Quantifying the impact propagation of cyber attacks using business logic modeling

Marwan Lazrag, Christophe Kiennert, Joaquin Garcia-alfaro

Quantifying the Impact Propagation of Cyber Attacks using Business Logic Modeling

Marwan Lazrag, Christophe Kiennert, Joaquin Garcia-Alfaro(✉)

SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau, France
{firstName.lastName}@telecom-sudparis.eu

Abstract. Cyber-attacks affect the security properties of critical systems, such as confidentiality, integrity, and availability of crucial business activities. They also affect mission quality and performance. Existing risk assessment tools handling the problem still present some limitations, owing to the difficulty of describing the enterprise infrastructure, such as identifying assets, missions, and their dependencies. Furthermore, little research has been conducted to assess the impact propagation of external events on business entities.

In this chapter, we survey existing methods aiming to solve the aforementioned limitations. We focus on two main families: financial and operational impact assessment. The latter aims to specifically assess the impact propagation of cyber-attacks. For instance, cyber-attacks targeting the infrastructure assets and perturbing the execution and performance of the company's activities. It can also include the evaluation of the financial impact based on former financial assessment methodologies.

We also present a concrete operational impact propagation assessment contribution. This contribution extends previous work by enhancing the definition associated to organizational activities that might be impacted by cyber-attacks. It relies on business impact analysis via business logic modeling. It also includes metrics to quantify (i) the impact propagation probability on the business entities, and (ii) critical time (i.e., the time during which the business entity is not be impacted).

Keywords: Cybersecurity · Risk Analysis · Impact Assessment · Cyber-Attack · Impact Propagation · Resource Dependency Graph · Mission Dependency Graph.

1 Introduction

Modeling the technical assets and missions of a company, as well as identifying the dependencies between them, can assist the security analyst in responding to the incident more effectively. With the discovery of new vulnerabilities and an increase in attacks aiming at compromising the confidentiality, integrity, and availability of business activities, as well as deteriorating mission quality and performance, assessing the impact of these external events may help the operator in determining the level of emergency and making decisions.

In addition to cyber-attacks, the limitations of existing cyber defense tools to protect missions and enterprise networks, some recent research has been conducted to quantify the impact of attacks based on attack graph tools and Common Vulnerability Scoring System (CVSS) scores. However, due to the difficulty of modeling the company's business functions and processes, little of this research has focused on assessing the operational impact of external events on business missions to protect critical infrastructure and complex enterprise architecture.

Previous work focuses on methodologies based on extending metrics such as CVSS scores, the Impact Factor (IF) to assess the attack impact [10]. For instance, Cao *et al.* [2] use CVSS scores over attack graphs, in order to compute an eventual business impact score. Operational impact assessment consists of estimating the impact of interrupting services and functionalities of missions, such as business functions and processes, due to an attack. For acquiring knowledge about the business activity of a company, we assume Business Process Model Notation (BPMN), seen as a common standard to derive the list of business functions and organizational processes.

Regarding the assessment of the propagation of impacts on business functions and processes, we evaluate the operational impact of an external event that has already occurred within the infrastructure, composed of technical assets and business entities. Our approach is not just aimed at studying the propagation of the attack within assets, but also assesses the operational impact at the level of the business entities.

We extend previous work [9,13–15] by including business logic into previous models. The added business logic model is a layered structure composed of assets and business entities, such as business functions and business processes. It consists of a graph-like structure, which links technical assets into business entities. The resulting model is used to assess the operational impact of shock events on missions, as well as to calculate the criticality of technical assets and estimate the downtime tolerance. Based on the business logic model, we implement a method for assessing the impact propagation of attacks on business entities, as well as to conduct a realistic case study on various business models to evaluate our method. To accomplish this, we define two metrics: (i) the probability of the impact on the business entities, and (ii) the critical time, which represents the time during which the business entity will not be impacted.

To identify the most critical assets and the most impacted missions, we assume graphical language for reasoning on functionalities such as: 1) calculation of the impact of events on the missions: impact probability and critical time, and 2) computation of the criticality of assets. The methodology is evaluated on realistic use cases and provides relevant results.

The focus of the contribution is on generating a business logic model, as well as demonstrating our method to assess the impact propagation of an external event into the nodes of this model. Then, using Monte-Carlo approximation, we can assess the scalability of the required computations needed by the system to evaluate the criticality of assets in some larger scenarios. We also focus on identifying the most critical assets in the infrastructure to determine which assets

contribute the most to the propagation of the impact by assigning a criticality value to each asset. Our approach is based on the following assumptions:

- Generate the business logic model by creating the resource dependency model and merging it with the mission dependency model, which represents the business functions and processes of the enterprise and their interdependencies.
- Assess the criticality of the assets and determine the most critical assets in the infrastructure.
- Assess the impact probability of an external event on the different business entities of the company: *Business Functions*, *Business Processes* and the *Business Company*, and evaluate the impact when duplicating and backing up assets in the infrastructure.
- Calculate the critical time by deducing the shortest path from the shock event to the *Business Company*.

For the literature survey, we examined contributions from various sources of information, such as academic articles, books and case studies. We identified methodologies and works assessing the impact of cyber-attacks on companies. We conducted our search using keywords and technical terms such as *Impact Assessment*, *Risk Quantification*, *Business Logic Modeling*, *Financial Assessment Methodologies* and *Operational Impact Propagation Assessment*. We searched for publications published between 2004 and 2023.

The chapter is organized as follows. Section 2 surveys relevant related work. Section 3 describes our contribution. Section 4 presents a proof-of-concept tool, implementing all the concepts and models of our contribution. Section 5 discusses future directions for research. Section 6 concludes the chapter.

2 Related Work

In this section, we discuss two complementary family methodologies for assessing the impact of an cyber-attacks affecting the business functions and processes of a company given company. The first family focus on the quantification of financial aspects associated to the perpetration of attacks, while the second family focus on operational impact assessment. Tables 1 and 2 summarize the references and findings covered for each of these two families, as covered in this section.

2.1 Financial Impact Assessment

Earlier research on quantifying the impact of cyber-attacks starts with a focus on financial aspects. Some representative contributions in the literature are presented next (and summarized in Table 1).

Table 1: Related work with a focus on financial impact assessment.

References	Contribution	Methodology	Metrics
– Brian *et al.* [3], (2004) – Rainer *et al.* [1], (2008)	– Survey the economic security metrics and the state of knowledge on the cost of cyber-attacks.	– Use of security metrics and models for security investments and cyber-risk measurement	– Annual Loss Expectancy (ALE) – Return On Security Investment (ROSI)
– Freund and Jones [8], (2014)	– Propose the Factor Analysis of Information Risk (FAIR) metric and how to use it for impact assessment.	– Discuss risk management using FAIR – Describe ontologies and terminology of the FAIR framework – Leveraging FAIR in risk decision-making and risk management	– Factor Analysis of Information Risk (FAIR)
– Dongre *et al.* [7], (2019)	– Develop a cost function to quantify the cost of the impact of data breaches.	– Cost function as the sum of costs incurred by providers and consumers. – Identify the cost components of the cost function for provider and consumers. – Present two case studies: Equifax data breach (2017) and the Target data breach (2013).	– Cost function
– Orlando [18], (2021)	– Analysis of the role of Cyber Value at Risk (Cy-VaR) model in quantifying the cyber risk.	– Definition of the role of the Cy-VaR model. – Highlight issues and difficulties in estimating Cy-VaR. – Description of the role of Cy-VaR in supporting security investment decisions.	– Cyber Value at Risk (Cy-VaR)

