extracted from the packet capture are considered to use TCP or UDP communication protocol only.
5. Horizontal dependencies are present only at asset layer, via connectivity among assets.

Limitations of the assumptions.

1. Assets that do not transmit network packets during the observed time period on the packet capture will not be detected. This also goes for asset connectivity.
2. Assets not included by the defined subnets will not be affected by the firewalls' rules.
3. Host-based firewalls are not being considered which could further refine firewall effect on asset connectivity.
4. Captured communication packets that use communications protocols other than those based on TCP or UDP protocols will not be recognized as a possible connectivity between assets.
5. Since horizontal dependencies are only present at asset layer, the subsequent impact propagation calculation takes a more vertical approach between the model's layers.

The Setup's five components, their inputs and data flows with the knowledge database are outlined in Figure 18 and their implementation is described in the following five subsections.

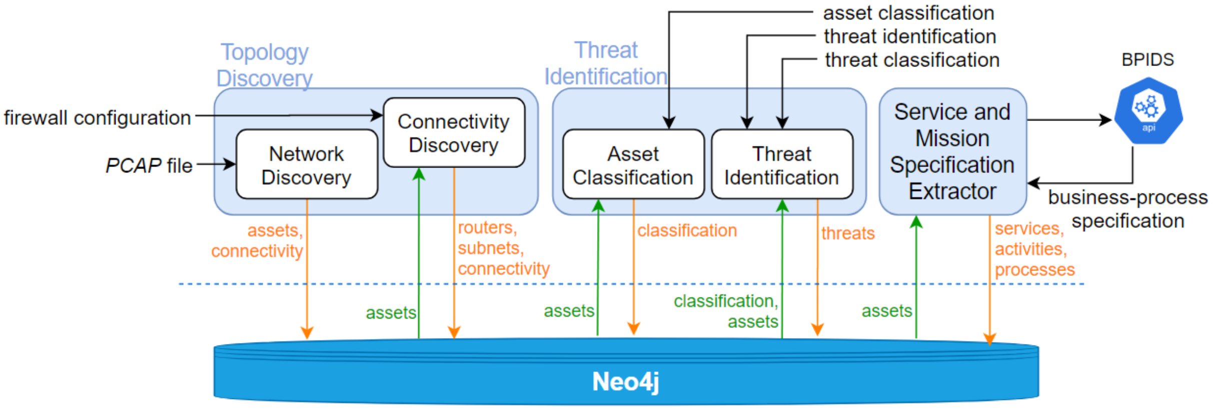


Figure 18 - Diagram of Setup stage's implementation and dataflows.

4.1.1. Network Discovery

The first component was architected to resort to a network analyser tool, specifically for inspecting network trace files (in *PCAP* format) to extract information about assets and their connectivity. To achieve this, a script was developed to call network protocol analyser tool *Tshark*[8], and then process its output as follows:

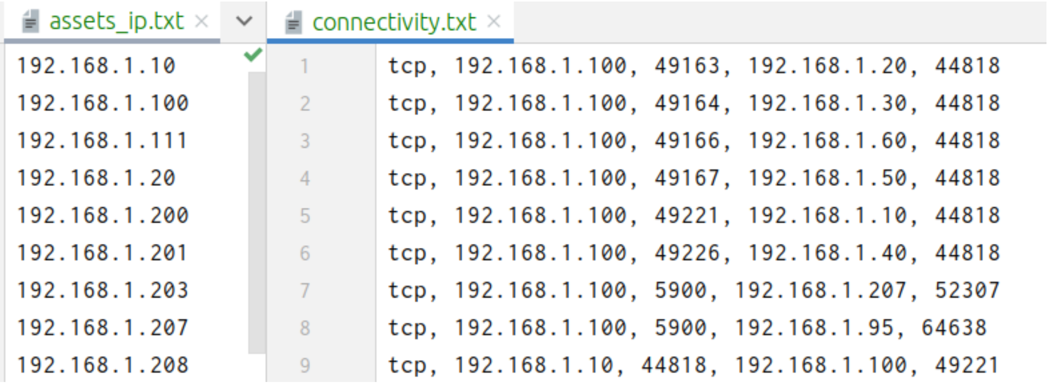
[8] https://www.wireshark.org/docs/man-pages/tshark.html

(1) First, the script invokes *Tshark* with a custom configuration, to analyse the network traffic recorded in the packet capture file. Any packet that contains TCP or UDP layer information is checked for its source and destination's IP address and the network ports used by that communication. The extracted information is stored in a data log file (in *CSV* format) with the format $< protocol, source_{ip}, source_{port}, destination_{ip}, destination_{port} >$ for each entry.

(2) Then, the script proceeds to order the data log file with the *Tshark* output, in order to remove duplicated entries and creates a new refined data log file with unique entries. This new result is then parsed to produce two files: one with a list of assets, identified by its IP address, $asset\ =< asset_{ip} >$, and another list with possible connections between assets, identified by $connection\ =< protocol, source_{ip}, source_{port}, destination_{ip}, destination_{port} >$ to represent asset connectivity. These two files are given as feedback to the user, as illustrated in Figure 19.

Afterwards, the Network Discovery component processes the lists of assets and connectivity and queries the graph database to upload this information, as nodes and edges between nodes, respectively, to represent the asset layer.

Additionally, the user can further edit the files to include (or exclude) assets and connectivity to be taken into (or out of) account by the MIA simulation.



```
192.168.1.10
192.168.1.100
192.168.1.111
192.168.1.20
192.168.1.200
192.168.1.201
192.168.1.203
192.168.1.207
192.168.1.208
```

```
tcp, 192.168.1.100, 49163, 192.168.1.20, 44818
tcp, 192.168.1.100, 49164, 192.168.1.30, 44818
tcp, 192.168.1.100, 49166, 192.168.1.60, 44818
tcp, 192.168.1.100, 49167, 192.168.1.50, 44818
tcp, 192.168.1.100, 49221, 192.168.1.10, 44818
tcp, 192.168.1.100, 49226, 192.168.1.40, 44818
tcp, 192.168.1.100, 5900, 192.168.1.207, 52307
tcp, 192.168.1.100, 5900, 192.168.1.95, 64638
tcp, 192.168.1.10, 44818, 192.168.1.100, 49221
```

Figure 19 - Example of the files created by the inspection of a packet capture. File ***assets_ip.txt*** *contain a list of discovered IP addresses, while* ***connectivity.txt*** *stores discovered connections between assets.*

4.1.2. Connectivity Discovery

To further consolidate the asset layer, the *Connectivity Discovery* component was developed to receive network infrastructure documentation that identifies existing firewalls and firewalls' domains (hosts protected by the same firewall) and firewall's policies (the list of rules that define what kind of traffic is allowed or blocked) to infer possible $connection$s missed by the Network Discovery and allowed by the firewall's policies in place. Motivated by *IPTABLES*[9] rule format, a firewall rule was modelled as $rule\ =< protocol,\ source_{ip},\ source_{port},\ destination_{ip},\ destination_{port},\ policy >$, where:

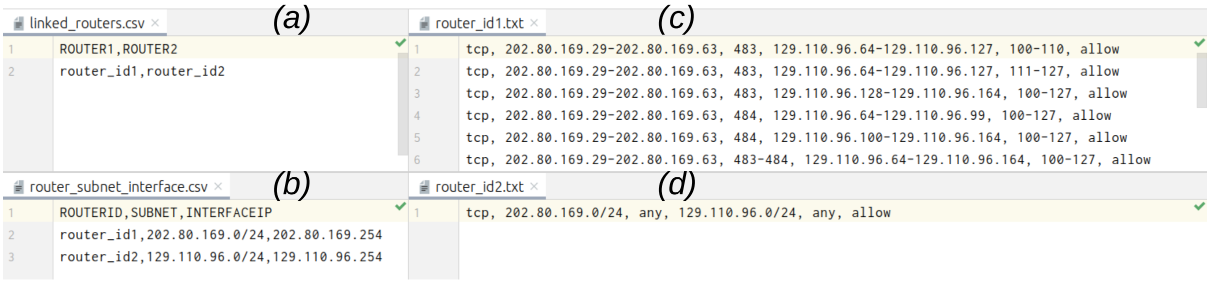
- $protocol$ may take the values $\{tcp, udp, any\}$;
- $source_{ip}$ and $destination_{ip}$ accept a single IP address, a range of IP addresses or a subnet IP address;

[9] https://linux.die.net/man/8/iptables

- $source_{port}$ and $destination_{port}$ receive a single integer or a range of integer;
- and $policy$ can take the value $allow$ or $deny$, to indicate if a communication related to this rule is allowed or blocked, respectively.

This model is also easily converted to the $connection$ representation presented in the Network Discovery component (previous Section 4.1.1).

The firewall configuration is given as input to this component by three different types of files, as can be seen in the example provided in Figure 20. A file is used to identify each router, define router domain (subnets connected to the router) and router's interfaces on each domain (Figure 20-*b*). Another file (Figure 20-*a*) is required to identify neighbour routers, and finally a file *per* router with the firewall policy (list of rules) is also given as input, as in Figure 20-*c* and Figure 20-*d*.



```
ROUTER1,ROUTER2
router_id1,router_id2
```

```
tcp, 202.80.169.29-202.80.169.63, 483, 129.110.96.64-129.110.96.127, 100-110, allow
tcp, 202.80.169.29-202.80.169.63, 483, 129.110.96.64-129.110.96.127, 111-127, allow
tcp, 202.80.169.29-202.80.169.63, 483, 129.110.96.128-129.110.96.164, 100-127, allow
tcp, 202.80.169.29-202.80.169.63, 484, 129.110.96.64-129.110.96.99, 100-127, allow
tcp, 202.80.169.29-202.80.169.63, 484, 129.110.96.100-129.110.96.164, 100-127, allow
tcp, 202.80.169.29-202.80.169.63, 483-484, 129.110.96.64-129.110.96.164, 100-127, allow
```

```
ROUTERID,SUBNET,INTERFACEIP
router_id1,202.80.169.0/24,202.80.169.254
router_id2,129.110.96.0/24,129.110.96.254
```

```
tcp, 202.80.169.0/24, any, 129.110.96.0/24, any, allow
```

Figure 20 - Example of an input to Connectivity Discovery component. (b) defines two routers, where each protect a single domain (subnet), (a) defines that both routers are neighbours; (c)-(d) depicts files with each firewall's configuration.

Upon receiving the required input, the Connectivity Discovery the component uses the proposed ***Comparing algorithm*** to first inspect each router object firewall's configuration to assess what communications are effectively allowed by firewalls when $allow$-rules are expurgated from their parts in common with $deny$-rules. This is achieved by a splitting algorithm DENY_SPLIT (Listing 1), inspired by previous work ([65]) that receives both rules, the field to compare and a new list to store the resulting rules. Recursively, for each field, DENY_SPLIT compares the rules and extracts only the disjoint part of the field according to the four possibilities previewed in Figure 11, creating new rules to be added to the resulting $allow$-rule list.

Listing 1 - DENY_SPLIT pseudo-algorithm

Algorithm. DENY_SPLIT(r: *Rule*, s: *Rule*, field: *STRING*, split_rules_list: *LIST*)

Variables. r:= deny rule, s:= accept rule, field:= field to be compared, split_rules_list:= empty list to store resulting rules

Goal. Produces a list of allowed rules with the parts in common with the $deny$ rules removed.

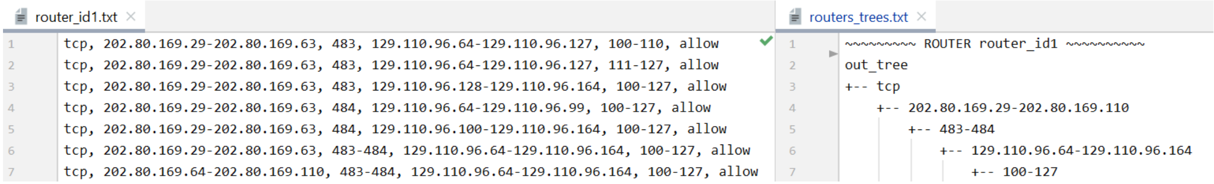
```
value1 := r.field
value2 := s.field
if field is POLICY then
    return
else if value1 = value2 then
    return DENY_SPLIT(r, s, next_field, split_rules_list)
endif

```

```
left := min(r.field.start, s.field.start)
right := max(r.field.end, s.field.end)
common_start := max(r.field.start, s.field.start)
common_end := min(r.field.end, s.field.end)

if r.field.start > s.field.start then
   left_rule := s
   left_rule.field := [left, common_start-1]
   split_rules_list.append(left_rule)
endif

if r.field.end < s.field.end then
   right_rule := s
   right_rule.field := [common_end+1, right]
   split_rules_list.append(right_rule)
endif

r.field := [common_start, common_end]
s.field := [common_start, common_end]
return DENY_SPLIT(r, s, next_field, split_rules_list)
```

When all *deny*-rules are compared with the original *allow*-rules, results a list with disjoint *allow*-rules only, the *allow*-list.

