Project Title	Protection and privAcy of hospital and health iNfrastructures with smArt Cyber sEcurity and cyber threat toolkit for dAta and people
Project Acronym	PANACEA
Project Number	826293
Type of instrument	Research and Innovation Action
Topic	SU-TDS-02-2018
Starting date of Project	01/01/2019
Duration of the project	36
Website	www.panacearesearch.eu

D2.3 Advanced Response Methods

Work Package	WP2 Research on advanced threat modelling, human factors, resilient response and secure interconnectivity
Lead authors	Claudio Ciccotelli (UROME), Mara Sorella (UROME), Silvia Bonomi (UROME)
Contributors	
Peer reviewers	Gianluca Pozzolo (RINA-C), Matteo Merialdo (RHEA)
Version	V1.1
Due Date	31/03/2020
Submission Date	31/03/2020

Dissemination Level:

X	PU: Public
	CO: Confidential, only for members of the consortium (including the Commission)
	EU-RES. Classified Information: RESTREINT UE (Commission Decision 2005/444/EC)
	EU-CON. Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC)
	EU-SEC. Classified Information: SECRET UE (Commission Decision 2005/444/EC)

The work described in this document has been conducted within the PANACEA project. This project has received funding by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 826293.

Version History

Revision	Date	Editor	Comments
0.1	18/01/20	Silvia Bonomi (UROME)	Added ToC & initial document structure
0.2	02/03/20	Mara Sorella and Claudio Ciccotelli	Added content to sections 4.1, 5.1-5.7
0.3	04/03/20	Mara Sorella and Claudio Ciccotelli	Added section Example scenario
0.4	14/03/20	Silvia Bonomi	Added executive summary, Introduction and Conclusions, Improved text in Section 4
0.5	16/03/20	Mara Sorella and Claudio Ciccotelli	Revision and content improvements
0.6	16/03/20	Mara Sorella and Claudio Ciccotelli	Added content to Related Work, Scenario improved
0.7	17/03/20	Silvia Bonomi	Text Revision and release for internal review
0.8	23/03/20	Mara Sorella, Claudio Ciccotelli and Silvia Bonomi	Updated content to address reviewers' comments. Example scenario improved.
0.9	24/03/20	Mara Sorella, Claudio Ciccotelli and Silvia Bonomi	Section 1, Related Work and Conclusion improved and general review.
1.0	30/03/20	Silvia Bonomi, Mara Sorella and Claudio Ciccotelli	Final version released.
1.1	31/03/20	Silvia Bonomi, Mara Sorella and Claudio Ciccotelli	Added Figure 6 and typos correction. Final release.

List of Contributors

The list of contributors to this deliverable are presented in the following table:

Section	Author(s)
Executive Summary and Section 1	Silvia Bonomi (UROME)
Section 2	Silvia Bonomi, Claudio Ciccotelli, Mara Sorella (UROME)
Section 3	Silvia Bonomi, Claudio Ciccotelli, Mara Sorella (UROME)
Section 4	Silvia Bonomi, Claudio Ciccotelli, Mara Sorella (UROME)
Section 5	Claudio Ciccotelli, Mara Sorella (UROME)
Section 6	Claudio Ciccotelli, Mara Sorella (UROME)
Section 7	Silvia Bonomi (UROME)

Keywords

RESPONSE PROBLEM; COUNTER MEASURES; OPTIMIZATION PROBLEM

Disclaimer

Executive Summary

Risk Management standards (e.g., ISO 31000 [ISO31000]) and best practices consider, as last phase of the risk management process, the risk treatment i.e., the set of activities aiming to identify and select means for risk mitigation and reduction.

A preliminary task that could be executed before opting for a specific treatment is the definition and evaluation of one (or more) response plan i.e., a collection of remediation actions that can be implemented in order to reduce the analyzed risk.

Appropriate response strategies against new and ongoing cyber-attacks must be able to reduce risks down to acceptable levels, without sacrificing the organization mission. Unfortunately, almost all the existing approaches either evaluate impacts without considering missions' negative-side effects, or are manually based on traditional risk assessments, leaving aside technical difficulties. To the best of our knowledge, few efforts have been spent to bridge this gap and, in these cases, only technical remediation actions have been considered.

This deliverable takes a step in this direction by providing a formalization of the response problem (i.e., finding the best set of remediation actions) as a multi-objective optimization problem. In addition, we perform a series of transformations on the problem to achieve a form that is tractable and solvable through available optimization suites.

We also provide an example to show how the proposed method can be applied starting from a multi-layer attack graph such as the one defined in [D2.2] and finally we will briefly discuss how it can be implemented inside the Dynamic Risk Management Platform (developed in WP3) and in particular in the Resilient Response Engine.

Table of Contents

List of figures

List of tables

1. Introduction

1.1 Purpose

The purpose of this document is to report about research activities carried out in the task T2.3 - Resilient Response Analysis. In particular, T2.3 had its focus on the definition of response strategies aimed at reducing identified risks.

Our research is heavily based on the outcome of task T2.2 - Human Factors, Threat Models Analysis and Risk Quantification where a multi-layer attack graph has been defined to represent threats to Health Care Organizations (HCOs) and quantify related risks.

Once relevant risks have been identified and quantified, a set of possible countermeasures (also referred as *mitigation actions*) need to be defined to identify efficient and appropriate response strategies.

Let us note that, in order to have efficient and effective response capabilities, it is fundamental to consider an extensive set of countermeasures including both technical (e.g. acting on the IT infrastructure) and non-technical actions (e.g. acting at the organisational or procedural level with humans involved in the process).

This is necessary as not all the countermeasures will have the same benefit in terms of risk mitigation and not all of them can be actuated at a given time but they rather depend on other considerations like, for example, their direct cost (in terms of resources needed to implement the countermeasures) and indirect costs (in terms of impact on the organization business).

This document will describe how it is possible to leverage on the multi-layer attack graph model to identify and prioritize remediation actions aimed at improving the overall security level of an organization.

The techniques and the formalization reported in this document will be an input for the development tasks of WP3. Figure 1 shows the input/output dependencies between deliverables of WP1, WP2 and WP3.

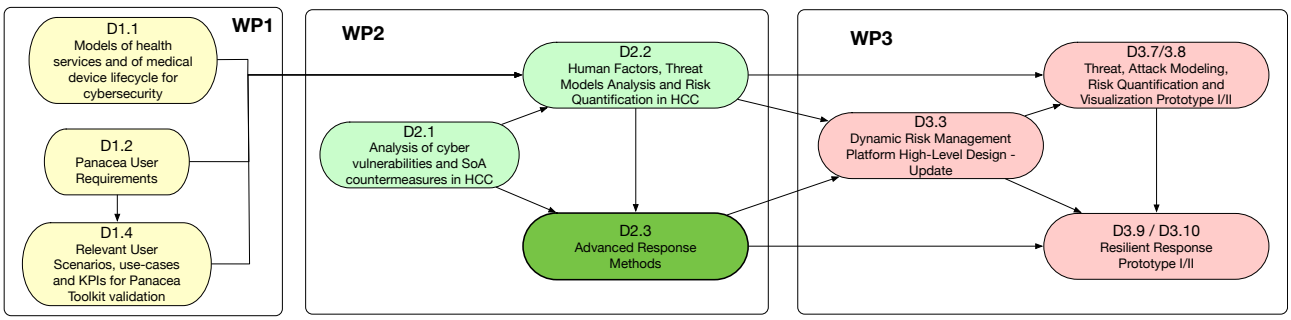


Figure 1 - WP and Deliverable Dependencies

1.2 Quality assurance

1.2.1 Quality criteria

The QA in the PANACEA project relies on the assessment of a work product (i.e. deliverable) according to lists of QA checks (QA checklists – [QAPeer]) established with the QAM, validated at a project management level and centralized in the [PMP].