Brian *et al.* [3] and Rainer *et al.* [1] focus on their works on economic security metrics. They conduct a survey of the state of knowledge on the cost of cyber-attacks. Two models will be discussed in our work: *Annual Loss Expectancy* (ALE) and *Return On Security Investment* (ROSI). The ALE is represented as a quantitative metric for IT security, that can calculate the expected loss due to a risk in one year. Rainer [1] considers ROSI as a methodology for determining whether a firm should invest in implementing a security measure or not, i.e. it can be used to support a decision for or against implementing a security measure. Freund and Jones [8] the Factor Analysis of Information Risk (FAIR) framework. The FAIR framework aims to analyze risk and provide quantitative risk analysis. It provides a foundational understanding of risk, as well as risk assessment and analysis. Dongre *et al.* [7] quantify the cost of the impact of data breaches. In this chapter, the authors focus on data breaches that have exposed personal information. They present a mathematical function that expresses the cost impacts of data breaches. The developed cost function quantifies the cost of data breaches for providers and consumers. Orlando [18] presents the Cyber Value at Risk (Cy-VaR) model and its role in quantifying and measuring cyber-risk in the cyber security domain. The Cy-VaR model could provide an estimation and quantification of losses caused by cyber incidents, as well as support security investment decisions.

2.2 Operational Impact Assessment

Operational impact assessment aims at assessing the impact propagation of cyber-attacks and evaluating the perturbation of such attacks against the activities of a given organization. Next, we survey some representative works in the related literature. Table 2 summarizes our survey.

Table 2: Related work with a focus on operational impact assessment.

References	Contribution	Methodology	Metrics
– Liu *et al.* [11], (2017)	– Layered-graphical modeling – Impact quantifiers	– Calculate the impact score (from NIST NVD) – Assign weight to missions – Map LEGs (Logical Evidence Graph) to BPDs (Business Process Diagram) – Compute cumulative mission impact	– CMI (short for Cumulative Mission Impact)

– Cao *et al.* [2], (2018)	– Extend Ref. [19] via CVSS – Implement a tool that automatically generates an interconnected graph (interconnects the attack graph and the entity dependency graph) and compute the impact scores of an attack on tasks – Assess attack impact via business processes	– Generate the interconnected graph – Prune the interconnected graph – Compute the impact score based on the CVSS score	– Impact Scores
– Musman *et al.* [17], (2011)	– Assess the impact of a cyber attack on a mission – Compute the impact (by measuring the measures of effectiveness)	– Model creation – Compute metrics by simulating system's mission under different initial conditions – Categorize attack effects into categories and modify the mission simulation depending on the category of the attack – Compute attack impact as the difference between nominal vs. system under attack	– MoE (Measures of Effectiveness)
– Jakobson [10], (2011)	– Assess the impact of cyber attacks – Compute impact propagation	– Compute direct attack impact – Compute nodes' impact propagation – Assess the impact of a cyber attack on a mission	– POC (Permanent Operational Capacity) – OC (Operational Capacity)

– Barreto *et al.* [6], (2013)	– Evaluate the mission impact	– Mission modeling – Collection of cyber and mission situation awareness & cyber impact assessment	– Cyber-ARGUS Framework Metrics
– Mukherjee and Mazumdar [16], (2019)	– Modeling the business process – Compute the *Security Concern* metric	– Identify the vulnerabilities – Analyze the possibility of exploiting vulnerabilities (by comparing the max_Effort with the minimum effort for exploiting a vulnerability) – Compute the impact on data items and software instances – Compute the impact on activities and information items (estimated from the impact on data items and software instances) – Calculate the Security Concern	– Security Concern
– Motzek *et al.* [9, 13–15], (2015–2018)	– Define a mathematical model for mission impact modeling – Assess the mission impact	– Use the Monte-Carlo approximation to compute the conditional probability: • Find paths leading to external shock events • Monte-Carlo simulation	– Conditional Probability Metrics

Liu *et al.* [11] illustrate the utility of a layered graphical model which has three layers: the upper layer, the middle layer and the lower layer that, on the one hand, model the tasks and missions and their inter-dependencies, and on the other hand, construct the attack scenarios and their inter-relationships in

order to compute the impact of attacks on missions. Cao *et al.* propose in [2] a method to assess the impact of attacks on business process by generating an interconnecting graph from the attack graph showing the possible attack paths from the vulnerabilities to the target, and the entity dependency graph that contains three layers: asset layer, service layer, and business process task layer, as well as the dependencies between these three layers and on each individual layer, and calculating the impact score of the attack on the tasks that compose a business process. Musman *et al.* [17] compare the measures of effectiveness for the simulation of a mission under baseline conditions and the measures of effectiveness for the simulation of a mission under attack to evaluate the impact of an attack on a mission. The impact evaluation in this paper was based on mission models created using BPMN.

Jakobson [10] presents the impact assessment of a cyber attacks on missions by using the impact dependency graphs. In this paper, he presents a framework that quantifies the impact of attack on directly attacked assets and also calculates the cyber attack impact propagation through the nodes of the Impact Dependency Graph using operational capacity. Mukherjee and Mazumdar [16] provide a hierarchical model of a business process, and the metric *security concern* which is introduced as a new metric for measuring the security of a business process. This metric represents the impact on the business process of vulnerability exploitations in the context of a threat scenario.

Barreto *et al.* [6] propose the cyber argus framework, which helps to understand how to assess the impact of a cyber attack on missions and which critical assets contribute the most to accomplish the tasks performed in a mission. In their work, the mission model is designed using BPMN. To avoid developing the mission ontology from scratch, Cyber-argus integrates the previous work of D'Amico *et al.* [5] and Matheus *et al.* [12] into its own architecture. Motzek *et al.* present in [9,13–15] a mathematical mission impact assessment model. Their model takes into account external shock events. Their contribution includes a probabilistic graphical model, which is generated from mission and resource dependencies. In the sequel, we present a novel contribution expanding the work of Motzek *et al.*, specifically, expanding the theoretical background for the Business Logic Modeling in [13, 14]. We also expand the approach, by generating a novel resource dependency model and automating the update of dependencies between technical assets and impact values.

3 Operational Assessment using Business Logic Modeling

To further improve the assessment of the operational impact of external events associated to organizational missions, we assume the need of adding business logic to describe the interdependencies between technical assets to those other business functions and processes. Hence, we extend previous work in Refs. [9, 13–15] with a novel Business Logic Model (BLM). More precisely, a new mission dependency model is derived, including the use of a novel resource dependency

model. A more detailed description of the two models, as well as an example of the BLM of an *Online Shopping* company, are provided next.

3.1 Resource Dependency Model

The resource dependency model represents dependencies between assets. These dependencies are defined using a traffic matrix that quantifies the amount of data exchanged between each pair of assets. The traffic matrix can be generated from, e.g., NetFlow data [4], representing network traffic flows (i.e., datagrams in packet-switched network) collected from routers. The matrix is eventually processed to build a probability matrix, which contains conditional impact probabilities. The list of assets in the resource dependency model and the amount of data exchanged between assets can be retrieved and updated periodically (for instance, every 24 hours).