As described earlier, the firewall hierarchy environment also needs to be addressed. The proposed **Filtering Algorithm** was implemented by four main algorithms:

- The CLASSIFY_AND_INSERT_RULES (Listing 2) to classify each the rules on the *allow*-list and group and store in a tree data structure. In this way, each firewall will be characterised by four tree data structures – INTER, INSIDE, OUT(BOUND), IN(BOUND) – to represent the rules accordingly.
- Afterwards, to facilitate the traverse of the trees when processing the rules, the MERGE [65] algorithm is used to traverse each three and combine fields that have consecutive values. For an illustrative purpose, consider the seven rules defined on the left of Figure 21. MERGE combines all seven in only one rule (on the right), thus removing redundant rules and reducing the number of rules to be processed.



```
tcp, 202.80.169.29-202.80.169.63, 483, 129.110.96.64-129.110.96.127, 100-110, allow
tcp, 202.80.169.29-202.80.169.63, 483, 129.110.96.64-129.110.96.127, 111-127, allow
tcp, 202.80.169.29-202.80.169.63, 483, 129.110.96.128-129.110.96.164, 100-127, allow
tcp, 202.80.169.29-202.80.169.63, 484, 129.110.96.64-129.110.96.99, 100-127, allow
tcp, 202.80.169.29-202.80.169.63, 484, 129.110.96.100-129.110.96.164, 100-127, allow
tcp, 202.80.169.29-202.80.169.63, 483-484, 129.110.96.64-129.110.96.164, 100-127, allow
tcp, 202.80.169.64-202.80.169.110, 483-484, 129.110.96.64-129.110.96.164, 100-127, allow
```

```
~~~~~~~~~~ ROUTER router_id1 ~~~~~~~~~~~
out_tree
+-- tcp
    +-- 202.80.169.29-202.80.169.110
        +-- 483-484
            +-- 129.110.96.64-129.110.96.164
                +-- 100-127
```

Figure 21 - Example of output after MERGE algorithm.

- Then, the algorithm APPLY_INNER_RULES was created to process every router firewall rule trees. First, it creates *connection* representations between all hosts on the same subnet. Then it proceeds to traverse the router firewall's INSIDE-tree and create every connection represented there, since the router firewall is the sole decider for this type of rules. The OUT-tree is then processed, where APPLY_INNER_RULES takes each OUT-rule and propagates it to the neighbour routers.

- When a firewall receives a rule from another, it is filtered by the APPLY_OUTER_RULES algorithm that is responsible to inspect the destination of the rule and determine if it belongs inside or outside the firewall's domain (Listing 3):
    - If it is inside, it looks up the rule in the firewall's IN-tree, filters it accordingly (employing a similar reasoning as the DENY_SPLIT algorithm) and creates the correspondent *connection* in the database based on the resulting rules (lines 1 to 6).
    - If, on the other hand, the incoming rule's destination is outside of the firewall's domain, then the INTER -tree is used to filter the rule, and the result of the filtration is propagated to the next adjacent firewall/router.

Listing 2 - CLASSIFIY_AND_INSERT_RULES pseudo-algorithm.

Algorithm. CLASSIFY_AND_INSERT_RULES(firewall: *Firewall*, rules_list: *LIST*)

Variables. firewall:= current firewall, rules_list:= rules associated with this firewall

Goal. Classifies a rule and inserts it in the correspondent firewall's rule trees.

```
for all rule in rules_list do
    source ← compare source IP with firewall protected subnets and classify it
    destination ← compare destination IP with firewall protected subnets and classify it
    if source is OUTSIDE then
        if destination is OUTSIDE then
            root := firewall.inter_tree
        else if destination = INSIDE then
            root := firewall.in_tree
        endif
    else if source is INSIDE then
        if destination is OUTSIDE then
            root := firewall.out_tree
        else if destination = INSIDE then
            root := firewall.inside_tree
    endif
    INSERT_RULE(root, rule)
endfor
```

Listing 3 - APPLY_OUTER_RULES pseudo-algorithm.

Algorithm. APPLY_OUTER_RULES (firewall, neo4j_driver, rule)

Variables. firewall:= current firewall, neo4j_driver:= driver for neo4j database, rule:= rule sent by another firewall to be filtered by this firewall

Goal. Inspect if arriving rule's destination is inside or outside this firewall's domain and filter and/or apply it accordingly in the database.

```
1. if rule destination is INSIDE then
2.     accepted_in_rules ← new list
3.     LOOKUP_RULE_ON_TREE(firewall.in_tree, rule, 0, accepted_in_rules)
4.     for all in_rule in accepted_in_rules do
```

```
            create_connection(neo4j_driver, in_rule)
    endfor
else if rule destination is OUTSIDE then
    accepted_inter_rules ← new list
    LOOKUP_RULE_ON_TREE(firewall.inter_tree, rule, 0, accepted_inter_rules)
    for all inter_rule in accepted_inter_rules do
        for all neighbour in firewall.neighbors do
            neighbour.APPLY_OUTER_RULES(neo4j_driver, inter_rule)
        endfor
    endfor
endif
```

At the end, the Connectivity Discovery component adds $connection$ edges between assets nodes in the database, corresponding to communications inferred by allowed firewall policy. Additionally, this component gives two types of feedback to the user: a file with all inferred $connection$s of allowed asset connectivity, and a file with firewall policy's merged trees.

4.1.3. Asset Classification

The Asset Classification component aims to map assets to their types, given by the user as input by a file with a list of tuples in the format $< asset_{IP}, classification >$, and uploads that association to the database. For instance, for the example given in Section 4.1.1, those discovered assets can be classified with five different types: PLC, SCADA, Historian, Access Point and IED (Table 6).

Table 6 - Asset Classification input example.

$< asset_{IP}, classification >$	
192.168.1.10	PLC
192.168.1.100	SCADA
192.168.1.111	Historian
192.168.1.20	PLC
192.168.1.200	Access Point
192.168.1.201	IED
192.168.1.203	IED
192.168.1.207	IED
192.168.1.208	IED

4.1.4. Threat Identification

BIA's Threat Identification component follows the Asset Classification to map asset types to threats. It receives two inputs given as *CSV* files: one with a list of asset types mapped to threats, with the format $< asset\ type, threat\ description >$; and a second input, with those threat descriptions classified using

STRIDE[10] framework, with a list of $< threat\,description, STRIDE\,classification >$ tuples. *STRIDE* framework has six possible threat classifications: spoofing, tampering, repudiation, information disclosure, denial of service and escalation of privilege.

Following the previous example for asset types (in Table 6), a possible input for BIA to create a corresponding threat layer could be given by the tuples presented in Table 7.

Table 7 - Example of Threat's Identification input.

$< asset\,type, threat\,description >$		$< threat\,description, STRIDE\,classification >$	
IED	Distributed DoS	Distributed DoS	DoS
Access Point	Man-in-the-middle	Man-in-the-middle	Spoofing, Tampering
PLC	Malware	Malware	Spoofing, Tampering, Repudiation, Information disclosure, DoS, Elevation of privilege
Historian	Information Leakage	Information Leakage	Information disclosure
SCADA	Communication hijack	Communication hijack	Spoofing

In this way, BIA creates the threat layer with STRIDE classifications, and maps the threat layer to the asset layer in the database.

4.1.5. Service and Mission Specification

To gather information about the service and mission layer, a Service and Mission Specification component was created to interact with BP-IDS [49] database through its API to receive business-process information. The API returns business-process information in a *JSON*[11] format, which is retrieved reformatted and uploaded to the database to form the service and mission layer.

4.2. Simulation Stage

BIA's Simulation stage was conceived as a simulation platform that leverages MulVAL [86] to perform MIA. Two components were implemented to achieve this purpose: a *Threat Propagation* component to convert the proposed Horn Clauses to express threat propagation (defined in Section 3.3.1) into MulVAL's knowledge base. These clauses are then used by MulVAL as rules to be validated by the organization's evaluation model and produce an attack graph; and a second component, the Impact Assessment component to extract relevant MIA information from the attack graph and present it to the user.

Figure 22 outlines the implementation and dataflows of this stage. The following sections describe in detail the implementation of each component.

[10] https://www.microsoft.com/security/blog/2007/09/11/stride-chart/

[11] https://www.json.org/

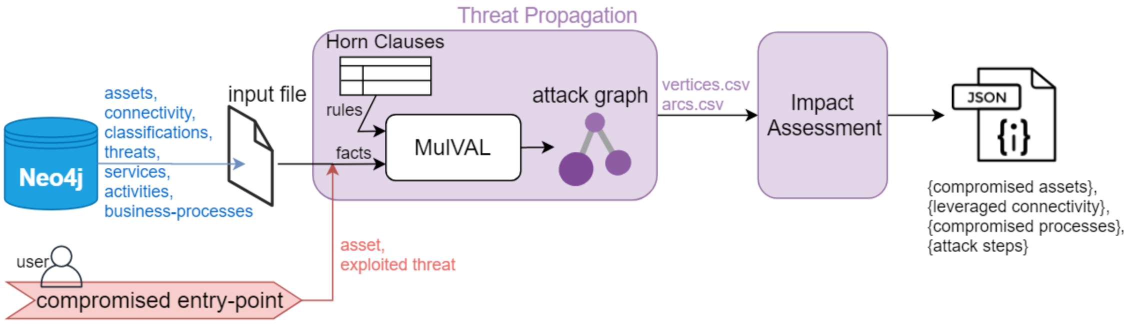


Figure 22 - Diagram of Simulation stage's implementation and dataflows.

4.2.1. Threat Propagation

To achieve the proposed approach for threat propagation, BIA leverages the attack graph tool MulVAL that acts as a processor of Datalog rules to generate attack graphs.

MulVAL's original rules do not consider a threat and mission layer, therefore BIA reformulates MulVAL knowledge base by expressing the proposed four Horn Clauses for threat propagation (previewed in Table 3) as new Datalog rules, implemented as a part of *interaction rules* in MulVAL, to be used by MulVAL to create an attack graph, that spans from a threat, to the asset, service and mission layer. The new interaction rules are presented in Listing 4.

Listing 4 - Horn Clauses for threat propagation described as interaction rules.

```
interaction_rule(
    (compromisedAsset(AssetDSTid):-
         compromisedAsset(AssetSRCid),
         interfaceHasAssetID(AssetSRCip, AssetSRCid),
         connectivity(Proto, AssetSRCip, PortSRC, AssetDSTip, PortDST),
         interfaceHasAssetID(AssetDSTip, AssetDSTid),
         threatExists(AssetDSTid, Threat)),
     rule_desc('Attack propagated to another asset', 'certain')
).

interaction_rule(
    (compromisedAsset(Assetid):-
        attackerLocated(Assetid),
        simulatedThreat(Assetid, Threat)),
    rule_desc('Entrypoint compromised', 'certain')
).

interaction_rule(
    (compromisedService(Assetid, ServiceID, ServicePort, ServiceName):-
        compromisedAsset(Assetid),
        runsService(Assetid, ServiceID, ServicePort, ServiceName)),
    rule_desc('Attack propagated to the applicational service', 'certain')
).

interaction_rule(
    (compromisedProcess(ProcessID, ProcessName):-
        compromisedService(AssetID, ServiceID, ServicePort, ServiceName),
        isBusinessAsset(AssetID, AssetType),
        runsActivity(ServiceID, ActivityID, ActivityName),
        runsProcess(ActivityID, ProcessID, ProcessName)),
     rule_desc('Attack propagated to the business-process', 'certain')
).
```

Interaction rules are based on *primitive* and *derived* facts to represent preconditions and postconditions, respectively, of Horn Clauses. BIA transforms the organization's infrastructure and mission profile, identified in the Setup stage, into primitive facts. MulVAL then applies the interaction rules towards the primitive facts and, if all preconditions are met, produces derived facts. All the facts (primitive and derived) considered by BIA are presented in Listing 5.

Listing 5 - Facts used to implement threat propagation rules.