For the purpose of the QA of this deliverable, it has been assessed according the following checklists:

- PEER REVIEW (PR) QA CHECKLIST [QAPeer]: this deliverable is a report, it then requires a proper peer review according to the checks defined in this checklist;

1.2.2 Validation process

For the final validation of work products (i.e. deliverables) within the PANACEA project, a final QA review process MUST be used before the issuing of a final version. This QA validation process follows the Quality Review Procedure established with the QAM and validated at project management level in order to guarantee the high-quality level of work products and to validate its adequacy according to the defined quality criteria chosen and defined for each deliverable. The Quality Review Procedure itself and the selection of the QA Review Committee are described in the [PMP]. The QA validation process is scheduled in the QA Schedule [QASchedule] managed by the QAM.

1.3 Structure of the document

This Document is structured as follows:

- Section 1 provides an introduction to the Deliverable.
- Section 2 reports about Applicable references and Document
- Section 3 provides the Glossary of Acronyms used throughout the document
- Section 4 introduces the response problem as well as the technical notation used
- Section 5 introduces and discusses the response problem as an optimization problem and discusses all steps that lead to the final formulation
- Section 6 provides an example scenario where we apply a specific formulation from Section 5 to find an optimal set of remediation actions
- Section 7 concludes the document.

1.4 General Background

Risk Management standards (e.g., ISO 31000 [ISO31000]) and best practices consider as last phase of the risk management process the risk treatment i.e., the set of activities aiming to identify and select means for risk mitigation and reduction.

When dealing with risk treatment, we can identify the following four options:

- *Risk Reduction*: its aim is to reduce the likelihood and/or consequence of possible incidents related to the identified risk. In particular, when implementing a risk reduction policy, the objective is to identify, for each relevant threat, a set of remediation actions that can be implemented either to remove factors

contributing to a possible vulnerability exploit (i.e., to reduce the likelihood) or to limit the impact on the business activities in case of an incident.

- *Risk Retention*: in this case, risks are simply accepted. This option is typically selected when risks have associated a negligible impact or when the cost of the mitigation is significantly higher than the cost of the mitigation.
- *Risk Avoidance*: in this case, risks are treated by avoiding the activity that gives rise to the risk in question (i.e., by acting on the threat source and avoiding it). Sometimes, this treatment is the only option for unacceptable risks.
- *Risk Sharing*: in this case, the treatment consists in transferring the risk (or parts of it) to another party, for example, covering the risk via insurance or sub-contracting.

A common task before opting for one of these four options is to evaluate which could be the optimal set of remediation actions that should be applied and its cost in order to decide which is the best option.

The set of remediation actions to be implemented against cyber-attacks must be able to reduce risks down to acceptable levels, without sacrificing the organization's business objectives in favor of security. Existing approaches either evaluate impacts without considering missions' negative-side effects, or are manually based on traditional risk assessments. Few efforts have been done in order to consider all these factors together, but all of them (to the best of our knowledge) just consider the possibility do deploy technical remediation actions.

In this Deliverable we are going to address the problem of identifying an optimal set of remediation actions (i.e., a response plan) that is able to reduce the risk associated with (a set of) identified threats by:

- Considering both technical and non-technical remediation actions and
- Taking into account both direct and indirect costs associated to each remediation action.

In particular, we will leverage on the multi-layer attack graph model (introduced and deeply discussed in [D2.2]) in order to identify the relevant threats represented by multi-step attacks involving the exploitation of vulnerabilities both at the network and at the human level. This has the direct effect to allow us to consider together both technical remediation actions (having their effect mainly at the network layer level) and non-technical remediation actions (having their effect mainly at the human layer level).

Finally, concerning costs of the mitigations, we will consider in the optimization problem both direct costs related to the implementation of a remediation action (e.g., the cost in terms of hardware and software resources needed to apply a patch or the cost incurred for personnel training) and the indirect cost of the mitigation (e.g., the cost incurred by the organization in terms of negative gain or penalty due to the unavailability of a certain business process for a certain period of time).

For the sake of completeness and in order to deliver a self-contained document, we briefly provide here an overview of the multi-layer attack graph model considered. Interested readers may find details in [D2.2].

The core principle behind the multi-layer attack graph is that an organization can be described as a complex composite object, made of different facets that we will call *layers*. Such layers are fundamental to describe all entities playing a key role in the context of cybersecurity analysis.

Our model considers four layers: (i) *human*, (ii) *network*, (iii) *access* and (iv) *business*.

The human layer has the aim to model the personnel of the organization. Typically, these individuals are linked by relationships, e.g., co-work, co-operation or other interaction-based relationship, which are either directly observable, or can be implicitly determined by the organization (e.g., spatial proximity).

The network layer has the aim to represent the ICT network infrastructure serving the organization mission and used by individuals represented in the human layer. Indeed, to carry out their job, individuals make use of a series of assets held by the organization, such as IT devices and medical devices. Those devices are typically networked, and linked to the organizational ICT infrastructure network, forming what we refer to as the network layer. The network layer is one of the primary targets of cyber-attacks.

Individuals are authorized to use assets via various kinds of access credentials, such as badges, tokens, or user accounts, which provide, to various extents, authorization/authentication mechanisms to the network assets. We call the set of such access credentials the access layer.

Finally, the business layer describes the set of business processes that support the organization mission. Business processes have dependency relationships between them, and typically rely on the correct functioning of software services and assets from the network layer, for example, medical devices, computer(s), or a set of networked equipment devices.

Figure 2 below provides an overview of these layers and their connections.

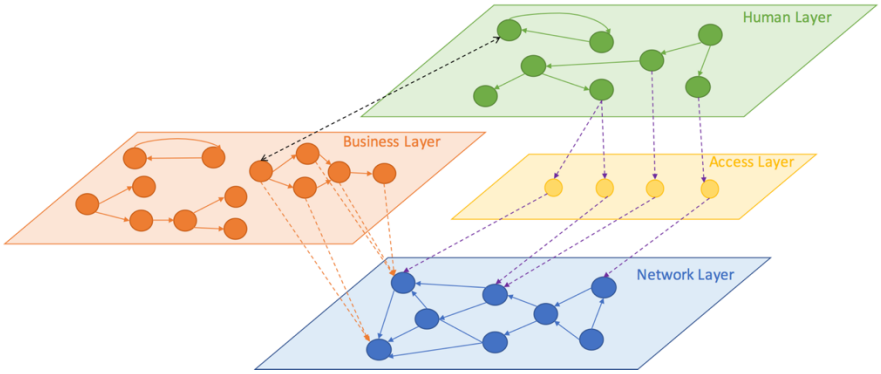


Figure 2 - Multi-layer attack graph Overview

2. Applicable and Reference Documents

2.1 Applicable Documents (ADs)

The following documents contain requirements applicable to the generation of this document:

Reference	Document Title	Document Reference	Version	Date
[PMP]	PANACEA Project Manager Plan		0.5	01/01/2019
[QAPeer]	PANACEA Peer Review QA Checklist		0.5	01/01/2019
[QAReqs]	PANACEA Requirements Review QA Checklist		0.5	01/01/2019
[QASchedule]	PANACEA QA Schedule		0.5	01/01/2019

Table 1: Applicable Documents

2.2 Reference Documents (RDs)

The following documents have been consulted for the generation of this document:

Reference	Document Title	Document Reference	Version	Date
[Almohri2016]	Security optimization of dynamic networks with probabilistic graph modelling and linear programming.			2016
[D2.2]	Panacea Deliverable D2.2: Human Factors, Threat Models Analysis and Risk Quantification			2019
[Fielder2016]	Decision support approaches for cyber security investment.			2016
[Gurobi20]	Gurobi Optimizer Reference Manual	http://www.gurobi.com		2020
[ISO31000]	ISO 31000 Risk management -- Principles and guidelines	https://www.iso.org/standard/ 65694.html		2018
[Khouzani2016]	Efficient numerical frameworks for multi-objective cyber security planning.			2016
[Khouzani2019]	Scalable min-max multi-objective cyber-security optimisation	https://doi.org/10.1016/j.ejor.2019.04.035		2019