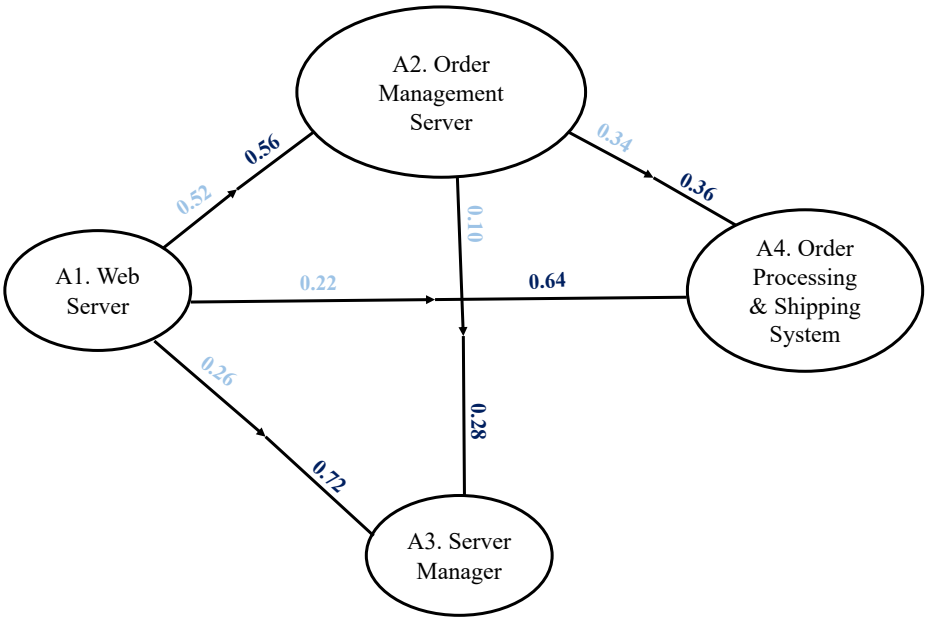

Fig. 1. Resource dependency model example. It provides a high-level representation of the interactions and dependencies between four different technical assets of the organization, depicted as vertices *A1* (*Web Server*), *A2* (*Order Management Server*), *A3* (*Server Manager*), and *A4* (*Order Processing & Shipping System*). The edges represent the interdependencies between assets. For a bidirectional impact, the value of the forward impact probability is colored in dark blue, and the value of the backward probability is colored in light blue. For instance, the impact probability from *A1* to *A3* is 0.72. The impact probability from *A3* to *A1* is 0.26.

Table 3. Matrix associated to the resource dependency model depicted in Figure 1.

	A1	*A2*	*A3*	*A4*
A1	0.0	0.56	0.72	0.64
A2	0.52	0.0	0.28	0.36
A3	0.26	0.10	0.0	0.0
A4	0.22	0.34	0.0	0.0

Figure 1 depicts an example of our resource dependency model. We assume it is obtained from the probability matrix shown in Table 3. The example contains four representative assets (identified as *A1. Web Server*, *A2. Order Management Server*, *A3. Server Manager* and *A4. Order Processing & Shipping System*) and their interdependencies. Interdependencies between assets are represented by the edges. For a bidirectional impact, the value of the forward impact probability is colored in dark blue, and the value of the backward probability is colored in light blue. For example, the impact probability from *A1* to *A2* is 0.56, and the impact probability from *A2* to *A1* is 0.52.

3.2 Mission Dependency Model

The mission dependency model describes the interdependencies between business functions and processes, as well as the assets that directly support business functions. A Business Process Model Notation (BPMN) model and expertise in the business activity of a company are needed to derive the list of business functions and processes and build the mission dependency model. The mission dependency model can be built manually, using such an expert knowledge from the company's business activities.

Table 4. Mission Dependency Model.

	BF1	*BF2*	*BF3*	*BF4*	*BF5*	*BF6*	*BP1*	*BP2*	*BC*
A1	0.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A2	0.0	0.76	0.61	0.0	0.0	0.0	0.0	0.0	0.0
A4	0.0	0.0	0.0	0.67	0.48	0.41	0.0	0.0	0.0
BF1	0.0	0.0	0.0	0.0	0.0	0.0	0.33	0.0	0.0
BF2	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0	0.0
BF3	0.0	0.0	0.0	0.0	0.0	0.0	0.72	0.52	0.0
BF4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.47	0.0
BF5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	0.0
BF6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.56	0.0
BP1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.68
BP2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.72
BC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

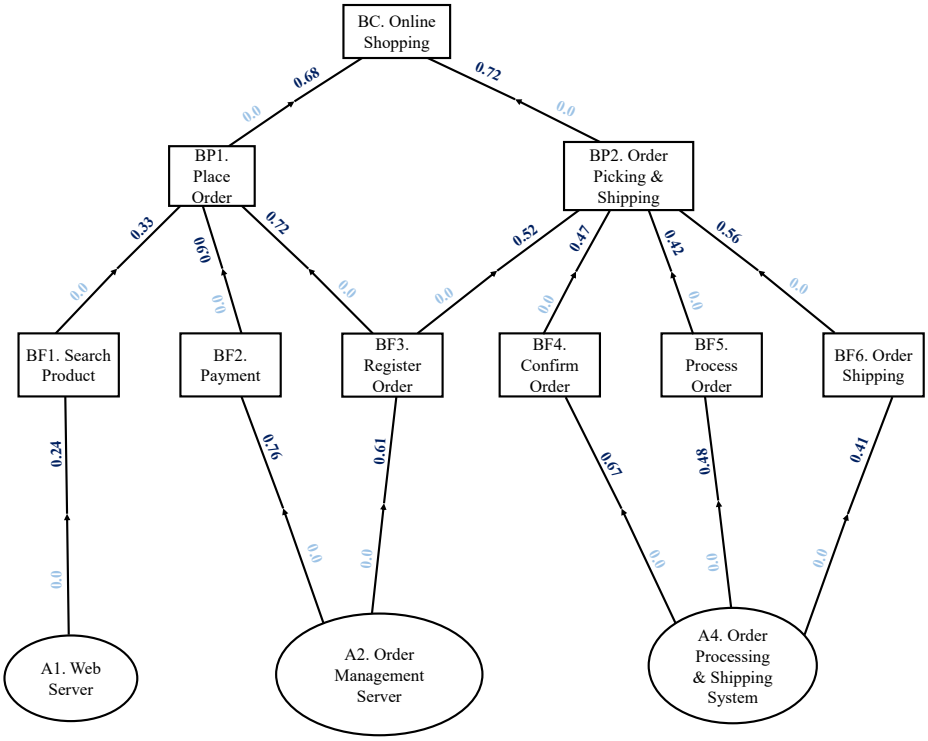

Fig. 2. Mission dependency model. It contains business functions, business processes, and assets directly supporting the business functions. The model also describes the interdependencies between those elements. In the depicted example, the model contains three assets (*A1. Web Server*, *A2. Order Management Server* and *A4. Order Processing & Shipping System*) that have a direct link with six business functions (from *BF1. Search Product* to *BF6. Order Shipping*), two business processes (*BP1. Place Order* and *BP2. Order Picking & Shipping*) and the global business of the company (*BC. Online Shopping*).