```
primitive(threatExists(_asset, _threatID)).
primitive(simulatedThreat(_asset, _threatID)).
primitive(connectivity(_prot, _src, _srcport, _dst, _dstport)).
primitive(attackerLocated(_asset)).
primitive(runsService(_assetAddress, _serviceID, _servicePort, _serviceName)).
primitive(runsActivity(_serviceID, _activityID, _activityName)).
primitive(runsProcess(_activityID, _processID, _processName)).
primitive(isBusinessAsset(_assetID, _assetType)).
primitive(interfaceHasAssetID(_assetAddress, _assetID)).
derived(compromisedAsset(_asset)).
derived(compromisedService(_asset, _serviceID, _servicePort, _serviceName)).
derived(compromisedProcess(_processID, _processName)).
```

In addition to facts and rules, MulVAL requires an initial point to start its verification process, and a target to direct and conclude that process. BIA defines MulVAL's target as the business-processes identified in the Setup stage that MulVAL will try reach, while the initial point is provided as an external input to this component and defined as the entry-point to the system by the data tuple:

$$< asset, threat >$$

The entry-point is then transformed and combined with the rest of the primitive facts, which completes the required input to run MulVAL, and effectively triggers the start of the simulation. Furthermore, the entry-point is chosen by the user, which can choose to run the simulation several times for different entry-points independently of the Setup Stage. Here lies another main features of BIA's MulVAL extension, where every time the user chooses an entry-point, MulVAL's required input is automatically changed accordingly.

The attack graph generated by MulVAL is constructed by loading all this information (rules, primitive facts, entry-point and target) in the required format, into an XSB [95] execution engine to evaluate the interaction rules on the environment facts (primitive and derived). The resulting attack graph is output by MulVAL in *PDF* format (optional), together with two *CVS* files, one with the nodes and the other with all edges present in the attack graph, and a *TXT* file with all this information combined. Since the graphically representation of the attack graph (in *PDF*) often results in an image difficult to digest at naked-eye, and it is the option that takes longer to produce results, BIA's Threat Propagation component only outputs the two *CSV* files for the next component, to assess relevant MIA information.

4.2.2. Impact Assessment

When performing MIA, often users want to quickly assess which organization's business-processes are impacted. A further analysis may then be required to understand how the attack may have propagate through the organization's infrastructures. Hence, this component is implemented to parse the attack graph produced by the previous component and retrieve relevant information about the compromised performers, and the threats and connectivity exploited to that effect. This information is then presented to the user in *JSON* format, for its readability, and versatility to be further extended and integrated.

4.3. Summary

This chapter has presented how BIA's implementation was designed to embody the proposed approach for MIA and the assumptions it runs under.

The Setup stage's implementation, in short, revolves around the concretization of the proposed evaluation model using the *Neo4j* graphical database. Five components were implemented to process five different sources: (1) packet captures, leveraging *Tshark*, (2) firewall configuration in *IPTABLES* format, by implementing two algorithms to tackle both individual and hierarchical firewall policy, to retrieve allowed asset connectivity, (3) asset's classification, to classify discovered assets according to their purpose, (4) threat's classifications and their *STRIDE* counterparts, and (5) BP-IDS database, by a component designed to interact with its API. The stage's implementation concludes with a script to create a file with the organization's profile in the formatted representation required for simulation.

The Simulation stage's implementation is essentially based around the MulVAL attack graph tool. The Horn Clauses previously proposed to express threat propagation are implemented to reformulate MulVAL original knowledge base as new MulVAL's *interaction rules* to be used to create the attack graph, as rules' pre and postconditions are being verified. The user is able to choose the entry-point (an asset and exploited threat) he/she wishes to simulate the impact, which prompts the simulation to start. Upon the given entry-point, MulVAL constructs an attack graph that tries to find impact propagation paths to reach business-processes. If the simulation is successful in finding attack paths to any business-process, the attack graph is parsed to retrieve information about the impacted mission performers, the exploited threats and leveraged asset connectivity and presents the information in a *JSON* report.

Together, the tool aims to offer all these features without requiring any technical knowledge from the user about the integrated tools, by conducting the interaction between the user and BIA with standard input and output files, such as *TXT*, *CSV* and *JSON*.

5. Evaluation

This chapter presents a series of experiments conducted to study BIA's efficacy in constructing a multi-layered mission-oriented evaluation model to perform mission impact assessment. To do so, the evaluation process focus on applying BIA to different case-studies developed upon a testbed that mimics a real-world power system in a scaled-down replica. This dataset is detailed in Section 5.1, along with additional settings required to setup an evaluation testbed for BIA. In Section 5.2, the developed case-studies are detailed, followed by an analysis of the results, while evaluating both BIA's key features and limitations. Lastly, the evaluation process and results are summarized in Section 5.3.

5.1. Evaluation Setup

BIA was deployed in an Ubuntu 18 virtual machine with 9 GB of RAM and 100% access to the resources of two of the four cores of an Intel Core i7-7500U CPU 2.70-2.90 GHz processor, where a series of experiments were conducted on an ICS dataset called EPIC (Electric Power Intelligent Control) from iTrust labs[12] that contains the essential elements of a fully operational infrastructure for power supply in a scaled-down replica capable of generating up to 72kVA power.

EPIC's network architecture is illustrated in Figure 23 consists of four distinct physical process stages: Generation, Transmission, Micro-grid, and Smart-home.

There is a total of 36 assets that are classified by five types: one SCADA workstation (SCADA WS), Programmable Logic Controllers (PLCs), Intelligent Electronic Devices (IEDs), Access points (APs) and Switches (SWs). All assets are prefixed with G, T, S and M, respectively, for Generation, Transmission, Smart-home and Micro-grid stages. For instance, the PLC in Generation is represented as GPLC. These four stages are connected to a control network (with C prefix).

In essence, the master PLC (CPLC) is responsible for the control of the overall operation and the other PLCs. The stages PLCs issue commands to the IEDs to close or open circuit breakers (CBs). The SCADA workstation (WS) is used to monitor the entire system and provides supervisory control, and the Historian collects and stores the physical-processes data [96]. The operator must choose between using either a wired mode or wireless mode of communication. The data collected from the EPIC testbed consists of 8 scenarios under normal operation, where for each scenario, the facility is running for about 30 minutes during which network traffic was saved in packet capture files (*PCAP* format).

Besides asset documentation, asset classification and packet capture files provided by EPIC, BIA requires an additional set of information for minimal functionality: the organization's mission and threat landscape the user desires to simulate. The next sections describe the remainder of the evaluation setup where three business-processes (BPs) are formulated according to EPIC's functionality and goals, and the threats and main entry-point to impact the organization are defined according to previous work on the subject.

[12] https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_epic/

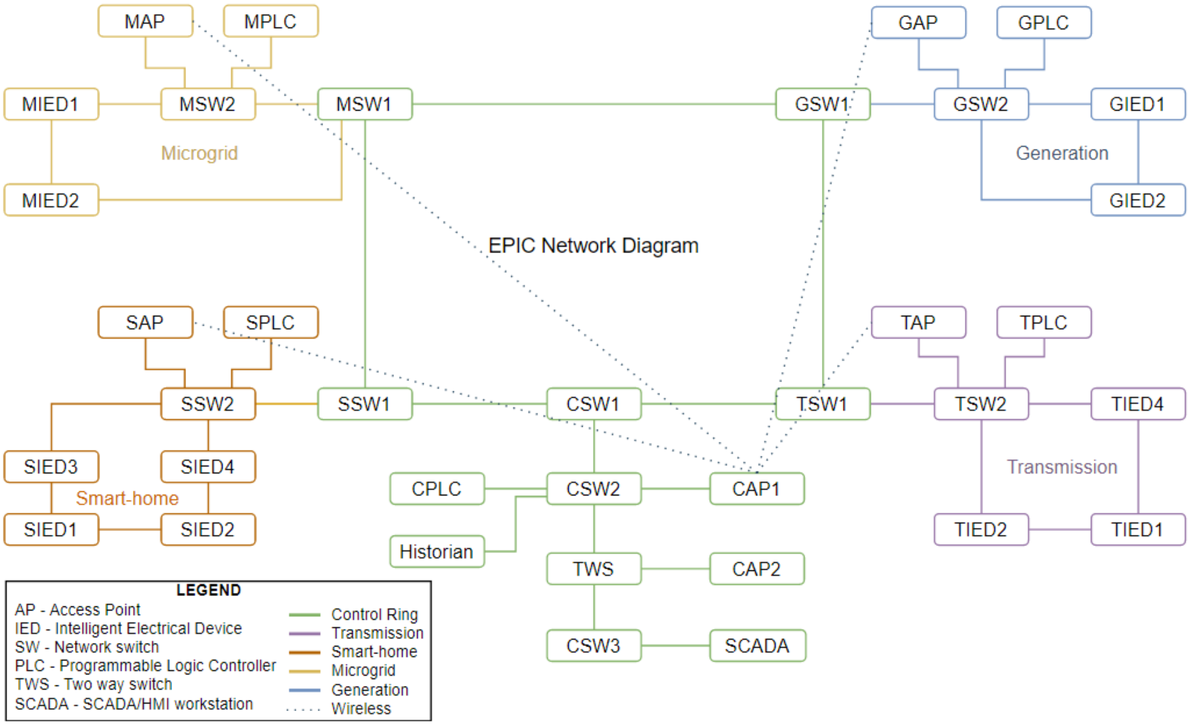


Figure 23 - EPIC's Network Diagram.

5.1.1. Mission

BIA was envisioned to interact with BP-IDS database to retrieve mission and service information mapped onto the asset layer, but it also offers an alternative, where BIA is called upon a user's file containing mission specification instead. The user can define the organization's mission manually by following the results returned by BP-IDS *REST* API (in *JSON* format).

BIA is being evaluated in SATIE[13] project in a real operational scenario at Zagreb airport, where BP-IDS stored data are EU classified information, hence, for the purpose of BIA's evaluation the manual alternative is employed to define and edit the mission as desired for the evaluation and simulation on EPIC's dataset.

Research on attack scenarios on EPIC [96] has found a *Power supply interruption* attack to be feasible, where false data injection attacks on SCADA and PLC system can lead to power supply interruption or tripping the overall system. Taking into account EPIC's undergoing physical processes and running software ([96], [97]), let us propose a sample of 3 realistic business-objectives that could be a part of EPIC's mission and affected by the aforementioned attack scenario, described as follows and depicted in Figure 24.

Power supply to smart-home. This business-objective, as the name suggests, aims to supply electrical power to load banks at the smart-home stage of EPIC. To achieve this, it is suggested that SCADA sends a command to close (1st activity) the circuit breaker that was open and interrupting the current flow. This command is sent to the SPLC (2nd activity) that in turn sends it to the IED responsible for the circuit breaker (3rd activity).

[13] H2020 project nr. 832969: http://satie-h2020.eu

Power supply in grid-connected mode. EPIC's generators can produce the power required for the system along with power drawn directly from the main grid [97], in grid-connected mode, whereas in the islanded mode only the local generators supply power to meet the demand and grid connection is disabled [96]. Accordingly, a business-goal is defined to operate the system in grid-connected mode, and tree activities are suggested following the same rationale as before: (1) SCADA sends GPLC a close command of the main circuit breaker to connected EPIC with the main grid, (2) GPLC sends that command to GIED1 responsible for the main circuit breaker and (3) in turn, GIED1 closes the circuit breaker.

Read electrical voltage. The Transmission stage is representative of a distribution grid, supplying power to Smart-home stage. A transformer is used to control the voltage to the Smart-home. A business-process to read the voltage current value is proposed, where CPLC sends a read request to TPLC (1st activity) that sends the request to the TIEDs (2nd activity).

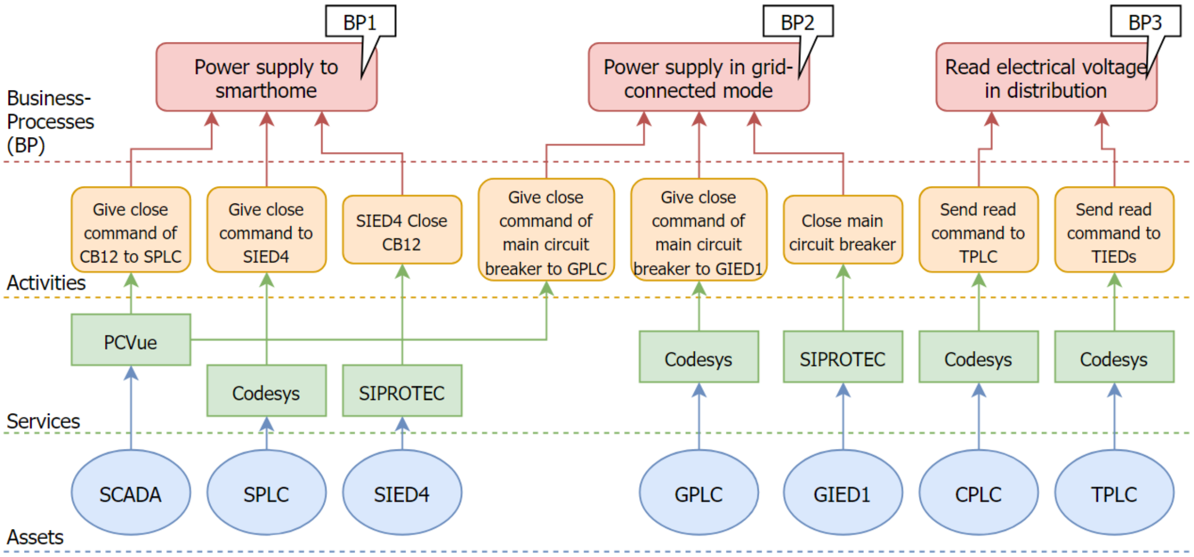

BP1
BP2
BP3