Reference	Document Title	Document Reference	Version	Date
	over probabilistic attack graphs			
[Marler04]	Survey of multi-objective optimization methods for engineering	https://link.springer.com/article/10.1007/s00158-003-0368-6		2004
[Rakes2012]	IT security planning under uncertainty for high-impact events.			2012
[Sawik2013]	Selection of optimal countermeasure portfolio in IT security planning.			2013
[Schilling2016]	Optimal selection of IT security safeguards from an existing knowledge base.			2016
[Schilling2017]	A framework for secure IT operations in an uncertain and changing environment.			2017
[Schrijver99]	Theory of linear and integer programming	ISBN 978-0-471-98232-6		1999
[Viduto2012]	A novel risk assessment and optimisation model for a multi-objective network security countermeasure selection problem.			2012
[Zheng2019a]	A budgeted maximum multiple coverage model for cybersecurity planning and management	https://doi.org/10.1080/24725854.2019.1584832		2019
[Zheng2019b]	Interdiction models for delaying adversarial attacks against critical information technology infrastructure	https://doi.org/10.1002/nav.21859		2019
[Zonouz2009]	RRE: A game-theoretic intrusion Response and Recovery Engine	10.1109/DSN.2009.5270307		2009
[Chung2013]	NICE: Network Intrusion Detection and Countermeasure	10.1109/TDSC.2013.8		2013

Reference	Document Title	Document Reference	Version	Date
	Selection in Virtual Network Systems			
[Baykal-Guersoy2014]	Infrastructure security games	https://doi.org/10.1016/j.ejor.2014.04.033		2014
[Panaousis2014]	Cybersecurity Games and Investments: A Decision Support Approach	http://dx.doi.org/10.1007/978-3-319-12601-2_15		2014

Table 2: Reference Documents

3. Glossary of Acronyms

Acronym	Description
CVE	Common Vulnerabilities and Exposures
HCO	Health Care Organization
LP	Linear Program
MILP	Mixed Integer Linear Program
WP	Work Package

Table 3. Table of acronyms

4. Preliminaries

4.1 Related Works

Khouzani et al. [Khouzani2019] address the problem of identifying an optimal remediation plan (as a set of cybersecurity controls) that allows minimising the risk, while also minimising the costs associated to the implementation of the security controls. They propose a formulation based on a min-max multi-objective optimisation problem and show how to convert it in a Mixed Integer Linear Program (MILP) that can be solved efficiently. Our optimization problem formulation is strongly based on that of [Khouzani2019], where we adapted it to our model based on the multi-layer attack graph of [D2.2]. This allows us to account for the human factor when dealing with the problem of identifying the remediation plan. This represents an interesting improvement, since in many settings (such as in healthcare) even in presence of effective technical cybersecurity measures, the human (e.g., personnel) may still represent the weakest point and a potential source of vulnerabilities. The integration of our multi-layer model allows for a more accurate selection of non-technical remediation actions acting on the human layer. Moreover, we propose a methodology to compute the impact of reaching a given target node in the attack graph, based on our business layer model described in [D2.2], and we integrate it in the optimization problem definition.

Almohri et al. [Almohori2016] propose a similar min-max formulation and provide an approximate solution for the inner maximisation problem based on Taylor expansion and sequential linear programming. Differently from them, we adopt the approach proposed in [Khouzani2019] to convert the min-max problem into a MILP problem. This allows to solve the overall min-max problem more efficiently with respect to [Almohori2016].

A number of other single/multi-objective optimisation approaches have been proposed in the literature [Viduto 2012], [Rakes2012], [Sawik2013] [Khouzani2016], [Fielder2016], [Shilling2016], [Shilling2017]. However, all these approaches consider only *single-step* attacks, while with our multi-layer attack graph model we consider *multi-step* attacks.

Zheng et al. [Zheng2019a] address a problem similar to the budgeted version of our problem and propose a solution based on coverage models formulated as MILPs. Moreover, they present polynomial-time heuristics for identifying near-optimal solutions to the proposed models and a Benders branch-and-cut algorithm for (optimally) solving the expected-value versions of the proposed models. Differently from our solution, their approach requires an exhaustive enumeration of all the attack paths to define the problem (as they have a constraint for each path). This may limit the scalability of the approach, as the number of paths in an attack graph may grow exponentially with the number of nodes in many practical scenarios.

In another work, Zheng et al. [Zheng2019b] propose an approach based on a Stackelberg game (a two-stage zero-sum sequential game) between the attacker and the defender, where the attacker's objective is to complete his/her attack as soon as possible (by selecting the most quick attack path), while the defender's objective is to maximally delay the attacker by applying mitigation actions. They model the problem as a max-min optimization problem that (similarly to our approach) they transform in a nested max-max problem by dualizing the inner problem. Due to the large number of variables and constraints they propose a Lagrangian heuristic to efficiently compute near-optimal solutions, which decomposes the max-max problem into a number of easier subproblems. Differently from [Zheng2019b], our approach is based on reducing the cybersecurity risk rather than delaying the attacker, by modelling attacks in terms of likelihood and impact.

Other game theoretical approaches include [Zonouz2009], [Chung2013], [Baykal-Guersoy2014], [Panaousis2014]. Baykal-Guersoy et al. [Baykal-Guersoy2014] propose a solution, in the context of transportation infrastructures, based on a game between an adversary who aims to cause the maximum damage to a transportation network and a first responder that allocates resources to the transportation network sites to contrast the adversary. Panaousis et al. [Panaousis2014] model the problem of selecting the best

cybersecurity investment through a series of non-cooperative cybersecurity control-games between an attacker and a defender and identify the related Nash equilibria. The solutions to these games are used in a multi-objective, multiple choice Knapsack to determine the optimal cybersecurity investment. Finally, [Zonouz2009] and [Chung2013] address the online version of our problem, that is, they select remediation actions as soon as they detect attacks. Conversely, in this document we address the (offline) problem of identifying a remediation plan that allows to reduce the overall cybersecurity risk while minimising costs.

4.2 The Response Problem

In order to appropriately define the response problem, the following aspects should be addressed:

- The response problem must take into consideration one or multiple threats.
- Ideally, the solution to the response problem should provide the set of actions that should be put in place by the organization to reduce the risk related to all the possible threat sources.
- Every remediation action has a cost and there exist some threat sources that cannot be removed (e.g., a corresponding remediation action does not exist or it cannot be applied without violating business requirements).
- Many possible sets of remediation actions could be applied and there is the need to identify a criterion and a procedure to select the best one.

To accomplish these goals, we decided to formulate the response problem as a multi-objective optimization problem that accounts for the trade-off between risk reduction and incurred costs.

In particular, we will use the multi-layer attack graph described in [D2.2.] as reference threat model and we will use the following optimization parameters: (i) the direct and indirect costs of the remediation actions and (ii) the remediation actions efficacies.

4.3 Threat model

The multilayer attack graph described in [D2.2] models the attack paths that an attacker may exploit in order to reach his/her goal and compromise a target inside the organisation. We also consider the threat agent model and the corresponding attacker profiles defined in Section 6.1.1 of [D2.2] to estimate the likelihood of attacks.