Figure 2 shows an example of a mission dependency model. The model is generated from the matrix in Table 4, containing the impact probabilities. The example contains the interdependencies between business functions and processes, as well as the assets that directly support business functions. In this example, we can see that three of the initial assets already identified in the resource dependency model (cf. Figure 1, assets *A1. Web Server*, *A2. Order Management Server*, and *A4. Order Processing & Shipping System*) have a direct link to six business functions (*BF1. Search product*, *BF2. Payment*, *BF3. Register Order*, *BF4. Confirm Order*, *BF5. Process Order* and *BF6. Order Shipping*). We can also see their link to two representative business processes (*BP1. Place Order* and *BP2. Order Picking & Shipping*) and one business company (identified as *BC. Online Shopping*), that represents the most important business function in the company.

The mission dependency model describes the dependencies between business processes and business functions, as well as between the business functions and the technical assets that directly support them. In Figure 2, the first asset (*A1. Web Server*) may have an impact on one business function (*BF1. Search Product*). Similarly, the second asset (*A2. Order Management Server*) may have an impact on two business functions (*BF2. Payment* and *BF3. Register Order*). Finally, the last asset (*A4. Order Processing & Shipping System*) can have an impact on three business functions (*BF4. Confirm Order*, *BF5. Process Order* and *BF6. Order Shipping*). Figure 2 also shows how business functions may impact business processes (e.g., impact of *BF1* over *BP1*, and *BF3,BF4* over *BP2*). It also shows that business processes *BP1* and *BP2* may have an impact on the company's mission (identified in our example as *BC*).

3.3 BLM Generation

Once the resource dependency model and the mission dependency model have been generated, they get fused into a single adjacency matrix representing the business logic model. The adjacency matrix generated from Tables 3 and 4 is summarized in Table 5.

Figure 3 represents the probability graph of the Business Logic Model built from the adjacency matrix associated to Table 5. The graph includes assets, business functions and processes, and the business company. It also depicts the dependencies between the different nodes on the graph. Figure 3 depicts the probability graph of the business logic model. This graph aims to help security experts to understand the impact of the attack and its propagation within the resources of an organization. Figure 4 displays the temporal graph associated to our business logic model, but highlighting the downtime tolerances instead of conditional probabilities.

As shown in Table 5, the business logic model requires an adjacency matrix presenting the interdependencies between the different nodes and containing the impact probabilities. Another adjacency matrix is required to provide the downtime tolerances between the nodes. The downtime tolerance represents an

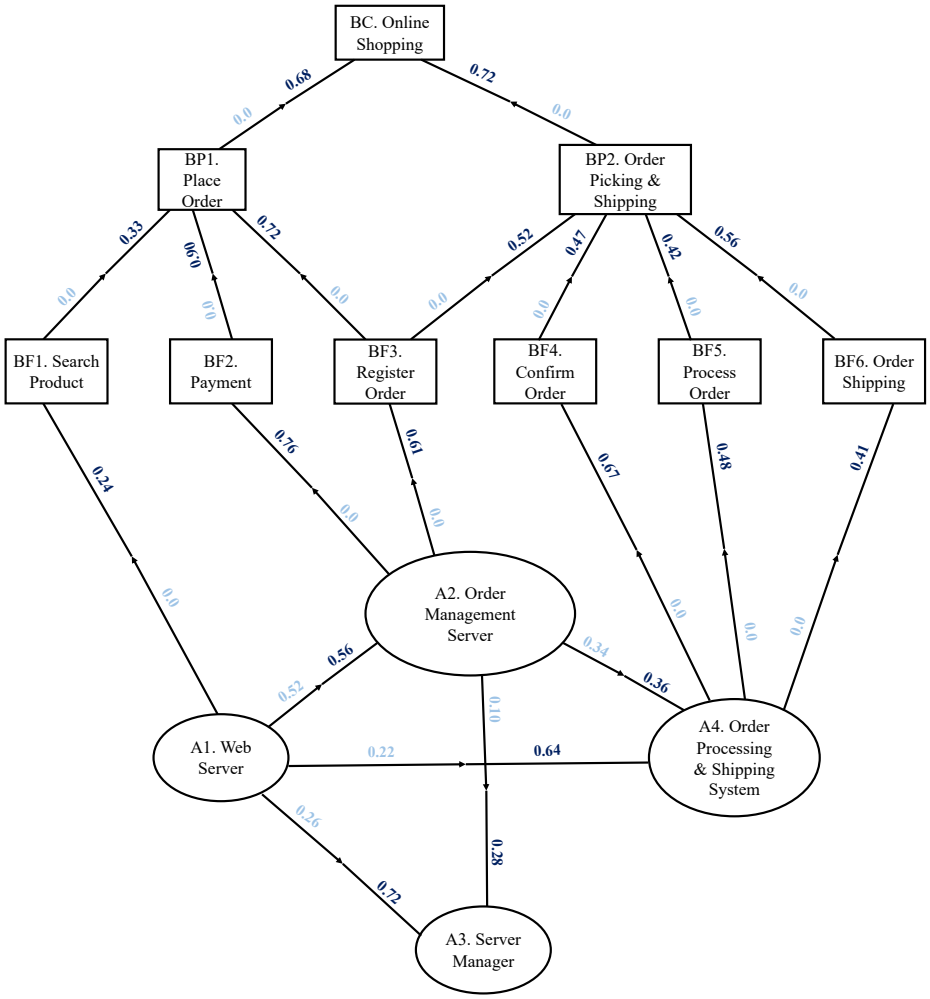

Fig. 3. Business Logic Model with conditional impact probabilities. This model represents the probability graph of the business logic model and it is generated from the resource dependency model and the mission dependency model. It includes assets, business functions and processes, business company, as well as dependencies between nodes, and contains the impact probabilities.