Business-Processes (BP)
Power supply to smarthome
Power supply in grid-connected mode
Read electrical voltage in distribution
Activities
Give close command of CB12 to SPLC
Give close command to SIED4
SIED4 Close CB12
Give close command of main circuit breaker to GPLC
Give close command of main circuit breaker to GIED1
Close main circuit breaker
Send read command to TPLC
Send read command to TIEDs
Services
PCVue
Codesys
SIPROTEC
Codesys
SIPROTEC
Codesys
Codesys
Assets
SCADA
SPLC
SIED4
GPLC
GIED1
CPLC
TPLC

Figure 24 – Mission (BPs and activities) and service specification mapped onto the asset layer for BIA's evaluation.

In total, the mission layer is thus comprised of 8 activities providing 3 BPs (BP1, BP2 and BP3) and mapped to 7 different services running on 7 different assets, as depicted in Figure 24. For evaluation purposes, only services supporting mission activities are considered in the service layer, however, a detailed description of EPIC services can be found in previous work ([96], [97]). Additionally, Figure 24 represented assets are referred to as *mission assets*, to distinguish assets partaking in the mission from the ones that do not

5.1.2. Threat landscape

Unlike for the mission layer, there is no benchmark of threats affecting EPIC dataset, however, previous research on feasible attacks [96] and emulated threat scenarios [97] on EPIC and research on typical threats affecting ICSs [98] can be leveraged to define some possible scenarios for threats present on the testbed. Hence, a brief rationale for threat distribution in EPIC testbed for BIA's evaluation is presented next, and is summarized in Table 8 as follows.

Table 8 - Threat landscape defined for evaluation.

Threat / Asset	Spoofing	Tampering	Repudiation	Information disclosure	DoS	Elevation of privilege
IEDs	-	-	-	-	✓	-
PLCs	-	✓	-	-	-	✓
Historian	-	✓	-	✓	-	-
APs	✓	✓	-	✓	-	-
SCADA WS	✓	✓	✓	✓	✓	✓
Routers/SWs	-	-	-	-	-	-

Research [97] shows that, due to the lack of security of MMS (Manufacturing Message Specification) protocol used in EPIC it is possible to perform spoofing attacks and to inject malicious messages in the SCADA WS that could mislead the SCADA system to initiate wrong control commands. Additionally, the SCADA WS includes the HMI software that can be remotely controlled by an attacker to send out a large number of circuit breaker open commands [97]. Furthermore, on SCADA WS in EPIC, it is also possible to evaluate the impact of attacks caused by malware, such as CrashOverride [99]. From all these possibilities it can be concluded that the SCADA WS can be compromised by various types of threats, hence, it will be mapped to all STRIDE threat classifications. Additionally, manipulated sensor readings (e.g., by spoofing sensors) also would lower the situation awareness of SCADA, and compromise the integrity of the information stored in the Historian (as in a *tampering* threat).

EPIC's PLCs models are known to have vulnerability to allow attackers to modify or delete arbitrary files [97], so, PLCs will be mapped to a *tampering* threat. Malware can also be mounted on PLCs to obtain administrative access to the PLC logic thus enabling modification of the control logic, which can be associated (but not exclusively) with an *elevation of privilege* threat.

Since IEDs are programmed to execute certain functions depending on the data on the others (sent by multicast messages), they can be exposed to a *distributed DoS* threat [98].

There are no firewall rules configured in the default setting of the router before SCADA server. As a result, if an attacker is able to associate to the access point (e.g., by guessing credentials), link layer attacks such as ARP (Address Resolution Protocol) spoofing are possible for most devices by anyone connected to the main network [97]. Accordingly, APs are here considered to be exposed to *spoofing* and *tampering* threats, and the Historian with an *information disclosure* threat.

Finally, CSW1 is considered to be a single point of failure for all communication between process stages, the Historian and SCADA [97]. Hence, an exploited threat in this type of assets can lead to the compromise of the entire organization. For evaluation purposes, let us consider SWs and routers are not exposed to any threats, to be able to simulate the impact propagation of other threats.

5.1.3. Entry-point

In the previous chapter it was defined that the SCADA WS was the asset exposed to the most threats, including all 6 STRIDE classifications, as it was the asset most threat scenarios focus on [97].

This reasoning is also supported by research in threats affecting ICS [98] where experts' feedback has agreed that one of the most affected asset is the HMI. Hence, the SCADA WS (running HMI software) will be considered as the entry-point on BIA's simulation, given as a $< SCADA, tampering >$ tuple, to evaluate how tampering with SCADA WS can impact the BPs defined in section 5.1.1.

Finally, with the entry-point defined for mission impact simulation, the evaluation setup can be summarized as follows in Table 9.

Table 9 - Evaluation setup overview.

# EPIC assets	# threatened assets	# services	# activities	# BPs	Entry-point
36	24	7	8	3	SCADA/HMI by tampering

5.2. Evaluation Process

This section focusses on the evaluation conducted on BIA to determine its capability on performing MIA. Therefore, in the next sections, a series of experiments is conducted to test and analyse BIA's results driven by the following research questions:

1. *How does BIA perform MIA?* Section 5.2.1 evaluates BIA's Network Discovery capability on extracting assets and their connectivity from packet captures and how accurate the resulting MIA is.
2. *How BIA's Connectivity Discovery aids in MIA?* To study the influence of firewall policy in the overall MIA result, two routers acting as firewalls and their policies are added to the testbed in Section 5.2.2.
3. *How BIA's result addresses organizational changes?* In the real world, an organization's cyber infrastructure is not static. Dynamic changes are bound to happen and can be simulated by BIA to perform MIA, by re-running BIA after changes to its input. Therefore, in Section 5.2.3.1 BIA is applied for a new threat landscape; Section 5.2.3.2 presents a case-study based on a update to the mission layer; finally, in Section 5.2.3.3 BIA is applied to simulate mission impact for another entry-point.

For each case, MIA report statistics are presented about what could be impacted, and how that impact propagates, and a detailed analysis of the results and their practicality is provided.

5.2.1. Topology Discovery

The organization's topology and connectivity constitute the asset layer of the organization's profile, and are firstly handled by the Network Discovery component, which evaluation will focus on its accuracy on discovering assets, and the influence of the organization's connectivity on MIA.

5.2.1.1. Asset discovery

A first series of experiments is undertaken to evaluate the overall accuracy of this component on discovering the organization's infrastructure topology (assets). To do so, BIA's Network Discovery component was tested for the packet capture files of all EPIC's 8 scenarios and the obtained results are presented in Figure 25.

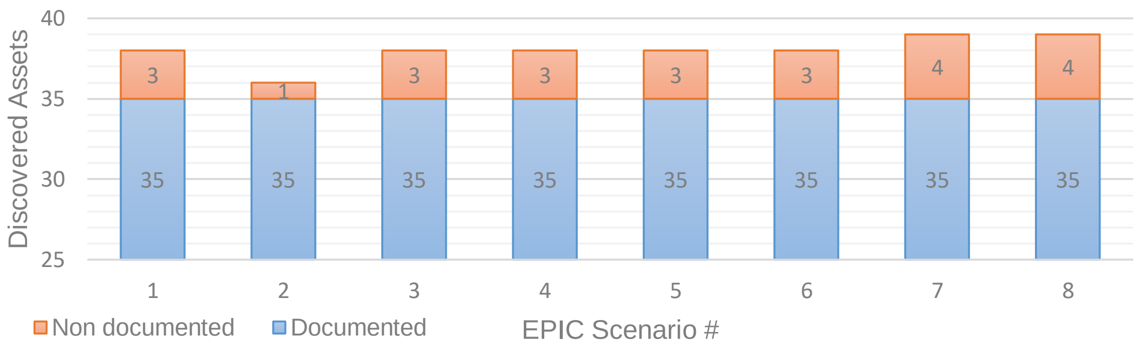


Figure 25 - Accuracy evaluation of Network Discovery.

Taking into account that EPIC documents a total of 36 assets, from Figure 25 it is possible to conclude that the Network Discovery component is successful in discovering 97.2% of the infrastructure's assets: for all scenarios 35 out of 36 documented assets were discovered. The asset that was systematically unobserved for all scenarios is associated with the IP address 172.16.6.1, that EPIC reports as the default gateway for the master PLC (CPLC) to connect to SCADA WS, and, as such it does not appear in the network capture.

Moreover, this component also reported undocumented assets for all scenarios, namely assets respective to the IP addresses 224.0.0.252, 237.1.2.19, 239.255.255.250 and 172.16.8.12.

The first three belong to known IP multicast address ranges[14] and, as expected, are not associated with a specific asset. The fourth undocumented IP address may be associated with external operators' laptops, or assets from the other two testbeds EPIC was designed to power – *SWaT*, a scaled-down water treatment plant, and *WADI*, a scaled-down water distribution network. In the same way, this external IP address may also belong to an attacker that has successfully intruded the internal network. Either way, feedback on the discovered assets is given to the user to allow the validation of the organization topology being modelled.

5.2.1.2. Asset connectivity discovery

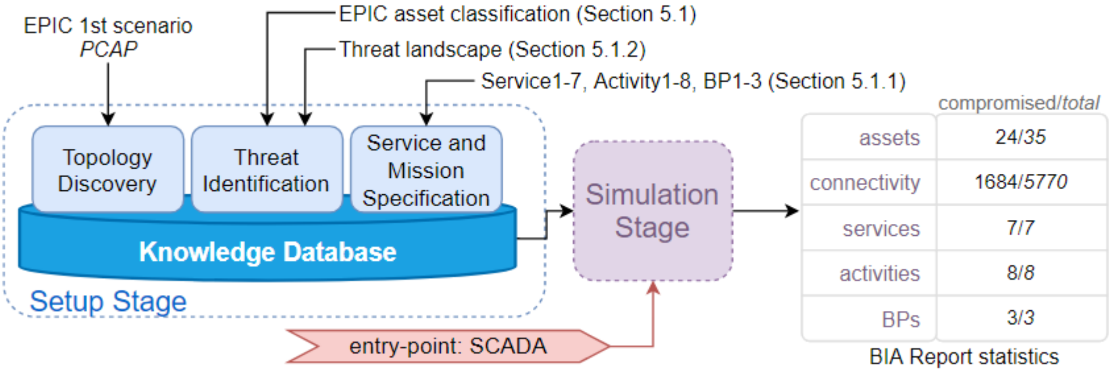
As a key precondition for BIA's threat propagation methodology, asset connectivity is expected to significantly influence BIA's results. Hence, to study this influence, BIA was validated using EPIC's first scenario packet capture (since it was seen the number of documented assets discovered is the same for every packet capture). The respective packet capture file contains 449 177 packets, from which BIA's Network Discovery component extracts 5770 unique $connections$ entries to represent EPIC asset connectivity. Although 5770 $connection$s entries are a significantly refined number compared to the number of packets recorded in the packet capture file, it is a high number that can condition BIA's simulated impact to propagate everywhere and result in a rough report for MIA, especially when considering EPIC's topology consists of 35 assets.

This vast connectivity panorama extracted from the packet capture can be explained by the presence of ephemeral network ports in the client side of client-server type of exchanges, to connect with a well-known port in the server side. Each newly allocated port will create a new $connection$ entry.

[14] Internet Assigned Numbers Authority (IANA) RFC 5771 guideline

As this happens for several communications, asset connectivity is increasing as new ports are used for the same asset-to-asset connectivity. Additionally, APs also allocate ports on the device for each connection they redirect among all assets they intend to connect. A particular example of this is from GAP, which has 123 different $connection$ entries caused by different ports, to a unique port on SCADA.

Before describing how this issue can be mitigated, let us simulation mission impact using EPIC's packet capture extracted connectivity as is, and the evaluation setup described earlier in Section 5.1. The corresponding simulation environment is illustrated in Figure 26, as well as the summarized statistics of the report issued by BIA (on the right).



	compromised/*total*
assets	24/35
connectivity	1684/5770
services	7/7
activities	8/8
BPs	3/3

Figure 26 - Simulation #1 – MIA for EPIC packet capture.

The report shows that all 24 threatened assets can be impacted, leveraging 1683 $connection$s, which results in a fully impacted mission, where all BPs, activities and services can be compromised. To understand how this result comes to be, a detailed analysis of the report shows how the impact propagated and is illustrated in Figure 27.