To properly define the response problem, we will adopt a slightly modified version of the graph that we call the *extended attack graph.*

Given an attacker profile A, the extended attack graph is a directed multi-graph $G^A = (V, E, h, t, \Lambda^A, S, T)$ where:

- V is the set of vertices, corresponding to the whole set of vertices of the multilayer attack graph (i.e., union of the nodes of the human, access and network layer)
- E is the multi-set of directed edges. For the human and network layers, edges correspond to single *attack steps* i.e., possible exploitations of vulnerabilities that allow an attacker to gain privileges/ access to resources and to compromise nodes. Note that there can be multiple edges between two vertices, corresponding to the exploitation of different vulnerabilities.
- $h: E \to V$ is a function that returns the vertex that is the head of the edge (i.e., the “tip” of the arrow)
- $t: E \to V$ is a function that returns the vertex that is the tail of the edge
- $\Lambda^A = \{\lambda_e \in [0,1] \mid e \in E\}$ is a set where λ_e is the conditional success probability of the attack step corresponding to the edge e given that it has successfully reached the tail of e. This probability depends on the difficulty of exploiting the corresponding vulnerability, and on the considered threat

agent A. We omit the superscript A on λ_e for readability since it unambiguously refers to the threat agent A. These values can be estimated as described in Section 6.2.1 of [D2.2], where we associate to each attack path a Markov chain, whose exit rates λ_e depends on the difficulty of exploiting the (atomic) attack steps of the corresponding attack path edges.

- $S \subseteq V$: a subset of the vertices specifying the possible initial states of an attacker (attacker entry points)
- $T \subseteq V$: a subset of the vertices labelled as targets (or sink vertices). Note that these vertices only belong to the network layer of the original multi-layer attack graph. Such vertices correspond to goal privilege states that an attacker may want to acquire on target assets.

Based on this graph and on the model of business dependencies between assets, services and business processes, (Section 6.1.2 of [D2.2]) we can estimate risk.

5. A multi-objective Optimization Problem to Identify Response Plans

In this section we will define the response optimization problem that we use to identify the response plan.

As we will see in the following, there are essentially two directions for optimization in the defence problem: *minimising risk*, but also *minimising the costs* of the remediation plan. To start, we can think of this problem as being decomposed into two *nested* optimization problems:

- The *inner* optimization problem is the *Attacker Problem*, that models the aim of the attacker to compromise a target asset. We assume that the attacker's strategy is to maximize the success likelihood of his/her attack, so we model this as a maximization problem.
- The *outer* optimization problem is the *Defence Problem*, whose aim is to select the best set of remediation actions to contrast the attacker, taking costs into account as a multi-objective optimization problem in its most general form, or as a constrained (i.e., budgeted) problem, in the special case where budgets are fixed and known beforehand.

In the following subsections we will discuss the Attacker Problem and the Defence Problem in details, from their initial definition, to a series of transformations that will allow us to write the final form of our optimization problem, which is tractable and solvable through the available optimization problem solvers.

5.1 The Attacker Problem

As already stated, we assume that the aim of the attacker is to maximize the likelihood of his/her attack. Note that this would require that the attacker has complete information on the organization infrastructure, an unrealistic assumption in most settings. However, this perspective allows to tackle the defence problem in a worst-case scenario. To this end, we model the attacker as an agent who aims at choosing the attack steps that, with highest probability, allow to compromise a target node. This corresponds to select the attack path (from any source node to any target node) that has the maximum likelihood. Therefore, the attacker problem models the problem of finding the path associated with maximum likelihood within the attack graph as an optimization (maximization) problem.

With the aim of simplifying the definition of the problem, initially we assume that there are only a single source and a single target node, i.e., $S = \{s\}, T = t$. Later, we will show how the general problem having multiple sources and multiple targets can be modelled as a single-source, single target problem. A path $p_{s \to t}$ is a sequence of directed edges $p = (e_1, \dots, e_j)$ starting from s and ending at t, i.e., $t(e_1) = s$, $h(e_i) = t(e_{i+1})$ and $h(e_j) = t$. A path describes an attack scenario: an attacker in the starting privilege state s, takes the sequence of atomic attack actions corresponding to the edges in that path, and reaches the target privilege state t. Each atomic attack step on edge e will succeed with probability p_e.

If we assume that the successes of the attack steps are independent from each other, then the overall probability of success of an attack will be the product of atomic attack step probabilities over the attack path.

Let $\mathcal{P}_{s \to t}$ be the set of all paths leading from s to t. We we will call the *Attacker Problem* the following maximization problem:

$$\max_{p \in \mathcal{P}_{s \to t}} \prod_{e \in p} p_e \qquad (1)$$

Note that this problem does not currently adequately represent risk, but rather only the objective function of an attacker of maximizing its attack likelihood. Indeed, it aims at determining the most effective attack path i.e., the path having the highest success probability of attack across all possible paths. To represent risk, we will further revise this problem to encompass impact on targets. In the following section we will also detail how we can calculate p_e.

5.2 Remediation Plans

The organisation seeks to reduce its cybersecurity risk, by choosing a plan of *remediation actions* (later *remediation plan*), that is, a set of security countermeasures. Remediation actions can be of different kinds and act at various levels of the organization; in particular we will distinguish between *technical* and *non-technical* ones. An example of technical remediation action is patching a vulnerability, while a non-technical one can be training personnel against phishing or adopting a new security policy.

Let $\mathcal{A}$ represent the set of all available remediation actions. A remediation plan x can be encoded using boolean variables x_i:

$$\boldsymbol{x} = \left(x_1, \dots, x_{|\mathcal{A}|}\right), x_i \in 0,1 \tag{2}$$

Where the i-th element of $\boldsymbol{x}$ is uniquely associated to the i-th remediation action (according to any ordering ord, such that if a is the i-th remediation action, $ord(a) = i$), and $\forall a \in \mathcal{A}$ variable $x_{ord(a)} = 1$ (i.e.,it belongs to the remediation plan) if remediation action a is selected to be implemented, and is 0 otherwise.

Each action corresponds to a specific security countermeasure that can affect a certain set of vulnerabilities. Note that, for the sake of simplifying the notation, in the following we will denote $x_{ord(a)}$ simply as x_a.

In practice, remediation actions have effect on edges of the attack graph by limiting the exploitation of attacks in the network. Furthermore, each action may have effect on multiple edges and an edge may be affected by multiple actions: we denote with $\mathcal{A}_e \subseteq \mathcal{A}$ the subset of actions that affect an edge e. For each edge e, let λ_e be the baseline probability of success of the attack step associated with that edge. Furthermore, for each edge we have a choice of possible remediation actions. The implementation of the chosen remediation plan can reduce this success probability depending on the efficacies of the single remediation actions. In particular, for each remediation action a we consider the *remediation efficacy* $p_{ea} \in (0,1)$, i.e., the probability that a will block exploitation attempts associated with edge e.

We can therefore express the overall conditional success probability of exploitation attempts on edge e as a function of $\boldsymbol{x}$:

$$p_e(\boldsymbol{x}) = \lambda_e \prod_{a \in \mathcal{A}_e} \left((1 - p_{ea}) \cdot x_a + (1 - x_a)\right) \tag{3}$$

The interpretation of eq. (3) is the following: suppose $x_a = 0$ (i.e., remediation action a is not in the plan), then the corresponding factor in the product will be 1. Conversely, when $x_a = 1$ for any $a \in \mathcal{A}_e$, the effect of the control is discounting λ_e by a factor $(1 - p_{ea})$ (i.e., reducing the success probability of exploitation attempts by the residual success probability, when action a is implemented). If no action a is selected for an edge e, then the success probability of exploitation attempts over that edge will be λ_e. We can then restate the attack problem as

$$A(\boldsymbol{x}) = \max_{p \in \mathcal{P}_{s \to t}} \prod_{e \in p} p_e(\boldsymbol{x}) \tag{4}$$

Where p_e from Problem (1) becomes a function of the remediation variables $\boldsymbol{x}$.

5.2.1 Costs of the remediation

The implementation of a remediation plan induces *direct* and *indirect* costs on the organization. The direct cost of a remediation action is its implementation cost i.e., the cost of performing the software/hardware change that must be implemented. An example of indirect cost of an action is a measure of the negative impact on the organisation of deploying that action. For example, patching a vulnerability might require rebooting a machine: the associated indirect cost can be the downtime of the services running on the machine. Another example has to do with the potential indirect impact of a non-technical remediation action on the organization personnel: implementing stricter password policies (for example, increasing the length of passwords, or their frequency of change) may lead to more staff requiring help desk support and therefore waste working time.