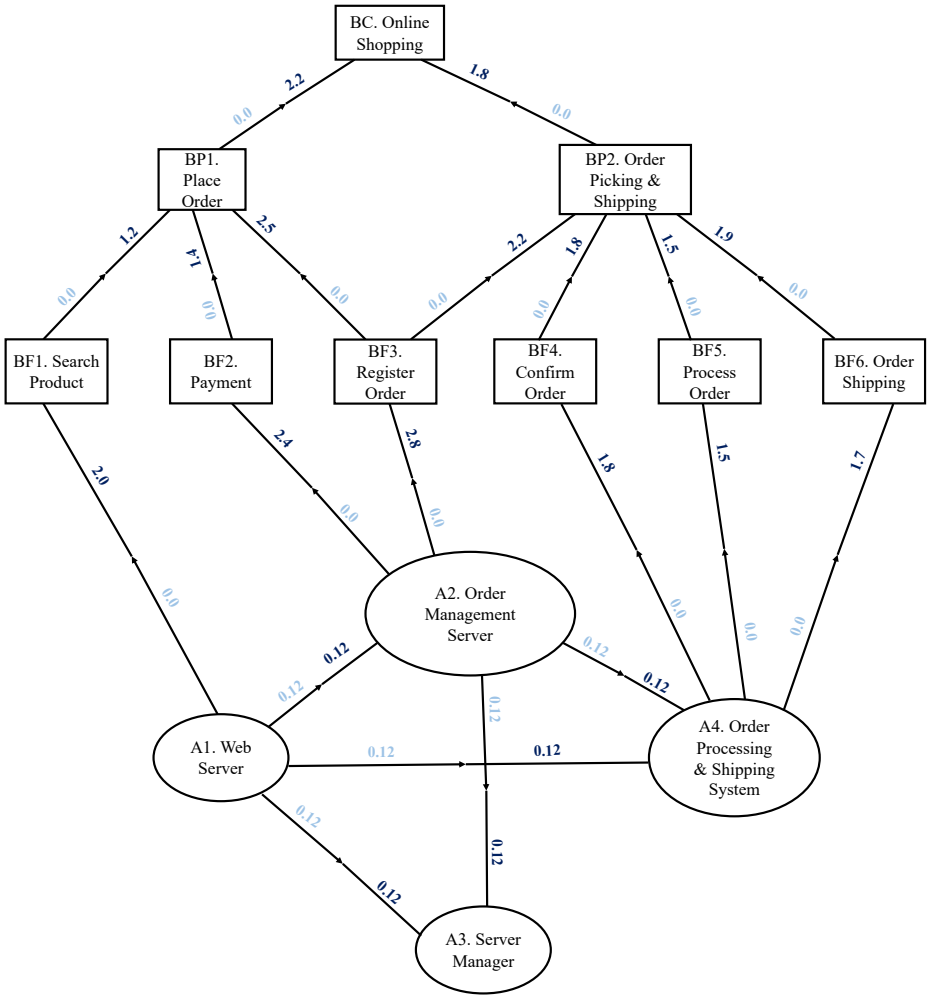

Fig. 4. Business Logic Model with downtime tolerances. This model represents the temporal graph of the business logic model and it is generated from the Business Entity downtime tolerance matrix and the Inter-asset downtime tolerance matrix. It includes assets, business functions and processes, business company, as well as dependencies between nodes, and contains the downtime tolerances.

Table 5. Complete adjacency matrix with conditional impact probabilities.

	A1	*A2*	*A3*	*A4*	*BF1*	*BF2*	*BF3*	*BF4*	*BF5*	*BF6*	*BP1*	*BP2*	*BC*
A1	0.0	0.56	0.72	0.64	0.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A2	0.52	0.0	0.28	0.36	0.0	0.76	0.61	0.0	0.0	0.0	0.0	0.0	0.0
A3	0.26	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A4	0.22	0.34	0.0	0.0	0.0	0.0	0.0	0.67	0.48	0.41	0.0	0.0	0.0
BF1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.33	0.0	0.0
BF2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0	0.0
BF3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.72	0.52	0.0
BF4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.47	0.0
BF5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	0.0
BF6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.56	0.0
BP1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.68
BP2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.72
BC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

estimation of how long the business functions or processes can be impacted without propagating the operational impact to other business entities. The adjacency matrix with downtime tolerances building the temporal graph depicted in Figure 4 is generated from the Business Entity downtime tolerance matrix and the Inter-asset downtime tolerance matrix. The Business Entity downtime tolerance matrix is presented by the same graph as the mission dependency model shown in Figure 4, but it displays the downtime tolerances instead of the impact probabilities. This model is also built manually because it requires knowledge of the company's business activities.

The inter-asset downtime tolerance matrix is presented by the same graph as the resource dependency model, shown in Figure 3, but highlighting the inter-asset downtime tolerances instead of the conditional impact probabilities. The inter-asset downtime tolerances values are set to 10% of the minimum interdependency value between the business entities, since the impact propagation delay between the assets is considered to be much lower than the impact propagation delay between the business entities. For instance, in our examples, the minimum interdependency value between the business entities could be set to 1.2 hours (or any other representative value extracted from the Business Entity downtime tolerance matrix) and the inter-asset downtime tolerances to 0.12 hours.

4 Implementation of our Approach

A proof-of-concept tool, hereinafter referred to as the *Business Impact Analyser* (BIA), is available on a companion code repository*. This tool implements all the concepts and models of our contribution. Next, we detail some additional

*A companion git repository with the code of the Business Impact Analyser (BIA) tool is available at https://gitlab.com/tsp-soccrates-components/bia

functionalities that have been included in the BIA tool, as well as an evaluation of performance and scalability associated to the tool.

4.1 Business Impact Analyser Functionalities

In addition to the business logic formalism underlying our contribution, the following additional functionality is also included in our models:

- Monitoring of assets criticality.
- Impact probability of shock events affecting business functions and processes.
- Monitoring of critical time.

Assets Criticality One of the tasks performed by the BIA tool is identifying the critical assets in the company. To identify the assets that contribute the most to the propagation of impacts on business functions and processes, our tool computes the value of the impact probability on the Business Company (BC) node when a shock event occurs on this asset with a local conditional probability equal to 1. The criticality value is between 0 and 1, in which 0 indicates that this asset has no impact on the BC, and 1 indicates that the impact of the attack is very high.

Impact Probability The existence of an external event may have an impact on the business functions and processes. To estimate how the impact of this external event will propagate in the architecture and to assess the impact propagation of this shock event to the missions, we compute the impact probability, which represents the degree to which this shock event impacts the missions. In order to assess the impact probability of a shock event on the missions, we implemented the Monte-Carlo approximation.

We have implemented the Monte-Carlo approximation to randomly explore the business graph a certain number of times (*ntimes*), while counting the number of times each node has been impacted by the shock events, which we want to assess their impacts on business entities. The steps of one graph exploration are defined as follows:

1. Initiate a queue Q, with the input list of Shock Events.
2. Explore node n, from Q.
3. For each edge of node n, try the impact probability of the edge by drawing a random number. If the drawn number is lower than the probability of impact of the tested edge, then the node at the other end of the edge is considered as impacted. If a node is impacted and was never added to Q, then add it to Q, and increase the impact counter for this new node.
4. While Q is not empty, repeat from Step 2.

The approximated impact probability is computed by taking the mean of the impact counter, i.e., *Impact Probability* gets as value *Impact Counter* divided by *ntimes*.

Critical Time The critical time represents the time when an impact to a business process or function will not have a significant operational impact on the company. To compute the critical time, we built the business logic model with the downtime tolerances, which is an estimate of how long the business functions or processes can be impacted without propagating the operational impact to other business entities. The Dijkstra algorithm is used to find the shortest path from the shock event to the Business Company and compute the critical time, which is equal to the sum of the downtime tolerance values between the nodes in the shortest path.

4.2 Applying the Functionalities

In this section, we show how to apply All the aforementioned functionalities over our contribution, in the example shown in Section 3.3, Table 5. The results of applying the functionalities are described next.

- **Assets Criticality Computation** — Figure 5 shows the probability graph after calculating the criticality of assets. Three levels of criticality have been configured to help the operator identify the assets that contribute the most to the propagation of impacts on the business entities. The criticality value is displayed in orange above each *Asset node*. According to this value, the assets are colored:
  - In red: The most critical asset is *A2. Order Management Server*.
  - In orange: The medium critical assets are *A1. Web Server* and *A4. Order Processing & Shipping system*.
  - In green: The low critical asset is *A3. Server Manager*.
- **Impact Probability Computation** — Figure 6 displays the probability graph after the computation of the impact probability of a shock event on the business entities.
  The shock event is given as input in this format: (Name of the shock event, target asset, impact probability). In this example, it is presented by: (*Shock Event*, *Order management server*, 0.84).
  After calculating the impact probability of this shock event on the business business, our tool displays the impact probability above each mission and according to the calculated values, it displays the missions in three colors:
  - In red: The business impact of this shock event on the business process *BP1. Place Order* is high.
  - In orange: The business impact of this shock event on the business functions *BF2. Payment* and *BF3. Register Order*, the business process *BP2. Order Picking & Shipping* and the business company *BC. Online Shopping* is moderate.
  - In green: The business impact of this shock event on the business functions *BF1. Search Product*, *BF4. Confirm Order*, *BF5. Process Order* and *BF6. Order Shipping* is low.