Simulation #1's report shows that the impact at entry-point directly propagates to all threatened assets, including all mission asset (block coloured nodes in Figure 27), therefore BIA report considers a mission fully impacted.

This indicates SCADA directly connects with all other assets, and a further analysis of the impact propagation shows there is no propagation among other assets, which is not consistent with EPIC's known network architecture, where it is known other assets communicate with each other. This suggests packet capture was done at SCADA level, and not at a network device which typically intercepts more communication's packets. Effectively, an analysis of connectivity extracted from EPIC's packet capture shows there are only two type of communications: *coming from* or *received by* the SCADA.

Another important aspect to note is that Microgrid assets are being reported as part of propagation paths (and being impacted themselves) even though they do not belong to any BP. This happens because they communicate back with SCADA that directly supports BP1 and BP2. This propagation behaviour causes *propagation cycles,* as the impact propagates to already impacted assets, and, combined with the vast asset connectivity panorama being considered, contributes to a report where the entire infrastructure and mission can be impacted. These propagation cycles constitute BIA's threat propagation methodology main limitation and should be addressed by future work, however, while the user may be interested to see a more refined result, this result helps to understand the numerous ways the user may not be aware his system could be compromised and, in fact, may represent a real event (if the attacker has the proper conditions).

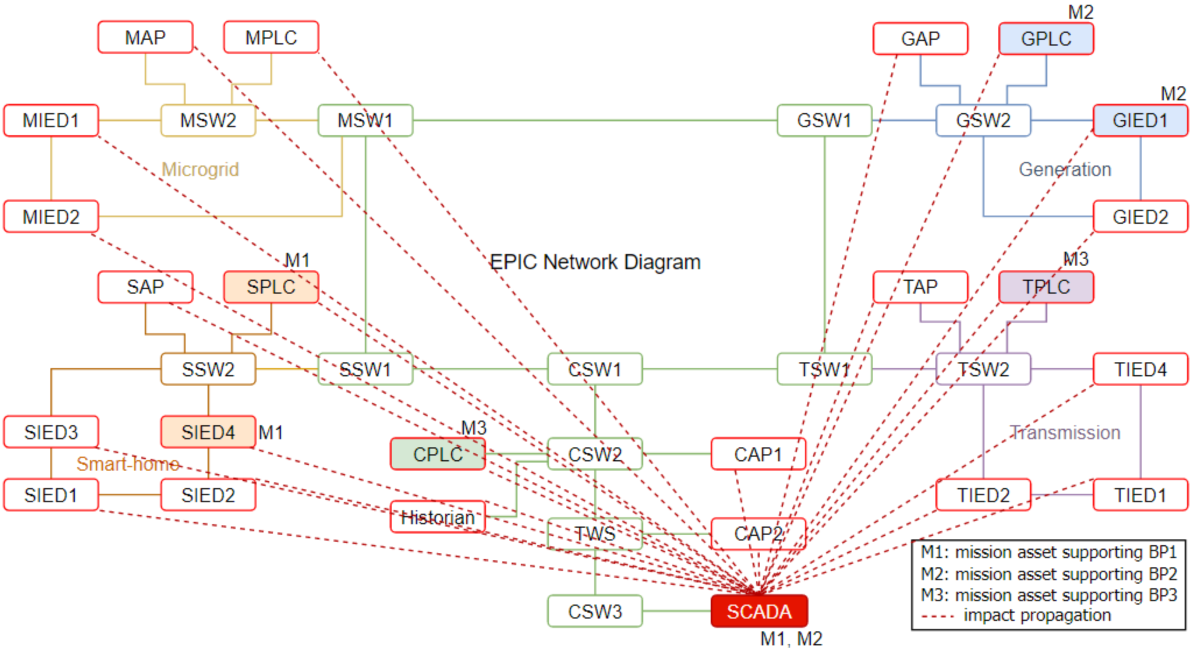


Figure 27 - Simulation #1 impact propagation. Block coloured nodes represent mission assets.

Nonetheless, this tool was implemented to populate the evaluation model in a semi-automatic way, enabling the user to validate the modelled dependencies, which can be edited as desired by reformulating the input files given to BIA. Hence, one way to refine the report's results is to define how asset connectivity should be taken into account for simulation. This can be achieved by editing Network Discovery connectivity input according to different solutions:

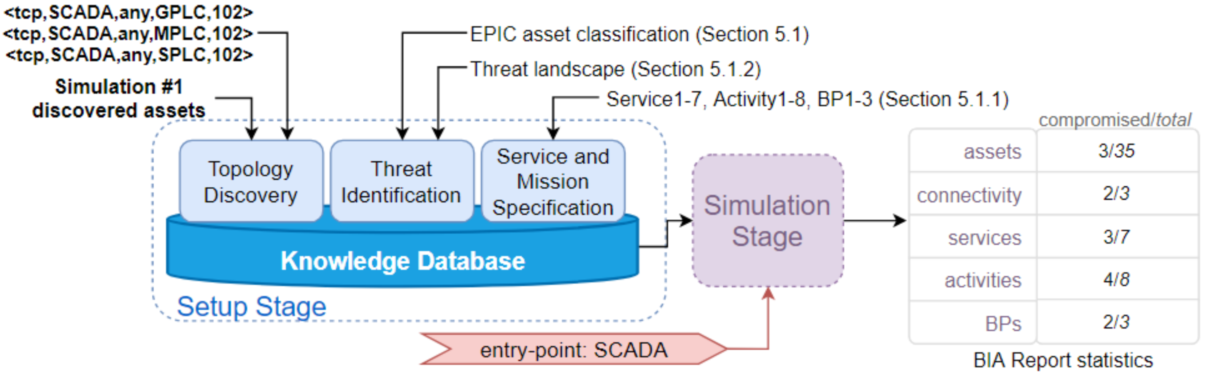
1) *Merge ephemeral ports*. Merging communications by a range of possible ephemeral ports (*Internet Assigned Numbers Authority*[15] suggests range 49152 to 65535) for every client-server type of communication can reduce significantly the number of $connections$ entries loaded to the simulation platform. Carrying out this suggestion the previously extracted 5770 $connections$ are reduced to 75 entries.
2) *Leverage unidirectional connectivity*. If one wants to simulate how the impact spans from a determined compromised entry-point, connectivity with destination back to the entry-point could be disregarded to reduce propagation cycles. This can be done manually by editing BIA's input or adjusting *Tshark*'s search filters used by Network Discovery component. To undertake this suggestion, *Tshark*'s filters *mms.confirmedServiceRequest* and *mms.confirmedServiceResponse* can be used to distinguish requests from responses sent by SCADA to interact with the physical processes.

Upon implementing these solutions, a second simulation is undertaken to study mission impact of compromised control messages sent by the SCADA requesting PLCs to change physical values – *write* commands. As *write* requests sent from SCADA WS are responsible for different control commands to

[15] https://www.iana.org/

SCADA system (to open/close circuit breakers, increase/decrease voltage, use/charge batteries, etc.), they can be considered as the target connectivity to be compromised at SCADA by the attacker [94]. Hence studying mission impact upon this type of connectivity can help refine last simulation's result.

On that account, analysing the packet capture shows 3 *write* commands recorded from SCADA to 3 PLCs (GPLC, MPLC and SPLC), and will be used as input for Topology Discovery component, along with the assets discovered from the packet capture and the remainder of the previous setup, as illustrated by Figure 28.



assets	3/35
connectivity	2/3
services	3/7
activities	4/8
BPs	2/3

Figure 28 – Simulation #2 - MIA for write *requests sent by SCADA.*

Simulation #2 shows BIA's MIA report where 3 assets could be compromised in total: the SCADA itself and two PLCs (SPLC and GPLC). These 3 compromised assets together are running 3 different services and 4 activities, which is consistent with the number of services and activities the simulation deems susceptible of being compromised/impacted. The impact propagation is outlined in Figure 29 as follows.

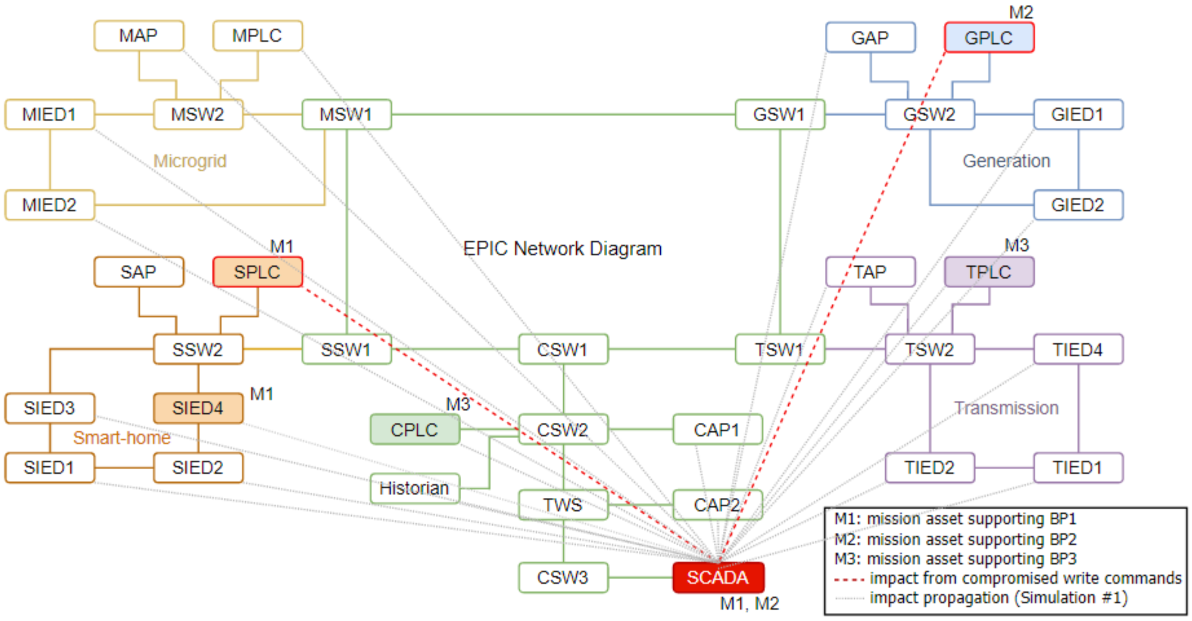


Figure 29 - Simulation #2 impact propagation.

This figure shows how BP1 and BP2 can be impacted from compromised *write* requests originated in SCADA: as entry-point and mission asset for BP1 and BP2, SCADA can directly impact those BPs. Additionally, the impact propagates to SPLC that contributes to BP1 impact, whereas an impacted GPLC contributes for BP2 impact. Contrarily, BP3 is not impacted because the impact from the entry-point does not reach the TPLC and the CPLC supporting that BP.

However, it is to be emphasized that it is known SCADA does send *write* requests to other PLCs, but EPIC's packet capture only captured this type of request to 3 PLCs, in the time period captured. Furthermore, it is also known that PLCs communicate with IEDs in the same process stage and IEDs communicate with each other, among other known communications. Since EPIC's packet capture was done at SCADA WS, none of these types of communications were captured. To address this issue BIA's Connectivity Discovery component can be leveraged to consider firewall policy to infer new connectivity among the testbed assets and provide a more accurate overview of MIA.

5.2.2. Connectivity Discovery

In the previous simulations it was seen how BIA aids in discovering the organization's topology and connectivity from parsing packet captures, and how it affects the MIA result. At the same time, relying exclusively on the captured communications to profile the organization's connectivity raised some issues:

- A vast connectivity panorama accentuates propagation cycles that leads the impact to propagate to every possible exploitable asset and, subsequently, to the mission.
- Assets that do not transmit network packets during the observed time period on the packet capture will not be detected. This also happens for communications that did not occur during that time. For instance, it is was highlighted that the SCADA sends "write" commands to all 5 PLCs, but this type of commands was only detected to 3 PLCs.
- Furthermore, since the packet capture was done at the SCADA workstation, only communications coming *from* or *to* the SCADA were taken into account. This, however, does not reflect the typical connectivity that does exist within a network with multiple assets where additional cross-communication exist. This is indeed the case for EPIC, where it is known that the PLCs also communicate with IEDs trough MMS, among others.

The aforementioned issues can be mitigated by leveraging BIA's Connectivity Discovery component to infer asset connectivity, either exclusively from firewall policy, or in addition to the connectivity already discovered. To study how this feature influences BIA's MIA result, two routers with firewall functionality are introduced to the testbed to create the following high-level network infrastructure depicted in Figure 30.

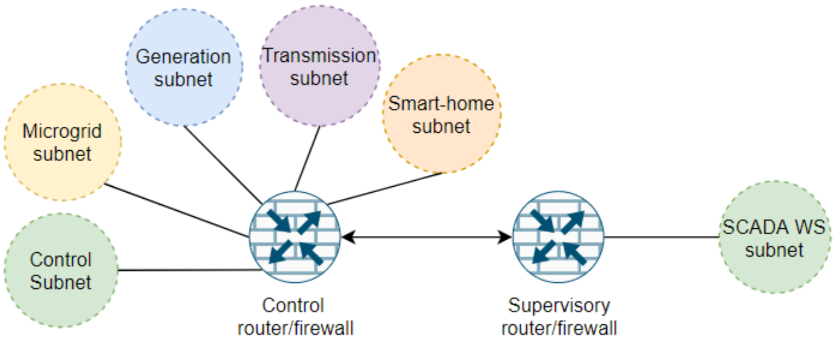


Figure 30 – High-level network infrastructure (routers and subnets) added to testbed.

The previously discovered assets are grouped into six subnets with 24-bit netmasks according to EPIC's process stages and architecture. The *Control* router is connected to EPIC's four process stages and the control network, while the *Supervisory* router is connected to the supervisory network where the SCADA WS connects. Additionally, EPIC's original SWs IP addresses are reused as routers' network interfaces IP addresses, which will decrease the number of total assets from 35 assets to 31 assets in total, where router assets have multiple IP addresses associated with it.

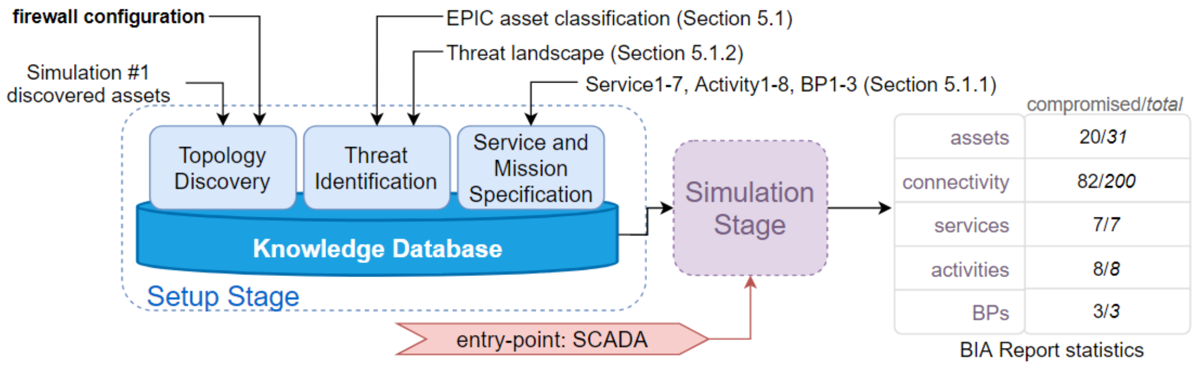
Even without firewall rules in place, this component automatically infers connectivity between assets on the same subnet upon the assumption they communicate freely (any protocol, using any network ports). Additionally, rules can be added to control connectivity between subnets. To study how this feature influences BIA's MIA, a set of 20 $deny$ and $allow$ rules, to block and allow communications respectively, is introduced for both firewalls.

In this way, in addition to the direct connectivity coming from the SCADA to all other assets, the resulting cross connectivity among other assets will be based on:

- Each stage's PLCs, IEDs and APs communicate with each other;
- CPLC communicates with all other PLCs;
- control assets (CPLC, Historian, CAP1 and CAP2) communicate with each other;
- main AP (CAP1) communicates with all other APs.

The Connectivity Discovery component first uses the *Comparing Algorithm* to determine the communications that are allowed by each firewall configuration and then the *Filtering Algorithm* to classify each rule according to its source and destination and apply them according to the firewall hierarchy in place. From the 20 rules given as input, 200 $connection$ entries were inferred: 190 from assets on the same subnet connecting freely with each other, and 10 from assets on different subnets that successfully represents the connectivity panorama described above.

With firewall policy in place and new connectivity inferred, let us simulate mission impact from the direct connectivity originated on SCADA to the CPLC and Historian, to understand how these control assets, if successfully compromised, can propagate the impact throughout the organization's infrastructure. The simulation's setup and result statistics are Figure 31.



	compromised/*total*
assets	20/*31*
connectivity	82/*200*
services	7/*7*
activities	8/*8*
BPs	3/*3*

BIA Report statistics

Figure 31 - Simulation #3 - MIA with firewall policy.

At first instance, it is possible to assess all BPs (and the activities and services supporting them) can be impacted. Even though the mission is equally impacted as in Simulation #1, a further analysis of how the impact propagates, illustrated in Figure 32, shows a key difference: Simulation #3 shows new propagations paths to impact 20 assets. This happens because in Simulation #1 there was no log of cross communications between other assets, which Simulation #3 was able to infer from firewall policy. Additionally, it shows the wireless network can also be leveraged to propagate the impact to the mission.

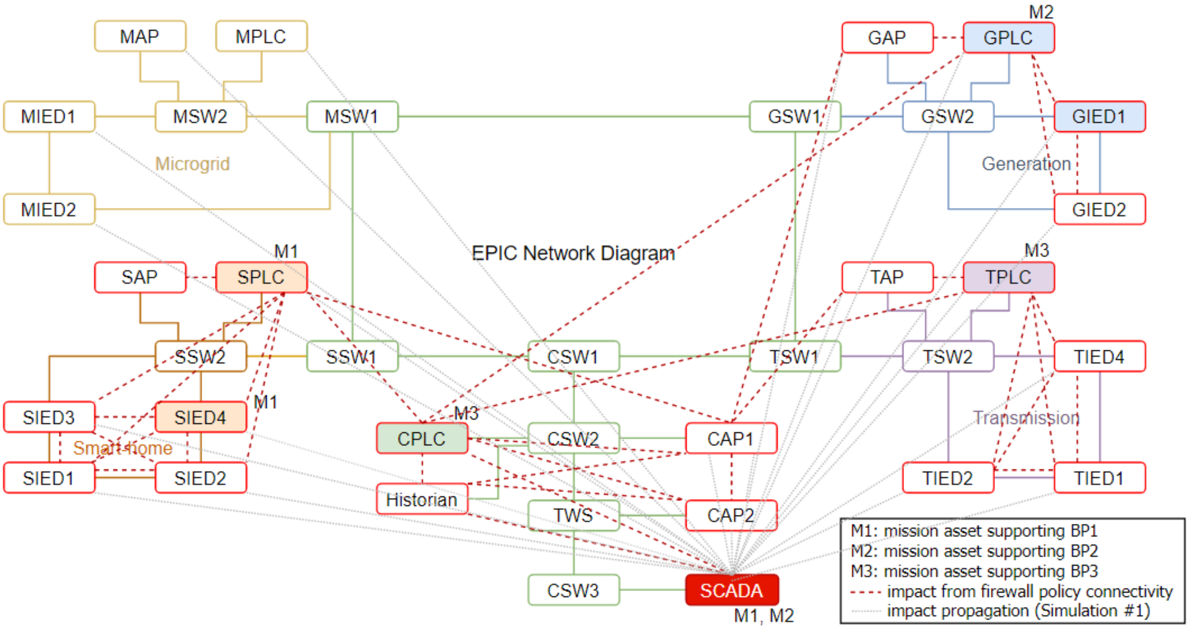


Figure 32 - Simulation #3 impact propagation.