We express the direct and indirect costs of a security plan $\boldsymbol{x}$ with the following two linear functions:

$$D(\boldsymbol{x}) = \sum_{a \in \mathcal{A}} c_a^d \cdot x_a\,, \quad I(\boldsymbol{x}) = \sum_{a \in \mathcal{A}} c_a^i \cdot x_a \tag{5}$$

Where c_a^d and c_a^i are, respectively, the direct and indirect costs associated to remediation action a. Note that here we choose the most natural form of cost function, the sum of the costs of the actions chosen by the plan, however in cases where more information is available to the modeller, it can be replaced by any linear function of the security plan variables (2).

5.3 The Defence Problem

Based on the Attack problem $A(\boldsymbol{x})$ defined in (4), and the cost functions $D(\boldsymbol{x})$ and $I(\boldsymbol{x})$ we state the defence problem as the following multi-objective optimization problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \big(A(\boldsymbol{x}), D(\boldsymbol{x}), I(\boldsymbol{x})\big) \\ s.t. \quad & x_a \in \{0,1\}, \forall a \in \mathcal{A} \end{aligned} \tag{6}$$

There are several methods to solve a multi-objective optimization problem [Marler04]. We opted for the ϵ-constraint method, where we keep only one of the objectives ($A(\boldsymbol{x})$) to build a single objective optimisation problem, where the other objectives are transformed as constraints. The solution to the original multi-objective optimization problem is then built by iteratively increasing the constraints' bounds by a small amount in each step. This leads to the following series of single-objective optimisation problems, hereafter referred to as the *defence problem (ϵ-constraint formulation)*:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & A(\boldsymbol{x}) \\ s.t. \quad & D(x) \leq B_D, \\ & I(x) \leq B_I, \\ & x_a \in \{0,1\}, \forall a \in \mathcal{A} \end{aligned} \tag{7}$$

where B_D and B_I represents, respectively, the direct and indirect "budget", corresponding to the bounds that are increased with the ϵ-constraint method. Note that in cases where budgets are known (and fixed) beforehand it is sufficient to solve the single-objective problem (7) directly, without recurring to the ϵ-constraint method. We will solve an instance of this problem based on an example scenario in Section 0.

Since $A(\boldsymbol{x})$ is a maximization problem, the defence problem becomes a *min-max* problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \max_{p \in \mathcal{P}_{s \to t}} \prod_{e \in p} p_e(\boldsymbol{x}) \\ s.t. \quad & D(x) \leq B_D, \\ & I(x) \leq B_I, \\ & x_a \in \{0,1\}, \forall a \in \mathcal{A} \end{aligned} \tag{8}$$

To solve the min-max problem, we first show how to solve the inner attacker problem, then we will show how to solve the outer defence problem.

5.4 Solving the Attacker Problem

Even though the attacker problem has a non-linear objective function (involving a product between $\boldsymbol{x}$ variables), it can be *exactly* converted to an equivalent simple Linear Program (LP). We will show how to reach the LP formulation via subsequent transformations.

As a first step, we show how the attack problem can be simplified by maximizing over the single edges rather than over the paths over the attack graph, whose cardinality is exponential in the input size. To do so, we change the variables of the optimization problem to new variables $y_e, \forall e \in E$ associated to the edges and we write the following equivalent problem:

$$\begin{aligned} \max_{y} \quad & \prod_{e \in E} \left(y_e p_e + (1 - y_e)\right) \\ s.t. \quad & \sum_{e:h(e)=v} y_e - \sum_{e:t(e)=v} y_e = \begin{cases} 1, & v = t \\ -1, & v = s \\ 0, & \forall v \in V - \{s,t\} \end{cases} \\ & y_e \in \{0,1\}, \forall e \in E \end{aligned} \tag{9}$$

To see why this problem is equivalent, recall that the attacker problem is the problem of finding the path $p_{s \to t}$ from the source s to the target t with the highest likelihood. In this new formulation, each variable y_e is 1 if the corresponding edge is part of the attack path selected by the attacker and 0 otherwise. Therefore, the objective function is equal to the product of the p_e of the edges for which $y_e = 1$.

Let us analyse the set of constraints, which ensures that the chosen edges consistently form a path:

$$\sum_{e:h(e)=v} y_e - \sum_{e:t(e)=v} y_e = \begin{cases} 1, & v = t \\ -1, & v = s \\ 0, & \forall v \in V - \{s,t\} \end{cases}$$

In particular, the third constraint enforces that for each vertex along a path which is not the source nor the target, the number of chosen edges (i.e., those whose $y_e = 1$) exiting the node should be equal to the number of entering edges whose $y_e = 1$. The first two constraints, on the other hand, ensure that for the source (target,

respectively), the number of chosen edges exiting the node must be 1 more (less, respectively) than the chosen entering edges.

Thus, either there is no $s \to t$ path in the attack graph, in which case both the original problem and this derived problem are infeasible, or this set of constraints forces the edges along an $s \to t$ path to have $y_e = 1$ and all other edges to have $y_e = 0$.

Furthermore, the only y's equal to 1 will be edges along the path maximizing the objective of Problem (9).

Note that the formulation does not explicitly forbid loops in the attack path, however, thanks to the maximization objective there will be no loops in the solution, as, due to the product in the objective, a path with a loop can never be more profitable than the corresponding acyclic path.

The problem is still non-linear, as we still have a product in the optimization function. However, since $\log(x)$ is strictly monotone for $x > 0$, we can equivalently maximize the logarithm of the objective function, which converts the product to a sum:

$$log\left(\prod_{e\in E}\left(y_e p_e + (1-y_e)\right)\right) = \sum_{e\in E}\log\left(y_e p_e + (1-y_e)\right)$$

Note that, since $y_e \in 0,1$, $\log\left(y_e p_e + (1-y_e)\right) = y_e \log p_e$. Indeed, for $y_e = 1$ we have $\log(p_e)$, while for $y_e = 0$, we have $\log 1$ which is 0. Therefore, to solve problem (9), we can solve an equivalent problem with $\sum_{e\in E} y_e \log p_e$ as objective function and subject to the same constraints as problem (9).

The resulting problem is still an Integer Linear Programming problem. However, this problem admits an *exact* relaxation to a Linear Programming (LP) problem due to Theorem 19.1 in [Schrijver99] (totally unimodular constraint matrix A and integer vector b). The exact relaxation corresponds to the following linear program:

$$\begin{aligned} \max_{y} \quad & \sum_{e\in E} y_e \log p_e \\ s.t. \quad & \sum_{e:h(e)=v} y_e - \sum_{e:t(e)=v} y_e = \begin{cases} 1, & v = t \\ -1, & v = s \\ 0, & \forall v \in V - \{s,t\} \end{cases} \\ & y_e \geq 0, \forall e \in E \end{aligned} \tag{10}$$

5.5 Solving the Defence Problem

To convert the defence problem from a min-max to a minimization problem, we leverage the strong duality of LP to replace the inner maximization problem with its dual LP problem which is a minimization problem. The dual of the LP problem (10) is the following:

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & v_s - v_t \\ s.t. \quad & v_{t(e)} - v_{h(e)} \geq \log p_e \quad \forall e \in E \end{aligned} \tag{11}$$

Moving from the attacker problem to the defence problem, we have that $\log\left(p_e(\boldsymbol{x})\right) = \log \lambda_e + \sum_{a\in\mathcal{A}_e} x_a \log(1-p_{ea})$. We can thus write the defence problem (ϵ-constraint formulation) as a Mixed Integer Linear Program (MILP):

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{v}} \quad & v_s - v_t \\ s.t. \quad & v_{t(e)} - v_{h(e)} \geq \log \lambda_e + \sum_{a \in \mathcal{A}_e} x_a \log(1 - p_{ea}) \quad \forall e \in E \\ & \sum_{a \in \mathcal{A}} c_a^d \cdot x_a \leq B_D \\ & \sum_{a \in \mathcal{A}} c_a^i \cdot x_a \leq B_I \\ & x_a \in 0,1 \qquad \forall a \in \mathcal{A} \end{aligned} \tag{12}$$

In the following section, we will further revise the problem to encompass attack impacts, and also allow multi-source attacks with multiple possible targets.