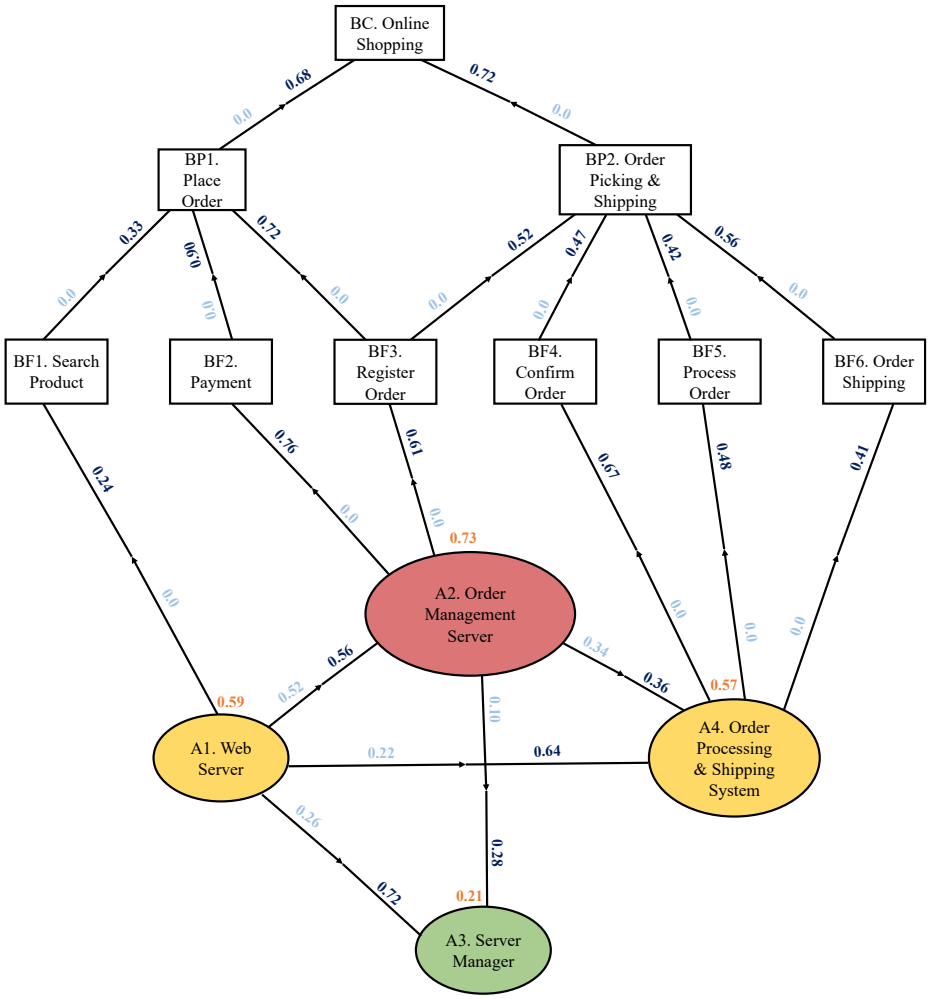

Fig. 5. Asset criticality computation. The computation of the criticality of assets aims to identify the most critical assets in the architecture. In this graph, the most critical asset, which is colored in red, is *A2. Order Management Server*.

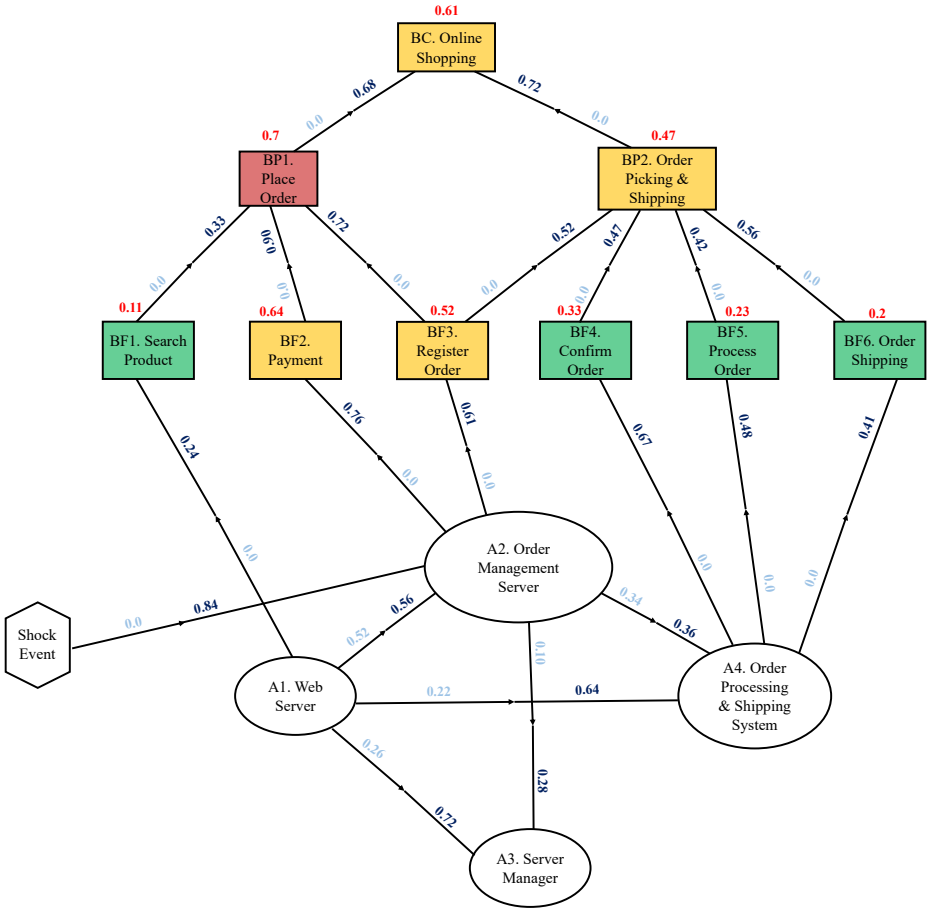

Fig. 6. Impact probability computation. The impact probability computation is performed on the probability graph using the Monte-Carlo approximation. This graph shows the impact of the Shock Event targeting *A2. Order Management Server* on the business entities. The most impacted business entity in this graph, which is colored in red, is the business process *BP1. Place order*.

– **Critical Time Computation** — Figure 7 shows the temporal graph after the critical time has been calculated. Our tool calculates the critical time value and displays the shortest path as a dotted line from the shock event to the business company. In this example, the shortest path is: *Shock Event -> A2. Order Management Server -> A4. Order Processing & Shipping System -> BF5. Process Order -> BP2. Order Picking & Shipping -> BC. Online Shopping* and the critical time, which is the sum of the downtime tolerances between the nodes in the shortest path, is equal to 5.02 hours.