These results show how BIA's Connectivity Discovery can be used to better reflect asset connectivity from firewall policy and how connectivity missed by Network Discovery can be leveraged to impact the mission, as well it shows how it can give the user a better control on asset connectivity being considered for BIA's simulation, since, using exclusively the Network Discovery component every $connection$ must be defined explicitly, whereas in Connectivity Discovery $connection$s entries are automatically inferred by rules. Furthermore, while an attacker is required to execute only a single attack path that leads to his objective, the defender is required to secure all possible paths. Therefore, recognizing available attack paths is especially relevant for MIA.

5.2.3. MIA case-studies

During the organization's lifetime is expected there will be changes to its modelled entities. For example, an existing threat can be mitigated, or a BP removed from the organization's mission. To evaluate and demonstrate how these dynamic changes are addressed by BIA to perform MIA, three case-studies were developed to apply BIA to realistic scenarios. The case-studies were designed to create different settings for simulation, according to changes on the initial setup of the mission layer, threat landscape and entry-point. As such, the input given to BIA's Topology Discovery knowledge unit will be static for the next experiments.

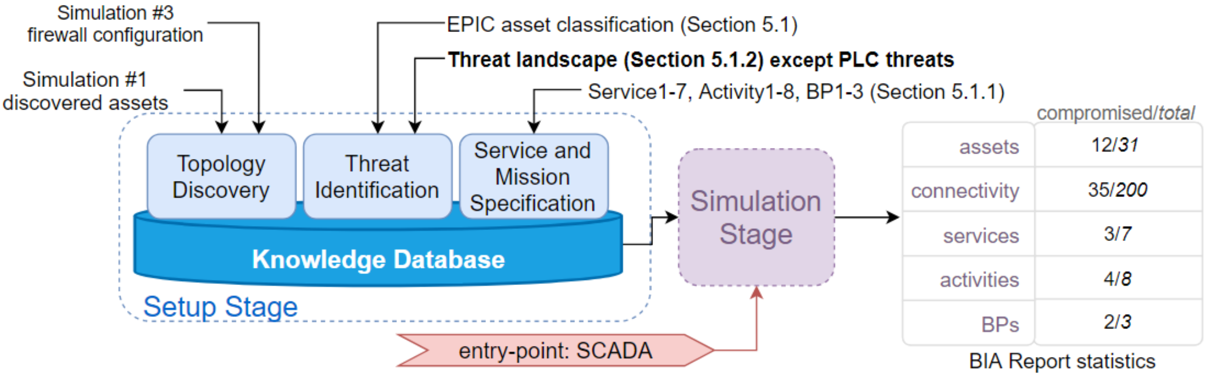
For demonstration purposes, asset topology discovered in Simulation #1 will be maintained but EPIC's asset connectivity extracted from packet capture will be omitted in favour of asset connectivity inferred by the firewall policy in Simulation #3, comprised of less $connection$ entries and a more diverse propagation scenario, to provide better illustration of mission impact. Therefore, asset connectivity panorama is based on SCADA only communicating with CPLC and Historian, at control network, CPLC connecting with all PLCs, CAP1 with other APs and all assets in the same subnet are allowed to communicate freely (as previously described in Section 5.2.2).

5.2.3.1. Threat Landscape

Being a key aspect for threat propagation (a precondition for the impact to propagate to another asset) it is expected that, changing the threat landscape affecting the infrastructure should produce a significant change on how the mission can be impacted. This section aims to study how changes in the threat landscape given to BIA's Topology Discovery knowledge unit, affects mission impact, specifically from a threat mitigation.

Threat mitigation techniques are employed to correct or reduce the impact of detected threats. Hence, let us evaluate how Simulation #3's results changes when PLC threats are mitigated (for instance, by updating the software running on the PLCs to the latest version). To achieve this, a new simulation, Simulation #4, is proposed where the initial threat landscape setup is changed to exclude any threat associated with PLCs.

From a first instance, since firewall policy in place determines CPLC as the connecting point between SCADA WS and the stages' PLCs, one may expect the mission to be much less susceptible to being impacted when threats affecting PLCs are mitigated, however, Simulation #4 shows how the organization can still be impacted (Figure 33 on the right).


Simulation #3
firewall configuration
Simulation #1
discovered assets
EPIC asset classification (Section 5.1)
Threat landscape (Section 5.1.2) except PLC threats
Service1-7, Activity1-8, BP1-3 (Section 5.1.1)
Topology Discovery
Threat Identification
Service and Mission Specification
Knowledge Database
Setup Stage
Simulation Stage
entry-point: SCADA

	compromised/*total*
assets	12/*31*
connectivity	35/*200*
services	3/*7*
activities	4/*8*
BPs	2/*3*

BIA Report statistics

Figure 33 - Simulation #4 – MIA for threat mitigation.

Comparing Simulation #4 to Simulation #3 (Figure 34), shows the infrastructure and mission is overall less impacted, as expected since the threat landscape was reduced, nonetheless, the simulation also results in 2 impacted BPs.

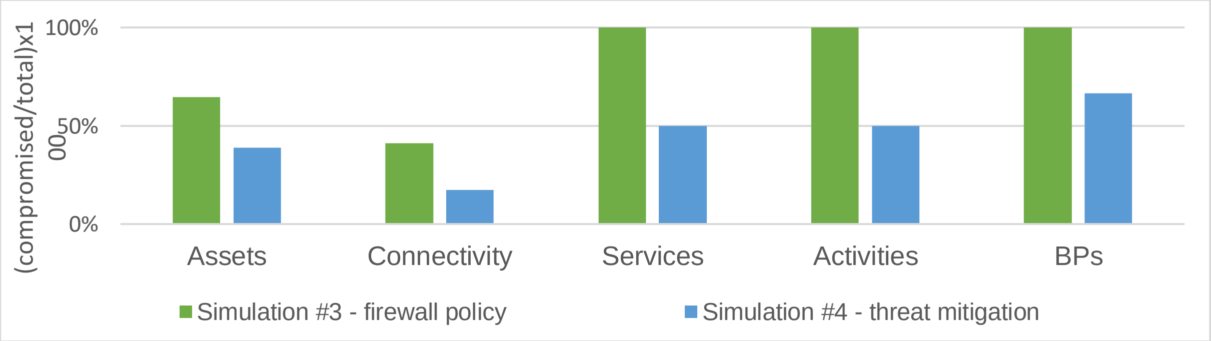


Figure 34 - Comparison chart between results of Simulation#3 and Simulation #4.

Furthermore, the report also shows how this happens (Figure 35): even though CPLC is not impacted, the entry-point is able to interact with the Historian, and consequently can exploit its *tampering* threat and compromise the integrity of the information it stores.

Since the Historian is able to communicate freely with any asset on the same subnet, the impact propagates to control APs, CAP1 and CAP2. From there, it is possible to take advantage of the wireless network through the APs in physical stages of EPIC, that in turn are able to communicate freely with all other assets in their respective subnets including the threatened IEDs supporting activities for 2 BPs (GIEDs and TIEDs).

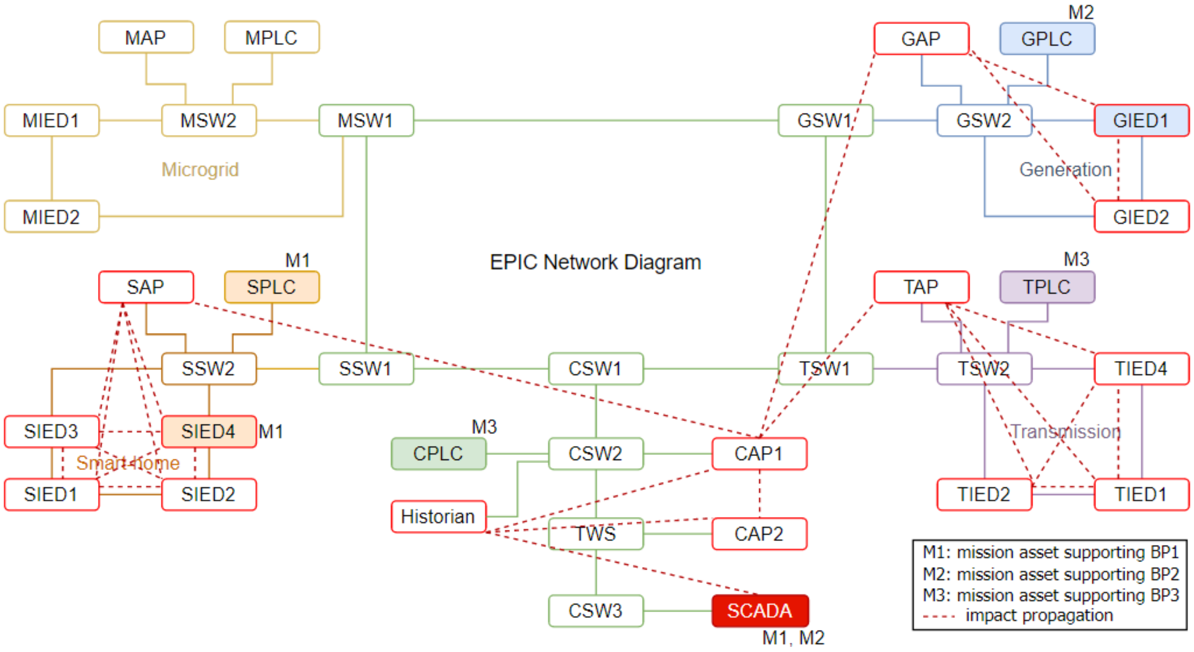


Figure 35 - Simulation #4 impact propagation.

5.2.3.2. Mission Layer

The mission layer is also expected to undergo several changes, when new BPs are added, or existing ones are removed or updated to a new version, with different or additional activities and/or services. Indeed, one high-concern attack scenario [98] is the infection of the ICS/SCADA systems during maintenance and upgrade processes, either by malware transmitted via the technicians' laptop, or via an infected firmware or update package.

In lieu of this, a key information for choosing appropriate prevention measures is to understand how an update to a BP contributes to overall mission impact before the update deployment. On that account, let us use BIA to simulate the impact of a BP update. Considering BP3 was not impacted in the last simulation when threats on PLCs were mitigated (review Figure 35 above), a process update is proposed for it.

BP3 goal is to obtain voltage values from TPLC (as described in Section 5.1.1 of the Evaluation Setup). In tandem with that goal, three new activities will be added to show those voltage values to the operator by returning the values to SCADA WS, as illustrated by Figure 36.

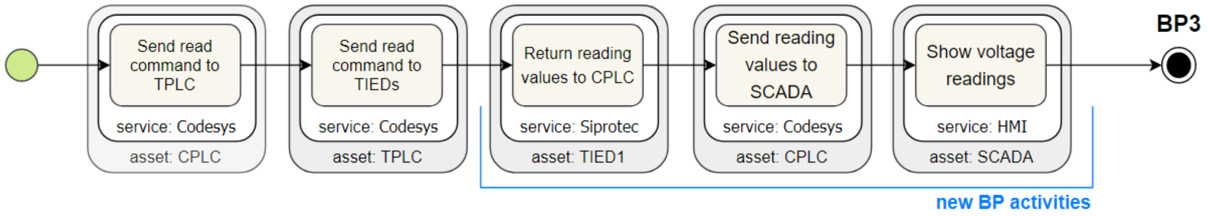
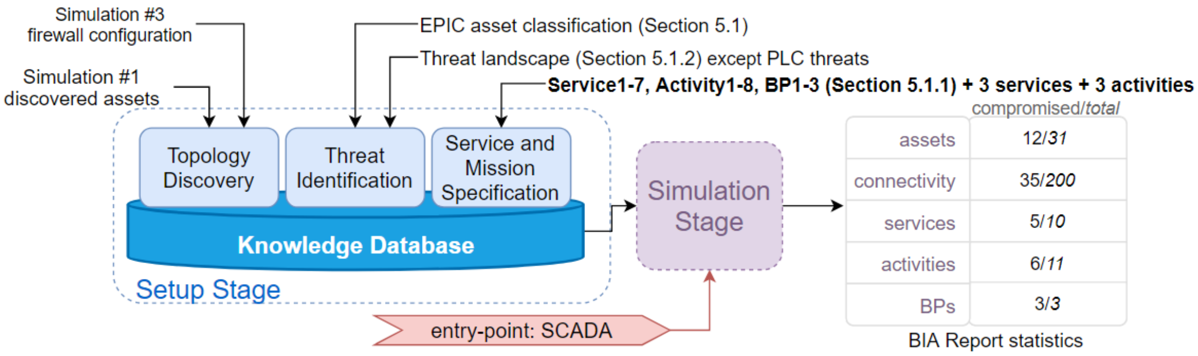


Figure 36 - Updated BP3 with 3 new activities and 3 new services.

This update to BP3 is represented by a new mission and service layer to be given to BIA's third knowledge unit, the Service and Mission Specification component. Combined with the simulation setup used for the previous experiment, Simulation #5 is presented next in Figure 37.


Simulation #3
firewall configuration
Simulation #1
discovered assets
EPIC asset classification (Section 5.1)
Threat landscape (Section 5.1.2) except PLC threats
Service1-7, Activity1-8, BP1-3 (Section 5.1.1) + 3 services + 3 activities
Topology Discovery
Threat Identification
Service and Mission Specification
Knowledge Database
Setup Stage
Simulation Stage
entry-point: SCADA

	compromised/*total*
assets	12/*31*
connectivity	35/*200*
services	5/*10*
activities	6/*11*
BPs	3/*3*

BIA Report statistics

Figure 37 - Simulation #5 - MIA for mission update.

Even though the impact propagates alike to the previous simulation propagation, Simulation #5 impact propagation, illustrated in Figure 38, shows how an updated version of BP3 can now be impacted: in the older version, on Simulation #4, the activities were exclusively based on PLCs which threats were mitigated, and now BP3's new activities are based on other types of assets as well, namely the SCADA and TIED1 which still have threats associated and are reachable from the entry-point.

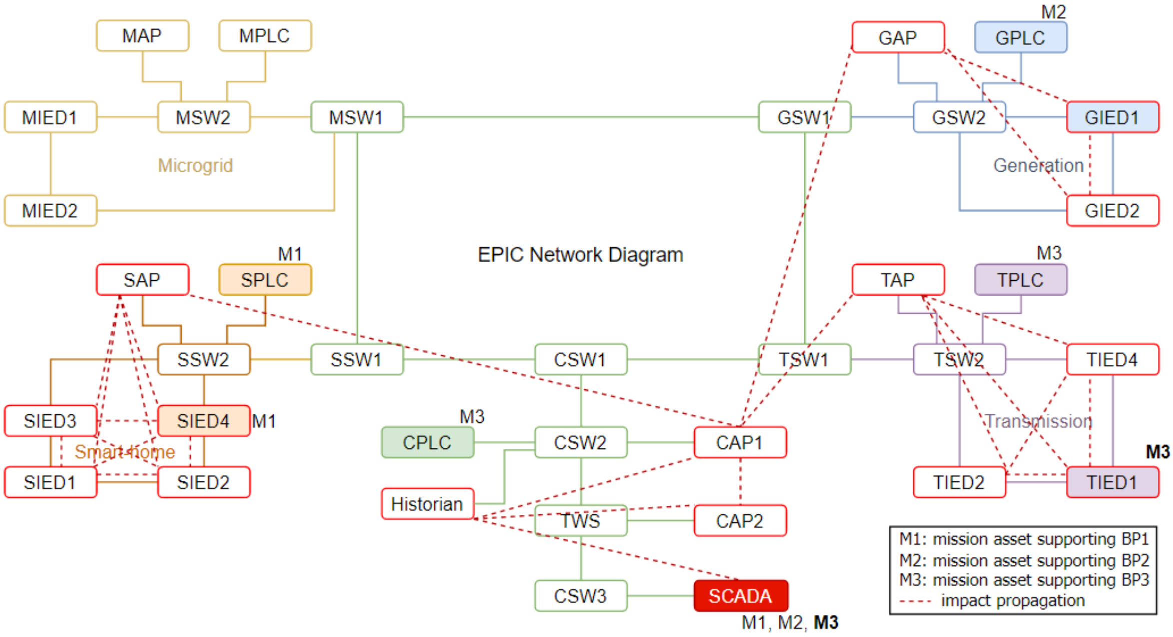


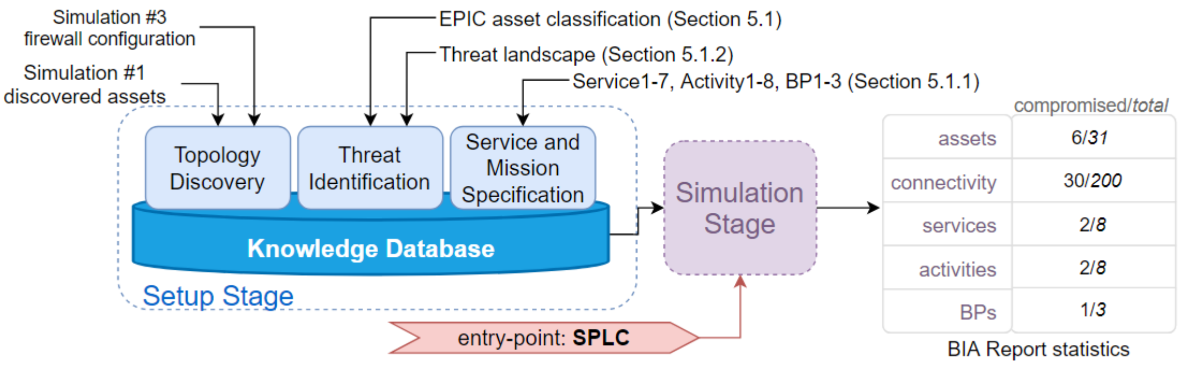
Figure 38 - Simulation #5 impact propagation.

This simulation shows how BIA assesses mission impact when changes are implemented to the mission, and how the final report can help to evaluate the updated impact to better understand susceptibilities a BP update can introduce in the overall mission security.

5.2.3.3. Entry-point

By re-running BIA's Setup stage after changes to the system, the previous simulations have analysed how BIA is capable of simulating and assessing the impact to the testbed's business-processes from an exploited threat in the SCADA workstation, according to different profiles of the organization, regarding its topology, connectivity, firewall policy, threat landscape and mission specification.

Nonetheless, based on a single organizational setup, different attacks can be designed and launched from different entry-points to impact the mission. BIA two-stage solution addresses this case allowing the user to rerun the Simulation stage for different entry-points independently of the Setup stage. Among others, a feasible attack scenario to launch in EPIC is a *nuisance tripping* attack [96] where a malware attack on the firmware of PLCs can result in a unwarranted tripping by triggering the protection functions in IEDs. Hence, to assess the impact a compromised PLC can cause on the mission, let us simulate again the initial threat landscape and mission layer, and assess the impact from the SPLC being compromised by a malware threat, with the simulation setup illustrated by Figure 39.



	compromised/*total*
assets	6/*31*
connectivity	30/*200*
services	2/*8*
activities	2/*8*
BPs	1/*3*

Figure 39 - Simulation #6 - MIA for SPLC as entry-point.

The resulting impact propagation is depicted in Figure 40 and shows how an exploited threat in the SPLC can indeed impact the stage's IEDs and detect a nuisance tripping attack on that stage. The impact is contained in the Smart-home stage because the considered firewall policy blocks the PLCs from communicating back with the CPLC, which, otherwise, would allow the impact to propagate to other stages. Even though the SPLC does not belong to a supervisory network as SCADA does, BIA shows how it still has the potential of compromising the mission, in this case, BIA reports 6 compromised assets in EPIC's Smart-Home stage where 2 of them are mission assets supporting 2 activities from BP1.

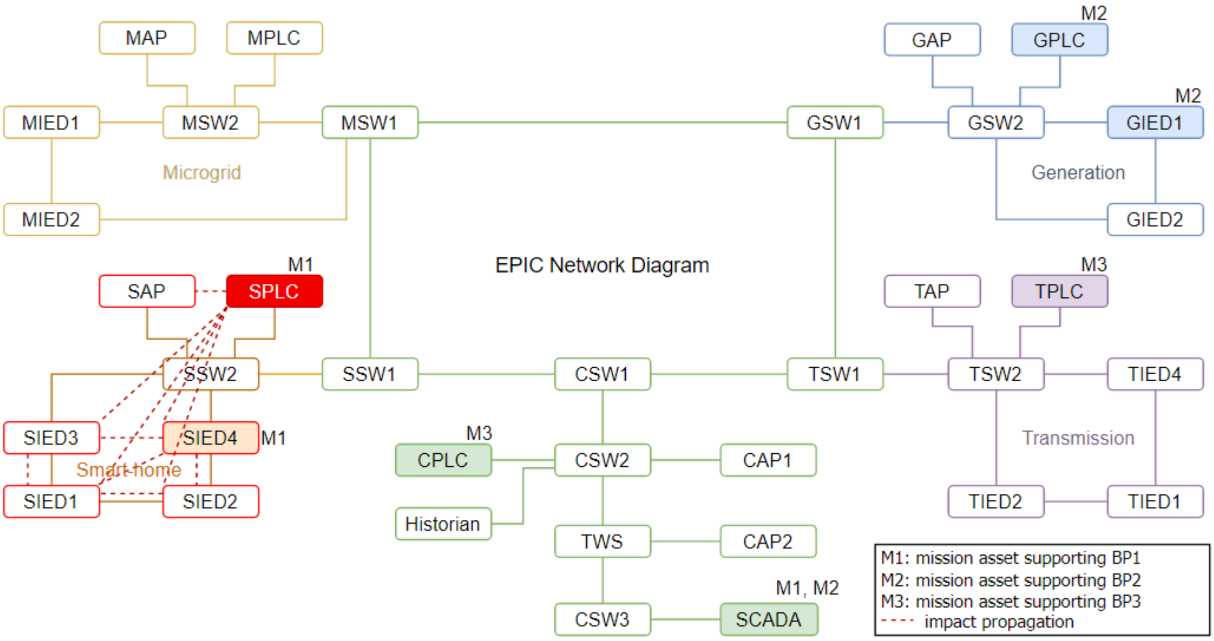


Figure 40 – Simulation #6 impact propagation.

5.3. Summary

This chapter has presented the evaluation conducted on the proposed BIA solution for MIA.

The first section described the evaluation's setup designed to outline BIA's requirements for minimal functionality, regarding the mission layer, the threat landscape, and the entry-point to be considered for the simulations that came after.

The second section proceeded to define the evaluation process to be undertaken. Conclusions were taken based on the content analysis conducted during the study of the simulations' results and limitations of the solution were also identified and explained. The evaluation process can be summarized as follows:

1. *Topology Discovery.* BIA was evaluated according to the discovery accuracy of BIA's Topology Discovery component. A series of experiments found it was successful in finding $\approx 97\%$ of documented assets along with undocumented assets. Additionally, BIA's discovered asset connectivity showed how the extent of asset connectivity can become a limitation for BIA's threat propagation methodology, based on the number of $connection$s and propagation cycles. This limitation was explained, and recommendations were provided and implemented with a new setting for simulation.