5.6 Encompassing Impact and Considering Multiple Sources and Multiple Targets

In this section, we show how to extend the defence problem (12) to consider multiple sources and targets, where targets may have different associated impacts. Encompassing impacts allows to express the problem in terms of risk (which account for both likelihood and impact) rather than just likelihood. In the following we show how to slightly transform the problem to consider multiple sources and multiple targets (with different impacts), and how this translates to adding two special nodes in the graph with associated weighted edges.

Without loss of generality, we combine all the potential attackers into one entity called the *attacker*. We do so by adding an extra node in the attack graph that we label as source σ, that we connect to each node in the sources set S through a special directed *entry edge*. We denote the set of edges outgoing from σ as E_σ. We also associate to each entry edge $e = (\sigma, s) \in E^\sigma$ an "entry rate" λ_s^σ , an analogous concept with respect to the exit rates over attack path edges, which models the likelihood that the attacker starts the attack from the $h(e)$ node. The single source σ, having outgoing edges towards all nodes, allows an attack to start from any node $v \in V$ with different probabilities. Note that, if an attack starting from a given node $v \in V$ is deemed impossible, it is sufficient to put $\lambda_v^\sigma = 0$. Moreover, we also identify a set D_T of devices or services (assets) in the organization's IT infrastructure as *target assets*. These represent the key functional assets of the company, and can correspond, for example, to all devices having a dependency on a subset, or possibly all the business processes of the organization. Like we did for the source, we also add a single node τ to represent a fictitious single global target asset. We then add edges from all nodes in the network layer of the attack graph that refer to target devices $d \in D_T$ (i.e., privilege states that can be acquired on d, previously corresponding to set T defined in Section 4.2) to the single target τ. We denote the set of such edges as E_T. To each edge $e \in E_T$, where $t(e) = t$, we associate a value $I_t \in [0,1]$ which represents the impact of reaching the target privilege level $t \in T$. A pictorial example can be found in Figure 3.

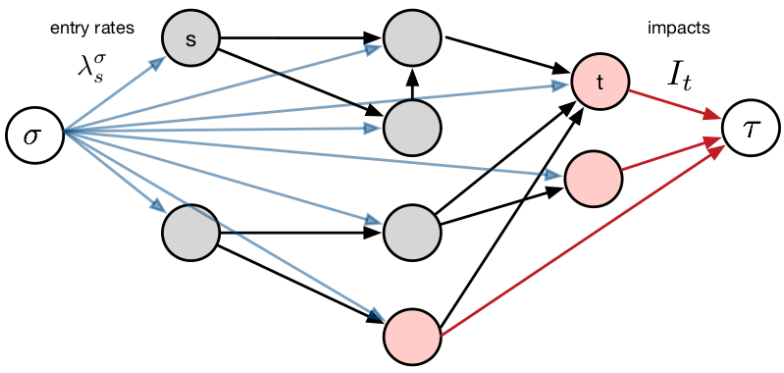


Figure 3: Example of an extended attack graph showing the added source and target nodes and their corresponding edges. Red nodes represent privilege state on target devices belonging to set D_T.

The two added nodes with the corresponding edges introduce minimal modifications to problem (12) that now also takes attack entry point probabilities and impacts on the considered targets into account for the computation of the optimal remediation plan. Following, the final form of the defence problem (ϵ-constraint formulation), encompassing all the above-mentioned considerations:

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{\nu}} \quad & \nu_\sigma - \nu_\tau & \\
s.t. \quad & \nu_\sigma - \nu_s \geq \log \lambda_s^\sigma & \forall s \in S, \\
& \nu_{t(e)} - \nu_{h(e)} \geq \log \lambda_e + \sum_{a \in \mathcal{A}_e} x_a \log(1 - p_{ea}) & \forall e \in E, \\
& \nu_t - \nu_\tau \geq \log I_t & \forall t \in T, \\
& \sum_{a \in \mathcal{A}} c_a^d \cdot x_a \leq B_D & \\
& \sum_{a \in \mathcal{A}} c_a^i \cdot x_a \leq B_I & \\
& x_a \in 0,1 & \forall a \in \mathcal{A}
\end{aligned} \tag{13}$$

5.6.1 Determining Impact using the Business Dependency Model

In the previous section we have shown how targets' impact fits into the defence problem. In this section we describe how to calculate impacts on target assets by leveraging the business dependency model described in [D2.2].

Let $t \in T$ be the target privilege levels, we define $V_{DG}^{EP}(t)$ as the set of all entry points ep of the dependency graph (Section 6.1.2 of [D2.2]) such that $t \in V_{AG}^{EP}(ep)$, that is:

$$V_{DG}^{EP}(t) = \{ep \in V_{SL}^{EP} \mid t \in V_{AG}^{EP}(ep)\}$$

Given a service level $s \in V_{SL}$ let $Dep_*^- \subseteq V_{SL}$ be the set of all nodes s' in V_{SL} such that there is a directed path from s' to s in the business dependency graph. That is, Dep_*^- is the set of all service levels that depend directly or indirectly (through the transitive property of dependence) on s.

Given a target privilege level $t \in T$, we can calculate I_t (that is the impact caused by an attack that reaches t) as:

$$I_t = \sum_{s \in V_{DG}^{EP}(t)} \sum_{s' \in Dep_*^-(s)} Impact(s')$$

where $Impact(\cdot)$ is defined in [D2.2].

6. Example scenario

In order to better understand our formalization, we will provide a simple example scenario based on the network depicted in Figure 4.

The network is composed of two internal local area networks (LAN1 and LAN2) where the first is devoted to workstations of medical staff (among which we consider a specific workstation labelled as W), and the second is the local network of one of the Radiology wards, and hosts various kinds of radiologic medical devices, among which a Tomography device, the Optima gt680 by *GE Healthcare* (labelled as T) and a server (S) devoted to storing radiology imaging data from all medical devices in LAN2. In particular, S is the only device in LAN2 reachable from LAN1 due to firewall rules of a firewall (F) which sits between the two LANs.

T is an important asset for the organization, as it allows the delivery of sanitary exams to the public, so we will consider it as a target (i.e. $D_T = \{T\}$) (in particular, we will consider as target the associated service level that mandates an availability guarantee on T).

Furthermore, a member (HS) of the medical staff who is responsible for the data acquisition of medical reports related to the tomographic exams has user access to the Lab PC S via credential CS.

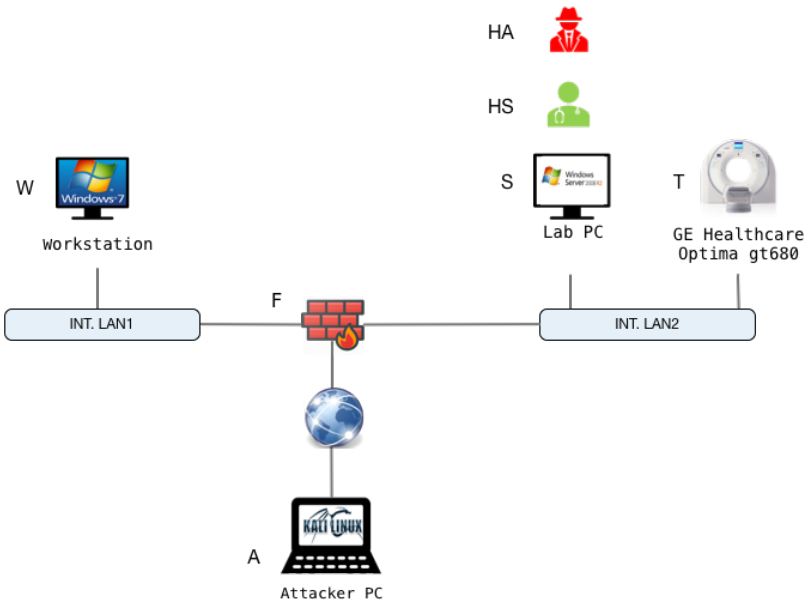


Figure 4 - Example Scenario (Network Topology)

The considered devices are affected by three CVEs:

- T has a web application (used for its administration) that is exposed on LAN2, and is affected by a *default credential* vulnerability (CVE1), allowing to obtain root access on the medical device;
- The operating system on S is affected by a network vulnerability (CVE2) that allows for arbitrary remote code execution and allows to obtain user access on the device;
- F has a vulnerability (CVE3) on a network service exposed on the Internet-facing interface, allowing for code execution and gaining user access.