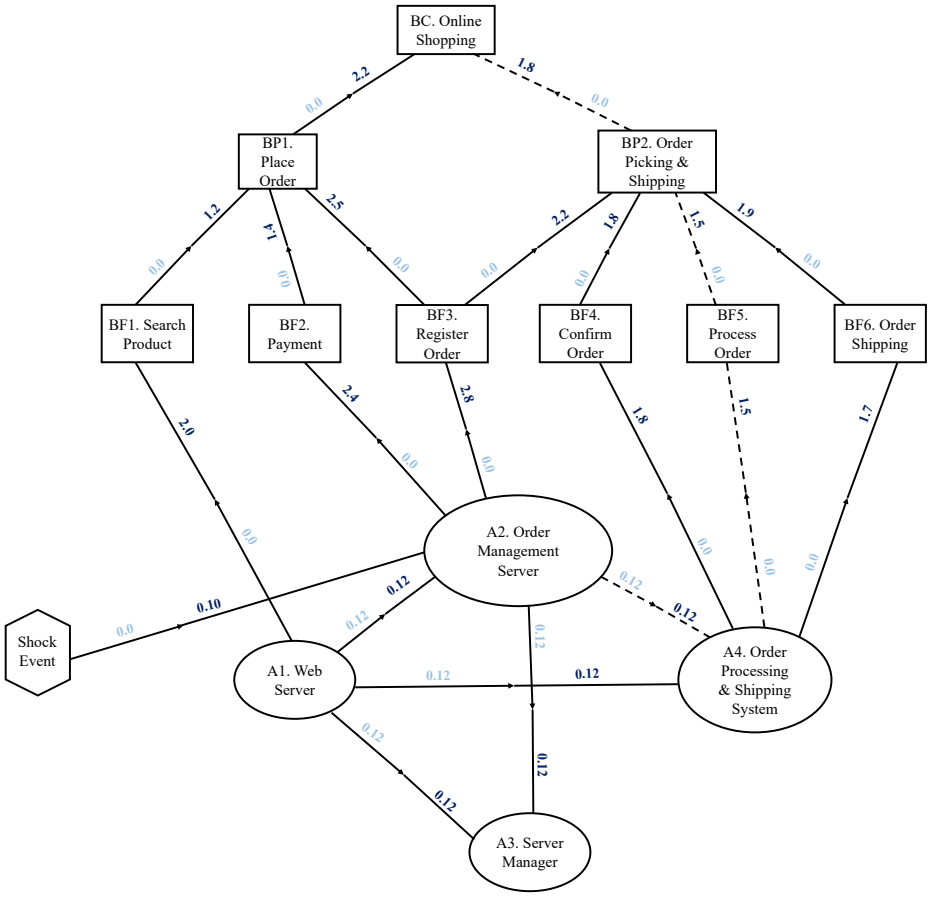

Fig. 7. Critical time computation. The critical time is calculated using Dijkstra's algorithm. The algorithm first finds the shortest path between the shock event and the Business Company *BC. Online Shopping*, and then calculates the critical time, which is equal to the length of the shortest path. The shortest path is displayed in dotted, and the critical time is equal to 5.02 hours.

4.3 Scalability Evaluation

In this section, we evaluate the scalability of our proof-of-concept tool[†]. We evaluate the main computations conducted by the tool, to estimate the time required to compute the criticality of assets of a representative example. The tool is implemented using Python REST (Representational State Transfer) APIs and the following external libraries: NetworkX[‡], NumPy[§], Uvicorn[¶], FastAPI[‖] and Pandas[**]. The evaluation is conducted on a 3-core CPU system, with 3 GB of memory and 20 GB of storage.

Table 6. Time spent to compute the criticality of assets depending on the number of assets and edges. This table shows the different tests performed by our tool, as well as the time required for each test to calculate the criticality of assets. To distinguish the edge density in the graph, three colors are used. The black rows indicate that for these tests, all assets communicate with each other, which means that all assets send and receive data. The orange rows indicate that for these tests, half of the assets communicate with each other, and the rest of assets only receive data. The green rows indicate that for these tests, only a quarter of assets communicate with each other, and the rest of assets only receive data.

# of Assets	# of Edges	Computation Time (in seconds)	Computation Time per Asset (in seconds)
50	2 450	1.96	0.039
50	1 224	1.09	0.021
50	613	0.69	0.013
250	62 250	24	0.095
250	31 124	12.2	0.048
250	15 562	6.45	0.025
1 000	999 000	197.6	0.197
1 000	499 500	101.9	0.101
1 000	249 750	52.4	0.052

Table 6 displays the results of various tests run by our tool with various models and numbers of assets and edges. It also displays the different tests performed by our tool and the time required to calculate the criticality of assets for each test. Each line of this table presents a test, with the size of the graph and the time required to compute the criticality of assets. The first and second columns display the number of assets and edges in the graph. The third column

[†]Available at https://gitlab.com/tsp-soccrates-components/bia
[‡]https://networkx.org/
[§]https://numpy.org/
[¶]https://www.uvicorn.org/
[‖]https://github.com/tiangolo/fastapi
[**]https://pandas.pydata.org/

shows the time required to calculate the criticality of all assets, and the fourth column shows the time required to calculate the criticality of one asset. For example, for a graph with 50 assets and 2450 edges, our method calculates the criticality of these 50 assets in 1.96 seconds. Several tests were performed with different edge densities. Three colors are used to differentiate the edge density in the graph:

- Black: all assets in the graph communicate with each other, which means that all assets send and receive data.
- Red: half of the assets in the graph communicate with each other, and the rest of the assets only receive data.
- Green: a quarter of the assets in the graph communicate with each other, and the rest of the assets only receive data.

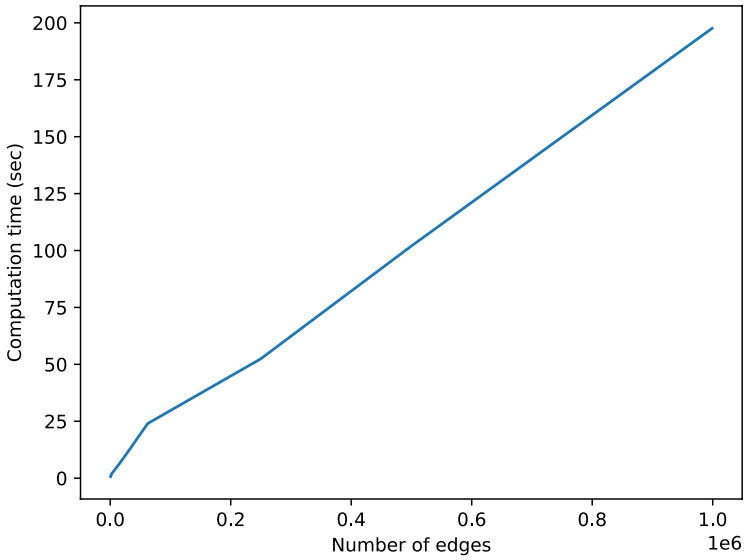

Fig. 8. Dependency between computation time and size of the graph. This plot shows the dependency between the time required to calculate asset criticality and the size of the graph. For example, for a graph with 1224 edges, our tool calculates the criticality of assets in 1.09 seconds.

Figure 8 displays the dependency between the computation time and the number of edges in the graph. The time required to compute the criticality of assets increases with the number of edges. F.i., the computation time for a very large graph with 1000 assets and 999000 edges is less than four minutes. These results confirm that our tool can perform computations in a reasonable time.

To sum up, we have presented in this section a practical implementation of our contributions in Section 3, in a proof-of-concept tool. We have validated

how to quantify the operational impact of shock events on a company's business entities using a representative example. One of the limitations of our work is the automation of models generation, for instance, the mission dependency model. Future research directions to address this limitation is discussed next.

5 Future Directions for Research

The mission dependency model is built manually because there is no easy method for automating the generation of this model. The generation of the mission dependency model requires defining the list of business functions and business processes, as well as the interdependencies between them. In order to define the list of the business functions and processes of the company, a lot of work needs to be done to derive these functions and processes from the Business Process Model and Notation (BPMN) model.