2. *Connectivity Discovery.* This case studied how BIA's Connectivity Discovery component can be used to infer missed connectivity by the previous component. The results showed how it can improve MIA with asset connectivity inferred by firewall policies, with additional propagation paths found to impact the discovered assets.
3. *MIA case-studies.* BIA's inputs were altered according to realistic adjustments that are expected to happen to the organization's profile during the organization's lifetime. To evaluate how the threat landscape can change MIA results, BIA was applied for a threat mitigation case-study. The results

showed how the mission can still be impacted, but also how it decreases. A second case-study was developed to evaluated how BIA may be leveraged to perform MIA for different settings of mission layer. A simulation was performed to evaluate how an update to a BP can affect mission impact. The results showed how a previous deemed secure BP can become impacted by an update. Finally, to evaluate how differently an entry-point can impact the mission, BIA was used to simulate how a nuisance tripping attack to a PLC, with the intent of opening circuit breakers, can impact the mission. BIA's results showed how all IEDs are indeed impacted and how the mission they support could be compromised.

In total, 6 simulations were put forward. Report content of each simulation was examined and used to describe the key aspects of the MIA result. This evaluation process has shown BIA's capability of performing MIA for a panoply of different organizational settings producing relevant mission impact information to address organization's security realistic dynamics.

6. Conclusion

This dissertation has presented a novel approach for MIA. To do so, a comprehensive survey of the state-of-the-art on the subject was performed to explore relevant approaches, that has led to the identification of three main stages of MIA: impact modelling, impact propagation and impact measurement. It was found that several works resort to an entity dependency graph with a multi-layered structure to model the organization with various abstraction layers, however it was also accounted that only a few works provide explicit sources for populating the proposed models, which supports the motivation that the required knowledge for profiling an organization is often difficult to obtain. In lieu of this, possible data sources for different assessment layers were also studied.

Next, a review of propagation methodologies drove to the classification of three types of model-based propagation approaches: logic-based, probabilistic-based and sensitivity-based. A further analysis of their features has concluded both probabilistic-based and sensitivity-based propagation methodologies require a high modelling overhead, which drove to the decision of employing a logic-based propagation method based on attack graphs by the present work. Finally, an overview of the impact metrics proposed by the studied MIA solutions was presented.

As a result, BIA was proposed to focus on the modelling and propagation aspects of MIA and to address (1) the insufficient information about mission impact of cyber-threats affecting the organization by including a cyber-threat layer in its evaluation model; (2) the lack of information about how firewall policies can further refine the asset layer by inferring asset connectivity from firewall configuration, and (3) the recurrent missing application by conceptual approaches, by offering a employable assessment model and impact propagation platform in a single tool that is easy to interact.

BIA was implemented with the aim of incorporating and combine existing studies, standards and tools into a tool that takes in disparate, but accessible, information about the organization's profile and produces a MIA report accordingly. To prove BIA's effectiveness in accomplishing its goal, several case-studies were developed upon an ICS dataset according to realistic dynamics that are expected to happen during an organization's lifetime. Applying BIA to these case-studies have shown that it can generate a relevant report on mission impact for a great number of different settings, to assess the organization's risk situation, which led to the conclusion that the main purpose of this dissertation was achieved. The following sections highlights the main achievements of this work and discusses possible directions for future work to address both this work's limitations and arisen opportunities.

6.1. Achievements

In reaching its goal, this dissertation accomplished two important achievements. The first achievement is the construction of a four-layer evaluation model for MIA, that offers a way to profile an organization and model the impact, which included a rarely considered threat layer that allows mapping cyber-threats onto the organization's assets. The second achievement is the concretization of a simulation platform that allows to simulate mission impact caused by an exploited cyber-threat. Moreover, from the development and application of BIA, some other contributions can be noted:

- BIA is capable of converting disparate information about assets, threats, firewall policy and business-processes into an impact assessment report.
- Its implementation incorporated existing and established tools, such as *Tshark* network analyser, MulVAL attack graph with a new knowledge base, *Neo4j* graph database and *BP-IDS* intrusion detection system, as well as known standards, as *IPTABLES* for the organization's asset connectivity and firewall policy, and *STRIDE* to classify threats. Additionally, previous work on inspecting firewall policies motivated BIA's algorithm to infer asset connectivity allowed by firewall policy and hierarchies.
- BIA was built in a way that is independent from the organization's domain (military, business or ICS), however, its application was done on an ICS, where not only a cyber network can become a target, but the physical network can also be impacted, which reinforces the need for MIA.

6.2. Future Work

From its accomplished achievements it can be concluded that BIA serves as consolidate baseline tool for MIA, and, as such, numerous options exist towards further development and improvement of the approach. In regard to BIA's assumptions and limitations, the main contributions would be based on:

- *Solving propagation cycles.* BIA's most important challenge to address would be the propagation cycles generated by bidirectional asset connectivity. This can be approached either *before* the simulation takes place, by automatically refining asset connectivity considered, *during* the simulation with additional Horn Clauses, or upon the attack graph generated by MulVAL by transforming the resulting directed cyclic graph to a tree of attack paths.
- *Integrating new horizontal dependencies*. Other great contribution to BIA is to integrate horizontal dependencies on other assessment layers to create a more authentic simulation.
- *Refining threat propagation heuristics*. The proposed heuristic for threat propagation was based on if an asset is accessible and has a threat associated with it then it can be compromised. This is not always the case, where other conditions must be present for a threat to be exploited, or even, if it is indeed exploited it does means it impacts the asset itself, but the impact not always propagate to others it can connect with. One possibility would be to map threats onto the services the assets run, and only propagate the impact to other assets if there is connectivity to the *port* the service runs on.

Moreover, during BIA's design and application some interesting opportunities arisen for future work to address as an extension:

- *Inclusion of impact metrics*. The most compelling contribution would be to integrate impact metrics in BIA's model and simulation platform. This could be done with qualitative and quantitative metrics at any assessment layer.
- *Inclusion of different asset types*. For instance, EPIC's physical assets, such as circuit breakers, loads, batteries, generators, transformers, etc…, can also be taken into account when performing MIA. With this in mind, BIA's database schema was effectively implemented to easily integrate other types of assets besides network assets.
- *Visualization of results*. An interesting view of BIA's report would be trough visualization, which is currently under development by other works.

7. References

[1] A. Kott, “Assessing Mission Impact of Cyberattacks Compiled,” in *Proceedings of the NATO IST-128 Workshop*, 2015, p. 100.

[2] B. E. Strom *et al.*, “Finding Cyber Threats with ATT&CK^{TM}-Based Analytics,” 2017.

[3] C. Cao, L. P. Yuan, A. Singhal, P. Liu, X. Sun, and S. Zhu, “Assessing attack impact on business processes by interconnecting attack graphs and entity dependency graphs,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 10980 LNCS, pp. 330–348.

[4] R. Langer, “Stuxnet: Dissecting a cyberwarfare weapon,” 2011.

[5] A. Cherepanov and R. Lipovsky, “Industroyer: Biggest threat to industrial control systems since Stuxnet,” 2017.

[6] X. He, S. Rass, and H. Meer, “Threat Assessment for Multistage Cyber Attacks in Smart Grid Communication Networks,” Fakultät für Informatik und Mathematik Universität Passau, Germany, 2017.

[7] H. Debar and A. Wespi, “Aggregation and Correlation of Intrusion-Detection Alerts,” pp. 85–103, 2007.

[8] S. Jajodia, S. Noel, P. Kalapa, M. Albanese, and J. Williams, “Cauldron: Mission-centric cyber situational awareness with defense in depth,” *Proc. - IEEE Mil. Commun. Conf. MILCOM*, no. May 2017, pp. 1339–1344, 2011.

[9] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, “Computer Security Incident Handling Guide Recommendations of the National Institute of Standards and Technology.”

[10] Imperva, “Cybersecurity and Healthcare A Survey of Healthcare IT Professionals at HIMSS.”

[11] R. Gula, “Correlating IDS Alerts with Vulnerability Information,” *Tenable Netw. Secur. http//www. tenablesecurity.*, no. January, p. 10, 2002.

[12] J. R. Goodall, A. D’Amico, and J. K. Kopylec, “Camus: Automatically mapping cyber assets to missions and users,” *Proc. - IEEE Mil. Commun. Conf. MILCOM*, pp. 1–7, 2009.

[13] H. Bahşi, C. J. Udokwu, U. Tatar, and A. Norta, “Impact Assessment of Cyber Actions on Missions or Business Processes: A Systematic Literature Review,” *Int. Conf. Cyber Warf. Secur.*, no. March, pp. 11–20, 2018.

[14] A. Kott, J. Ludwig, and M. Lange, “Assessing Mission Impact of Cyberattacks: Toward a Model-Driven Paradigm,” *IEEE Secur. Priv.*, vol. 15, no. 5, pp. 65–74, 2017.

[15] B. J. Argauer and S. J. Yang, “VTAC: virtual terrain assisted impact assessment for cyber attacks,” 2008.

[16] S. Noel *et al.*, “Analyzing Mission Impacts of Cyber Actions (AMICA),” *NATO IST128 Work. Cyber Attack Detect.*, no. 15, pp. 1–16, 2015.

[17] C. Liu, A. Singhal, and D. Wijesekera, “A layered graphical model for mission attack impact analysis,” in *2017 IEEE Conference on Communications and Network Security (CNS)*, 2017, pp. 602–609.

[18] A. Negotiation, “Cyber-Argus: Modeling C2 Impacts of Cyber Attacks,” *19th ICCRTS*, pp. 1–17, 2014.