Furthermore, a recent security assessment has revealed that HS has a tendency of leaving the PC unattended without logging out.

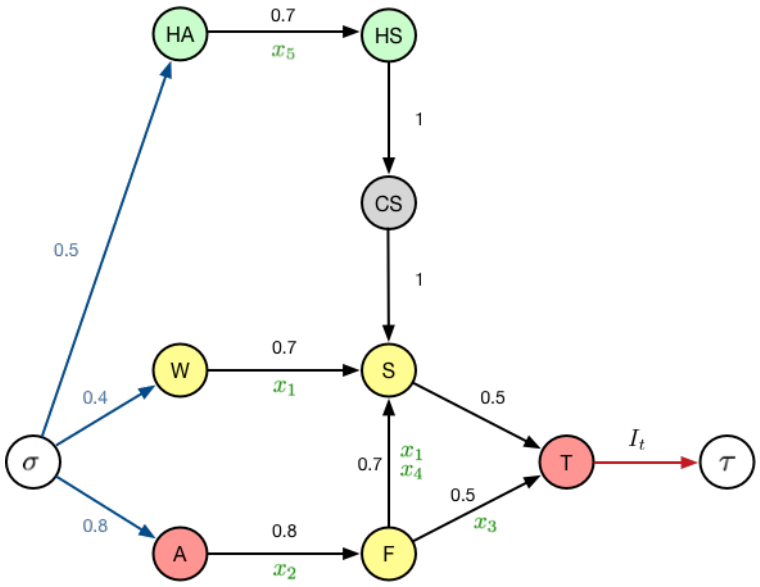
Three possible attack paths in this scenario are the following:

1. *Web attack*: an external attacker A can first exploit CVE3 on the firewall F and reach T to exploit CVE1 and, again gain root access and shut down T. Note that T is directly reachable from F due to misconfiguration of the firewall output chain allowing direct communications of F with LAN2.
2. *Insider threat 1*: an employee having user access on workstation W in LAN1 can perform a multi-step attack to T by connect to S and exploit CVE2, then use S as a pivot to connect to T and exploit CVE1 to obtain root access on the machine and shut it down.
3. *Insider threat 2*: an attacker HA having physical access to the Lab PC S can leverage the logged in session on S to perform the exploit against CVE1 of T.

The effect of all attacks is that T is made unavailable, impacting the target business process.

Figure 5 shows the extended attack graph of the considered scenario, which in its core part inherits the three layers of the original attack graph: the human layer (green nodes), the access (credential) layer (grey nodes) and the network layer (red and yellow nodes). Concerning the latter, network layer, *user* privileges acquired on devices are indicated with yellow nodes, while red nodes indicate *root* privileges. Concerning edges, we have three types of edges. The blue edges are entry edges (whose associated values are the entry rates). As we consider A, W and HA the only possible entry points, we show only their incoming edges (having nonzero entry rates). Black edges refer to the original attack graph edges with the corresponding λ values. The only red edge refers to the Impact of compromising T. Note that the interlayer edges (from HS to CS and from CS to S) have λ values equal to 1 as they are not associated to any vulnerability. Furthermore, remediation actions are shown in green: remediation action x_1 and x_2 refer to patching, respectively vulnerability CVE2 and CVE3, while x_3 and x_4 involve adding firewall rules that limit the reachability from F of, respectively, the web interface on T (exposed on port 80) and the port of the vulnerable service on S. Remediation action x_5 refers to performing training on best practices on software/hardware management (including the management of login sessions). Fixing CVE1 is not considered among the remediations as a corresponding patch is not available. The available remediation actions are summarized in Figure 5.

Note that other attack paths are also present in the graph. Moreover, note that for the sake of simplicity of the analysis we excluded from the scenario other humans plausibly involved, such as other personnel operating workstation W, firewall F and tomography device T.



action	effect	efficacy p_{ea}	d. cost	i. cost
x_1	patch CVE2	0.9	1	0
x_2	patch CVE3	0.9	1	0
x_3	¬reach(S,T)	0.7	1	0
x_4	¬reach(F,S)	0.7	1	0
x_5	train(cm) HS	0.8	1	0

Figure 5 - Attack graph of the example scenario. The table reports all available mitigation actions with efficacy and costs

For ease of interpretation, we set all direct costs of remediations to 1 and indirect costs to 0, and we assume to have a fixed budget. Therefore, we can solve problem (13), whose formulation for the considered scenario is the following:

$$\begin{aligned} \min_{x,\nu} \quad & \nu_\sigma - \nu_\tau \\ s.t. \quad & \nu_\sigma - \nu_w \geq \log(0.4), \\ & \nu_\sigma - \nu_a \geq \log(0.8) \\ & \nu_\sigma - \nu_{ha} \geq \log(0.5) \\ & \nu_w - \nu_s \geq \log(0.7) + x_1 \log(0.1) \\ & \nu_a - \nu_f \geq \log(0.8) + x_2 \log(0.1) \\ & \nu_f - \nu_s \geq \log(0.7) + x_1 \log(0.1) + x_4 \log(0.3) \\ & \nu_f - \nu_t \geq \log(0.5) + x_3 \log(0.3) \\ & \nu_s - \nu_t \geq \log(0.5) \\ & \nu_{ha} - \nu_{hs} \geq \log(0.7) - x_5 \log(0.8) \\ & \nu_{hs} - \nu_{cs} \geq \log(1) \\ & \nu_{cs} - \nu_s \geq \log(1) \\ & \nu_t - \nu_\tau \geq \log(1) \\ & 1 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3 + 1 \cdot x_4 + 1 \cdot x_5 \leq B_D \\ & x_1, x_2, x_3, x_4, x_5 \in \{0,1\} \end{aligned}$$

where B_D is the fixed budget that we vary from 1 to 5 to see different solutions picked by the optimizer. We solve the problem using the gurobi optimization suite [Gurobi20], for increasing budgets, by running it on the problem input in LP format (example for $B_D = 1$):

```
Minimize
       ni_sigma - ni_tau
Subject to
W:    ni_sigma - ni_uw >= -0.39794001
A:    ni_sigma - ni_ra >= -0.09691001
HA:   ni_sigma - ni_ha >= -0.30103000

WS:   ni_uw - ni_us + 1 x_1                    >= -0.15490196
AF:   ni_ra - ni_uf + 1 x_2                    >= -0.09691001
FS:   ni_uf - ni_us + 1 x_1 + 0.52287875 x_4   >= -0.15490196
FT:   ni_uf - ni_rt + 0.52287875 x_3           >= -0.30103000
ST:   ni_us - ni_rt                            >= -0.30103000
HAHS: ni_ha - ni_hs + 0.69897000 x_5           >= -0.15490196
HSCS: ni_hs - ni_cs                            >=  0
CSS:  ni_cs - ni_us                            >=  0

T:    ni_rt - ni_tau >= 0

BD:   x_1 + x_2 + x_3 + x_4 + x_5 <= 1
Binaries
      x_1 x_2 x_3 x_4 x_5
End
```

In the following we report the remediation plans chosen by the solver for increasing budgets B_D:

- For a **budget of 1** (corresponding, by our choice of costs, to 1 possible mitigation), the solver chooses the remediation plan $\boldsymbol{x_{r1}} = (0,1,0,0,0)$, formed by the single action x_2, (patch CVE3 on firewall F) effectively impacting the most efficient attack path existing in the graph (σ, A, F, T, τ).