In addition to the difficulty of retrieving the list of business nodes, a high level of expertise is necessary to determine the relationships and the weight of dependencies between business nodes. In other words, and in addition to using BPMN, an expert knowledge on the business activities is also required to build manually the mission dependency model.

Building the mission dependency model can make it difficult to keep the model up to date if the business activity of the company changes. Defining and implementing a standard model capable that can automate the generation of the list of business nodes from the BPMN as well as evaluating the relationships and dependencies between these nodes would improve the reliability of our model without requiring a high level of expertise and skills in business aspects.

Downtime tolerance represents an estimate of how long the business functions or processes can be impacted without propagating the operational impact to other business entities. Computing the critical time using a dynamic downtime tolerances is considered a contribution for further research, which means that the values of the dependencies between the business entities, which are the downtime tolerances, are dependent on the time of the occurrence of the attack on the technical asset. For example, if a technical asset that supports business functions is targeted by an attack during the night, the propagation of the operational impact may not be very significant since these business functions may not be very necessary during the night. Thus, the downtime tolerance should be much longer than the downtime tolerance if the attack took place during the day when these business functions are very necessary to the company's activity.

Finally, a research work could be conducted on the quantification of the operational impact of attacks on business entities when the company has a redundant technical assets, which means the redundant asset will replace the asset targeted by the attack and perform its tasks, and in this case the operational impact will be very low on the business entities of the company.

6 Conclusion

In this chapter, we have surveyed existing methods aiming to assess the impact of external events on business entities. We have started our study with a focus on two main families: financial and operational impact assessment.

We have also presented a practical contribution for assessing the operational impact of attacks targeting the infrastructure assets and affecting the execution and performance of the company's activities. We have provided detail on a novel method to assess the operational impact of shock events on a company's business entities, based on adding to previous contributions a new business logic model. The new model enables the assessment of impact probabilities associated to external events on missions and the calculation of critical time, as well as the computation of the criticality of technical assets, that helps security analysts to identify the most critical assets. Our model have been tested and validated on realistic use cases and real data provided by stakeholders in a practical tool. The code of the tool is available online. We have also evaluated and validated the scalability of the computations performed by our tool, via several tests with different business logic models and on large graphs, and all of the tests performed show that our tool is able to perform computations and provide mission impact assessment in a reasonable time.

In terms of future research, we have pointed out to further automation in the generation of the mission dependency models, as well as the necessity to further assess the operational impact of external events against missions when a company has redundant assets, and further contribution to better compute the critical time using a dynamic downtime tolerances.

Acknowledgments — Authors acknowledge support from the European Commission (Horizon Europe projects SOCCRATES and AI4CCAM, under grant agreements 833481 and 101076911).

References

1. Rainer Böhme and Thomas Nowey. Economic security metrics. *Dependability metrics: Advanced lectures*, pages 176–187, 2008.
2. Chen Cao, Lun-Pin Yuan, Anoop Singhal, Peng Liu, Xiaoyan Sun, and Sencun Zhu. Assessing attack impact on business processes by interconnecting attack graphs and entity dependency graphs. In *Data and Applications Security and Privacy XXXII: 32nd Annual IFIP WG 11.3 Conference, DBSec 2018, Bergamo, Italy, July 16–18, 2018, Proceedings 32*, pages 330–348. Springer, 2018.
3. Brian Cashell, William D Jackson, Mark Jickling, and Baird Webel. The economic impact of cyber-attacks. *Congressional research service documents, CRS RL32331 (Washington DC)*, 2, 2004.
4. Benoit Claise. RFC 3954: Cisco systems netflow services export version 9, 2004.
5. Anita D'Amico, Laurin Buchanan, John Goodall, and Paul Walczak. Mission impact of cyber events: Scenarios and ontology to express the relationships between cyber assets, missions and users. In *5th International Conference on Information Warfare and Security*, pages 1–11, April 2010.

6. Alexandre de Barros Barreto, Paulo Cesar G da Costa, and Edgar Toshiro Yano. Using a semantic approach to cyber impact assessment. In *STIDS*, pages 101–108, 2013.
7. Siddharth Dongre, Sumita Mishra, Carol Romanowski, and Manan Buddhadev. Quantifying the costs of data breaches. In *Critical Infrastructure Protection XIII: 13th IFIP WG 11.10 International Conference, ICCIP 2019, Arlington, VA, USA, March 11–12, 2019, Revised Selected Papers 13*, pages 3–16. Springer, 2019.
8. Jack Freund and Jack Jones. *Measuring and managing information risk: a FAIR approach*. Butterworth-Heinemann, 2014.
9. Gustavo Gonzalez-Granadillo, Samuel Dubus, Alexander Motzek, Joaquin Garcia-Alfaro, Ender Alvarez, Matteo Merialdo, Serge Papillon, and Hervé Debar. Dynamic risk management response system to handle cyber threats. *Future Generation Computer Systems*, 83:535–552, 2018.
10. Gabriel Jakobson. Mission cyber security situation assessment using impact dependency graphs. In *14th international conference on information fusion*, pages 1–8. IEEE, 2011.
11. Changwei Liu, Anoop Singhal, and Duminda Wijesekera. A layered graphical model for mission attack impact analysis. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 602–609. IEEE, 2017.
12. Christopher J Matheus, Mieczyslaw M Kokar, Kenneth Baclawski, Jerzy A Letkowski, Catherine Call, Michael L Hinman, John J Salerno, and Douglas M Boulware. SAWA: An assistant for higher-level fusion and situation awareness. In *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2005*, volume 5813, pages 75–85. SPIE, 2005.
13. Alexander Motzek, Gustavo Gonzalez-Granadillo, Hervé Debar, Joaquin Garcia-Alfaro, and Ralf Möller. Selection of pareto-efficient response plans based on financial and operational assessments. *EURASIP Journal on Information Security*, 2017(1):1–22, 2017.
14. Alexander Motzek and Ralf Möller. Context-and bias-free probabilistic mission impact assessment. *Computers & security*, 65:166–186, 2017.
15. Alexander Motzek, Ralf Möller, Mona Lange, and Samuel Dubus. Probabilistic mission impact assessment based on widespread local events. In *NATO IST-128 Workshop: Assessing Mission Impact of Cyberattacks, NATO IST-128 Workshop, Istanbul, Turkey*, pages 16–22, 2015.
16. Preetam Mukherjee and Chandan Mazumdar. "Security Concern" as a Metric for Enterprise Business Processes. *IEEE Systems Journal*, 13(4):4015–4026, 2019.
17. Scott Musman, Mike Tanner, Aaron Temin, Evan Elsaesser, and Lewis Loren. Computing the impact of cyber attacks on complex missions. In *2011 IEEE International Systems Conference*, pages 46–51. IEEE, 2011.
18. Albina Orlando. Cyber risk quantification: Investigating the role of cyber value at risk. *Risks*, 9(10):184, 2021.
19. Xiaoyan Sun, Anoop Singhal, and Peng Liu. Towards actionable mission impact assessment in the context of cloud computing. In *Data and Applications Security and Privacy XXXI: 31st Annual IFIP WG 11.3 Conference, DBSec 2017, Philadelphia, PA, USA, July 19-21, 2017, Proceedings 31*, pages 259–274. Springer, 2017.