[19] L. Gilbert, C. Henney, L. Alford, A. Khalili, and B. Michalk, “Impact modeling and prediction of attacks on

cyber targets," *Cyber Secur. Situat. Manag. Impact Assess. II; Vis. Anal. Homel. Def. Secur. II*, vol. 7709, p. 77090M, 2010.

[20] A. Motzek, R. Möller, M. Lange, and S. Dubus, "Probabilistic mission impact assessment based on widespread local events," *Assess. Mission Impact Cyberattacks*, p. 1, 2015.

[21] A. Barreto, P. Costa, and E. Yano, "A Semantic Approach to Evaluate the Impact of Cyber Actions to the Physical Domain," *Semant. Technol. Intell. Defense, Secur. 2012*, vol. 966, pp. 64–71, 2012.

[22] G. Jakobson, "Extending Situation Modeling with Inference of Plausible Future Cyber Situations," in *2011 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), Miami Beach, FL*, 2011.

[23] Y. Sun, T. Wu, X. Liu, and M. S. Obaidat, "Multilayered Impact Evaluation Model for Attacking Missions," *IEEE Syst. J.*, vol. 10, no. 4, pp. 1304–1315, Dec. 2016.

[24] B. Genge, I. Kiss, and P. Haller, "A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures," *Int. J. Crit. Infrastruct. Prot.*, vol. 10, pp. 3–17, 2015.

[25] H. Orojloo and M. A. Azgomi, "A method for evaluating the consequence propagation of security attacks in cyber–physical systems," *Futur. Gener. Comput. Syst.*, vol. 67, pp. 57–71, 2017.

[26] S. Musman, A. Temin, M. Tanner, D. Fox, and B. Pridemore, "Evaluating the impact of cyber attacks on missions," *5th Eur. Conf. Inf. Manag. Eval. ECIME 2011*, pp. 446–456, 2011.

[27] S. Musman and A. Temin, "A Cyber Mission Impact Assessment Tool," in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2015, pp. 1–7.

[28] S. Noel, E. Harley, K. H. Tam, M. Limiero, and M. Share, *CyGraph: Graph-Based Analytics and Visualization for Cybersecurity*, 1st ed., vol. 35, no. January 2016. Elsevier B.V., 2016.

[29] G. Jakobson, "Mission cyber security situation assessment using impact dependency graphs," *14th Int. Conf. Inf. Fusion*, pp. 1–8, 2011.

[30] I. Kotenko and A. Chechulin, "A Cyber Attack Modeling and Impact Assessment Framework," in *2013 5th International Conference on Cyber Conflict (CyCon)*, 2013, pp. 1–24.

[31] P. A. Porras, M. W. Fong, and A. Valdes, "A mission-impact-based approach to INFOSEC alarm correlation," *Int. Work. Recent Adv. Intrusion Detect.*, vol. 2516, pp. 95–114, 2002.

[32] X. Sun, A. Singhal, and P. Liu, "Who Touched My Mission : Towards Probabilistic Mission Impact Assessment," *SafeConfig '15 Proc. 2015 Work. Autom. Decis. Mak. Act. Cyber Def.*, pp. 21–26, 2015.

[33] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren, "Computing the impact of cyber attacks on complex missions," in *2011 IEEE International Systems Conference*, 2011, pp. 46–51.

[34] M. Lange, M. Krotofil, and R. Möller, "Mission Impact Assessment in Power Grids," in *Proceedings of the NATO IST-128 Workshop: Assessing Mission Impact of Cyberattacks*, pp. 51–59.

[35] R. Sawilla and X. Ou, "Identifying critical attack assets in dependency attack graphs," 2008.

[36] N. Kheir *et al.*, "Assessing the risk of complex ICT systems," *Ann. Telecommun.*, vol. 73, pp. 95–109, 2018.

[37] M. R. Grimaila and L. W. Fortson, "Towards an Information Asset-Based Defensive Cyber Damage Assessment Process," *Proc. 2007 IEEE Symp. Comput. Intell. Secur. Def. Appl. (CISDA 2007)*, no. Cisda, pp. 206–212, 2007.

[38] I. Kotenko and E. Doynikova, “Evaluation of Computer Network Security based on Attack Graphs and Security Event Processing.”

[39] J. Holsopple and S. J. Yang, “FuSIA: Future situation and impact awareness,” *Proc. 11th Int. Conf. Inf. Fusion, FUSION 2008*, 2008.

[40] A. Kim, M. Kang, J. Luo, and A. Velasquez, “A Framework for Event Prioritization in Cyber Network Defense,” 2014.

[41] International Standard Organization, “ISO / IEC 27032,” 2011.

[42] I. Standard, “ISO/IEC 27005,” 2008.

[43] W. M. P. Van Der Aalst, B. Benatallah, F. Casati, F. Curbera, and E. Verbeek, “Business Process Management: Where Business Processes and Web Services Meet.”

[44] D. Menijvar, J. Recker, and M. Weske, “Management and Engineering of Process Aware Information Systems: Introduction to the Special Issue ,” 2011.

[45] M. Malinova, “A Language for Designing Process Maps,” 2016.

[46] “BPMN Specification - Business Process Model and Notation,” 2010. [Online]. Available: https://www.omg.org/spec/BPMN/2.0.2/PDF. [Accessed: 20-Jan-2020].

[47] O. WS-BPEL Technical Committee, “Web Services Business Process Execution Language Version 2.0,” 2007.

[48] S. A. White, “Using BPMN to Model a BPEL Process,” 2005.

[49] INOV, “BPIDS: Using business process specification to leverage intrusion detection in public transportation.”

[50] W. Van Der Aalst, T. Weijters, and L. Maruster, “Workflow Mining: Discovering Process Models from Event Logs.”

[51] A. J. M. . Weijters and W. M. . Van der Aalst, “Process Mining Discovering Workflow Models from Event Based Data,” in *Proceddings of the 13th Dutch-Belgian Artficial Intelligence Conference*, 2001, pp. 295–302.

[52] A. J. M. M. Weijters, W. M. P. Van Der Aalst, and A. K. Alves De Medeiros, “Process Mining with the HeuristicsMiner Algorithm.”

[53] S. J. J. Leemans, D. Fahland, and W. M. P. Van Der Aalst, “Discovering Block-Structured Process Models From Event Logs-A Constructive Approach.”

[54] D. Marques, “Business Process Security Specification Automatic Extraction,” 2019.

[55] F. Apolinário, “ICS Cyber-Mission Discovery System,” 2020.

[56] G. Greco, A. Guzzo, L. Pontieri, and D. Saccà, “Discovering Expressive Process Models by Clustering Log Traces.”

[57] C. Mavrakis, “Passive Asset Discovery and Operating System Fingerprinting in Industrial Control System Networks.”

[58] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, “Traffic Classification through Simple Statistical Fingerprinting.”

[59] X. Chen, M. Zhang, Z. Morley Mao, and P. Bahl, “Automating Network Application Dependency Discovery:

Experiences, Limitations, and New Solutions."

[60] X. Xiong, X. Jia, and P. Liu, "SHELF: Preserving business continuity and availability in an intrusion recovery system," *Proc. - Annu. Comput. Secur. Appl. Conf. ACSAC*, pp. 484–493, 2009.

[61] L. Popa, B.-G. Chun, J. Chandrashekar, and N. Taft, "Macroscope: End-Point Approach to Networked Application Dependency Discovery," 2009.

[62] S. Jajodia, S. Noel, and B. O'berry, "Topological Analysis of Network Attack Vulnerability."

[63] S. Noel, E. Harley, K. H. Tam, and G. Gyor, "Big-Data Architecture for Cyber Attack Graphs: Representing Security Relationships in NoSQL Graph Databases," *IEEE Symp. Technol. Homel. Secur.*, no. 14–3549, pp. 1–6, 2015.

[64] E. S. Al-Shaer and H. H. Hamed, "Design and Implementation of Firewall Policy Advisor Tools," 2002.

[65] M. Abedin, S. Nessa, L. Khan, and B. Thuraisingham, "Detection and Resolution of Anomalies in Firewall Policy Rules," in *IFIP Annual Conference on Data and Applications Security and Privacy*, 2006, no. Data and Applications Security XX, pp. 15–29.

[66] E. S. Al-Shaer and H. H. Hamed, "Discovery of policy anomalies in distributed firewalls," in *IEEE INFOCOM 2004*, 2004, vol. 4, pp. 2605–2616.

[67] E. W. Fulp, "Optimization of Network Firewall Policies using Directed Acyclical Graphs," in *Proc. of IEEE Internet Management Conference*, 2005.

[68] A. Tongaonkra, N. Inamdar, and R. Sekar, "Inferring Higher Level Policies from Firewall Rules," *Proc. 21st Large Install. Syst. Adm. Conf.*, p. Pp. 17-26, 2007.

[69] L. Yuan, J. Mai, Z. Su, H. Chen, C.-N. Chuah, and P. Mohapatra, "FIREMAN: A Toolkit for FIREwall Modeling and ANalysis," in *2006 IEEE Symposium on Security and Privacy (S&P'06)*, 2006.

[70] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan, "Conflict Classification and Analysis of Distributed Firewall Policies," *IEEE J. Sel. AREAS Commun.*, vol. 23, no. 10, 2005.

[71] A. D. ' Amico, S. Salas, and S. Decisions, "Visualization as an Aid for Assessing the Mission Impact of Information Security Breaches," 2003.

[72] "CVE - Common Vulnerabilities and Exposures (CVE)." [Online]. Available: https://cve.mitre.org/. [Accessed: 03-Oct-2019].

[73] M. Schiffman, "The Common Vulnerability Reporting Framework Dictionary of Elements."

[74] S. Barnum, "Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX^{TM})," *MITRE Corp. July*, pp. 1–20, 2014.

[75] "Characterizing Malware with MAEC and STIX," 2014.

[76] "NVD." [Online]. Available: https://nvd.nist.gov/. [Accessed: 01-Oct-2019].

[77] "CERT/CC Vulnerability Notes Database." [Online]. Available: https://www.kb.cert.org/vuls/.

[78] "SecurityFocus, Bugtraq." [Online]. Available: https://www.securityfocus.com/vulnerabilities. [Accessed: 03-Oct-2019].

[79] "Symantec Corporation, Symantec DeepSight^{TM}." [Online]. Available: https://www.symantec.com/. [Accessed: 03-Oct-2019].

[80] M. Peter, K. Scarfone, and S. Romanosky, "Common Vulnerability Scoring System," *IEEE Secur. Priv.*, vol.

4, no. 6, 2006.

[81] M. Meier, N. Bischof, and T. Holz, “SHEDEL-A SIMPLE HIERARCHICAL EVENT DESCRIPTION LANGUAGE FOR SPECIFYING ATTACK SIGNATURES·,” 2002.

[82] “F5 BIG-IP ASM Attack Signatures.” [Online]. Available: https://worldtechit.com/f5-resources/f5-big-ip-asm-attack-signatures/. [Accessed: 18-May-2020].

[83] K. Ingols, R. Lippmann, and K. Piwowarski, “Practical Attack Graph Generation for Network Defense.”

[84] S. Noel and S. Jajodia, “Managing attack graph complexity through visual hierarchical aggregation,” 2004.

[85] P. Ammann, D. Wijesekera, and S. Kaushik, “Scalable, Graph-Based Network Vulnerability Analysis,” 2002.

[86] X. (Simon) Ou, W. F. Boyer, and S. Zhang, “MulVAL: A logic-based enterprise network security analyzer,” *14th USENIX Secur. Symp.*, 2013.

[87] S. Yi *et al.*, “Overview on attack graph generation and visualization technology,” *Proc. Int. Conf. Anti-Counterfeiting, Secur. Identification, ASID*, pp. 1–6, 2013.

[88] S. Govindavajhala, “Status of the MulVAL project,” no. May, 2006.

[89] X. Ou, W. F. Boyer, and M. A. McQueen, “A Scalable Approach to Attack Graph Generation,” 2006.

[90] M. Froh and G. Henderson, “MulVAL extensions II Defence R&D Canada-Ottawa,” 2009.

[91] E. Bacic, M. Froh, and G. Henderson, “MulVAL Extensions For Dynamic Asset Protection Defence,” 2006.

[92] X. Sun, A. Singhal, and P. Liu, “Towards Actionable Mission Impact Assessment in the Context of Cloud computing,” 2017.

[93] M. Petkovic and V. Mihajlovic, “Dynamic Bayesian Networks: A State of the Art,” 2001.

[94] M. R. Grimaila, L. W. Fortson, and J. L. Sutton, “Design considerations for a cyber incident mission impact assessment (CIMIA) process,” 2009.

[95] P. Rao, K. Sagonas, D. S. Warren, J. Freire, T. Swift, and D. Warren, “XSB - A System for Efficiently Computing Well Founded Semantics,” 1997.

[96] S. Adepu, N. K. Kandasamy, and A. Mathur, “EPIC: An electric power testbed for research and training in cyber physical systems security,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11387 LNCS, no. November, pp. 37–52, 2019.

[97] A. Siddiqi, N. O. Tippenhauer, D. Mashima, and B. Chen, “On Practical Threat Scenario Testing in an Electric Power ICS Testbed,” vol. 7, 2018.

[98] ENISA, *Communication network dependencies for ICS/SCADA Systems*, no. December. 2016.

[99] D. Inc., “Crashoverride: Analysis of the Threat to Electric Grid Operations.”