- For a **budget of 2** the solution $\boldsymbol{x_{r2}} = (0,1,0,0,1)$ is chosen, that corresponds to perform training on HS to reduce the probability that he/she leaves the Lab PC S unattended without logging out, mitigating the risk of the *insider threat 2* attack path.
- For a **budget of 3** the solution $\boldsymbol{x_{r3}} = (1,1,0,0,1)$ is chosen, that by further patching CVE2 on S also mitigates the risk of the *insider threat 1* attack path, and the "extended" web attack passing through S, i.e., (σ, A, F, S, T, τ).
- For a **budget of 4**, the solution is $\boldsymbol{x_{r4}} = (1,1,1,0,1)$, which further impairs the *web attack* by disallowing reachability on edge (F,T) through the firewall rule.
- For a **budget of 5**, the trivial solution is selected, corresponding to performing all remediation actions.

On the Risk-Budget Trade-off

Figure 6 shows how the risk of the *max path* (i.e., the attack path associated to the maximum risk) varies for increasing values of budget in the considered scenario.

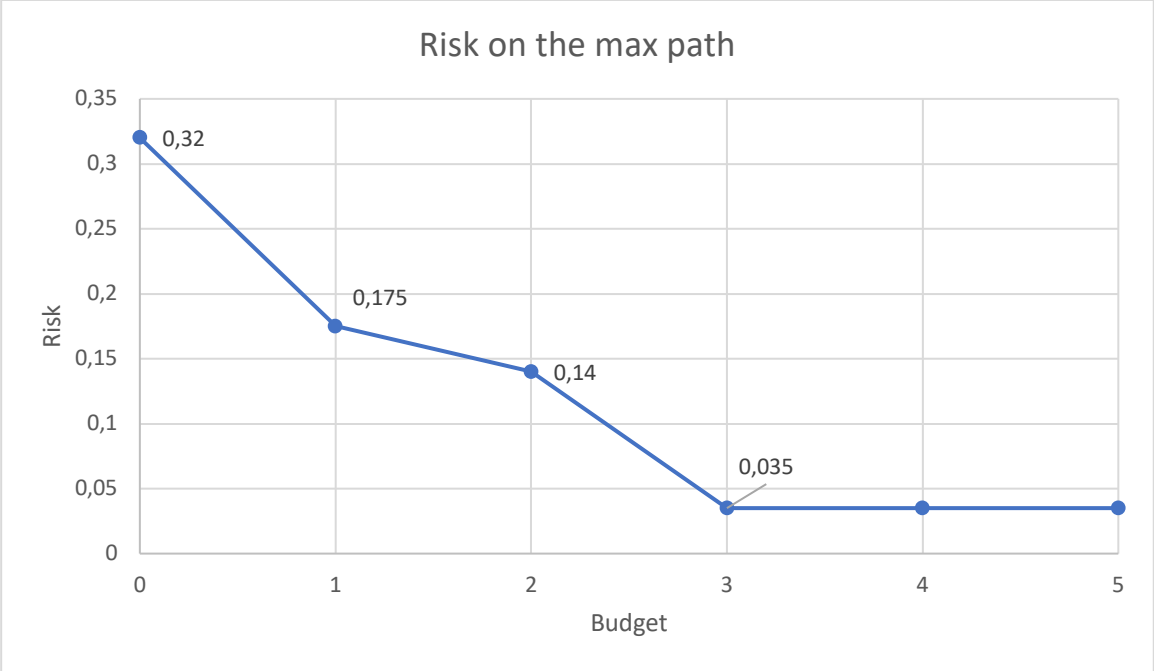


Figure 6 - Risk on the max path (path associated with the maximum risk) for varying budgets.

Initially, before selecting any remediation plan, the risk associated to the max path (σ, A, F, T, τ) is 0.32. With a budget of 1, the action x_2 is selected, reducing the risk on that path down to 0.032; indeed, according to equation (3), it drops of a factor of 0.1 (i.e., $1 - p_{ex_2}$). This leads to a new max path (σ, HA, HS, CS, S, T, τ) bringing the max path risk down to 0.175. With a budget of 2, the action x_5 is selected which further reduces the max path risk down to 0.14 with the new max path being (σ, W, S, T, τ). At this point, increasing the budget to 3, action x_1 is selected, causing the risk on path (σ, W, S, T, τ) to shrink from 0.14 down to 0.014. Note that the new max path, however, goes back to being (σ, HA, HS, CS, S, T, τ), causing the max path risk to be 0.035. Since there is no other remediation available on that path, it is bound to remain the max path from now on. However, the optimization framework is able to make further decisions considering the paths where other remediations are possible. Thus, while the max path risk does not decrease anymore, the risk associated to other paths may still decrease. For example, augmenting the budget to 4, causes the selection of remediation action x_3 on path (σ, A, F, T, τ) with associated risk dropping from 0.032 to 0.0096. Finally, for a budget of 5, remediation action x_4 is selected, which reduces the risk on path (σ, A, F, S, T, τ) from 0.00224 to 0.000672. Note that Figure 6 refers to the considered example and does not allow to take general conclusions. Moreover, the optimal trade-off cannot be determined a priori, since it depends on subjective choices, such as the acceptable level of risk that the HCO is willing to take.

7. Conclusions & Next Steps

This document addressed the response problem i.e., the problem of identifying a cost-effective set of remediation actions that can be implemented to reduce the risk associated to attack paths from a set of sources to a set of targets using the multi-layer attack graph model introduced in [D2.2].

We formalized the response problem as an optimization problem having the objective function of minimizing the risk of attack paths by selecting remediation actions (both technical and non-technical) with the lowest overall cost (considering both direct and indirect costs).

We also showed trough an example how the proposed approach is able to find the optimal plan.

The main innovation points behind the proposed approach can be summarized as follows:

1. *It allows to consider the response problem from a wider perspective with respect to other techniques available at the state of the art*. Being based on the multi-layer attack graph model introduced in [D2.2], the proposed approach is able to consider risks following from vulnerabilities affecting the software/hardware level but also vulnerabilities deriving from the interaction between humans and ICT networks. This allows to consider, evaluate and propose response plans including not only technical mitigation actions (like other approaches do) but also non-technical ones aimed at mitigating the effect of human-related vulnerabilities.
2. *It allows to consider the optimization from multiple sources*. Current state of the art approaches typically solves the optimization problem by fixing one source and one target. In addition, when studying the relaxation of the problem, we succeed to find a budgeted version that allows to solve it without requiring any approximation.
3. *It is tractable from the complexity point of view*. To the best of our knowledge, tractability of the optimization problem in the considered settings is currently an open issue and there exists in the literature only one available result that served as starting point for the development of our approach [Khouzani2019].

Let us remark that the proposed method is an input for the activities carried out in WP3 and in particular to the development of the Resilient Response component of the PANACEA Dynamic Risk Management Platform.

From a practical point of view, the proposed technique can be implemented in the resilient response component by leveraging on an existing MILP solver (e.g., CPLEX or gurobi). In order to do that, it is necessary to convert information coming from the multi-layer attack graph and from the pre-configured list of mitigation actions into variables of the optimization problem. In particular, the following tasks will be required:

- Starting from the analysis of the multi-layer attack graph, it is possible to identify, for each edge, a set of mitigation actions that could be applied. This set will then be used to define the set of variables needed in the optimization problem.
- Once the variable of the optimization problem has been identified, it is necessary to generate the specification of the optimization problem (i.e., a file written using the correct syntax based on the chosen MILP solver - typically LP format having .lp extension is accepted by most solvers).
- Once the problem specification is ready, it will be sufficient to pass it to the MILP solver that will produce as output the optimal solution.