POLITECNICO DI TORINO

Master's Degree in Ingegneria Informatica

Master's Degree Thesis

Hilti Internal Guided Risk Assessment tool: analysis, gaps and improvement definition.

Supervisor

Prof. Cataldo BASILE

Candidate

Giulia MEDDE

March 2023

Abstract

In today's world companies rely completely on automated information technology systems to achieve their goals and to build their business. Unfortunately, in parallel with the possibility to grow and provide efficiency to the business, there is also a black cloud of threats. Similarly to financial and reputational damages, cybersecurity risks can harm an organization's ability to innovate, gain and maintain customers, and at this purpose the risk management must be a central part of any organization's strategic management. It should be a continuous and developing process which runs throughout the organization's strategy and the possibility to automatize it using a tool provides the speed up of this process. For this reason, the Hilti internal Guided Risk Assessment tool is the central point of this thesis work, with the intention of making it as suitable as possible for a big international company as Hilti.

Table of Contents

Chapter 1

Introduction

The Guided Risk Assessment tool was introduced at Hilti in 2018, in order to support both project managers and IT risk & security team in assessing the risk of the solutions, preventing negative impacts for the company.

It is structured in different steps (performed by the Cybersecurity team firstly, and users implementing a solution secondly) that bring the tool to gain information about the solution to be assessed, identifying related vulnerabilities, calculating both the initial risk and the consequent risk accordingly to the eventual controls put in place to mitigate the vulnerabilities in scope.

If the initial purpose of the tool was to be focused on the risk calculation of a solution, over time it was mostly used to analyse the vulnerabilities identified for a solution, and put in place controls to mitigate them, to build robust and secure solutions within the organization.

Since the beginning of this thesis work, the difficulties of the Cybersecurity team in understanding the connections and the calculation behind the analysis provided by the tool, were clear. For this reason, the first part of my work was to analyse the tool and make clear the connections and the reasons behind not only the identification of the vulnerabilities for a specific solution, but also the calculation of the risk and the affection of the controls put in place, providing an explanation of how all the steps performed during the analysis take part in the process.

Since the documentation already present was not enough, I analysed the tool through a process that led me to go more in details at every step. If at the beginning I studied the documentation, and I used the tool to have an overview of its functionalities, then a work of reverse engineering was needed, performing test analysis, and changing parameters handled in the admin panel, together with an analysis of the database structure. Because of the complexity of the calculation and the connections, the only reverse engineering work in understanding the functionalities of the tool was not enough, and for this reason an in-depth analysis of the code was also needed.

Since the in-depth analysis of the tool made several gaps and inconsistencies clear, I carried on also a following phase, collecting all of them and identifying the reasons behind.

If at the beginning only the content of the pre-defined lists (e.g., Vulnerability list) and some UI problems seemed to be relevant, after the clarification of the functionalities and the purpose of the tool, I reviewed the entire strategy and structure, in order to identify the root cause of the problems. The collection of the gaps and inconsistencies, together with the problems in cascade causing them, brought me to put in place improvements not only in the content of the lists, but in the entire strategy and structure.

This thesis work will give the overview of the knowledge required to carry on this project, with an explanation of the frameworks and definitions studied and identified as relevant or not for the purpose of the tool. I provided also an explanation of the Guided Risk Assessment tool functionalities, in comparison with another tools already present in the market, and the collection of the gaps and inconsistencies identified, together with the improvements put in place. Due to the time limit of the thesis work, several improvements have been identified to be put in place in the future, and an overview of them is given to the reader in the last chapter of this work.

Chapter 2

State of the art: threat modelling, risk assessment and vulnerability assessment - frameworks and tools

In this chapter we will go through an overview of the meaning and application points of the "threat modelling", "risk assessment" and "vulnerability assessment" techniques, analysing the already existent frameworks and tools based on them. Before getting into the details of the different practices, an explanation of the most used terms is needed.

- Asset: Something that has value to the organization. An asset extends beyond physical goods or hardware, and includes software, information, people, and reputation.

  (Reproduced from [1])

- Vulnerability: A weakness of a control or asset.

  (Reproduced from [1])

- Threat: An activity, deliberate or unintentional, with the potential for causing harm to anautomated information system or activity.

  (Reproduced from [2])

- Risk: The probability that a particular security threat will exploit a system vulnerability.

  (Reproduced from [2])

- Control: Policies, procedures, and guidelines for managing risk.

  (Reproduced from [1])

- Confidentiality: Property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

  (Reproduced from [3])

- Integrity: Property of accuracy and completeness.

  (Reproduced from [3])

- Availability: Property of being accessible and usable upon demand by an authorized entity.

  (Reproduced from [3])

- CIA: Confidentiality Integrity Availability are the three pillars of security.

Talking about risk management and security, it's important to keep in mind that to have a 100% of protection, 0% of usability is needed (everything is blocked), and vice versa. For this reason, we have considered different kinds of manual analysis and relative methodologies, to find a trade-off between security and functionalities that could lead to the best possible coverage of the system.

Cybersecurity risks analysis is a process in which different steps are involved, consecutively.

Before starting with an overview of the analyses and relative methodologies and tools, it is important to highlight that as part of this thesis work, we started with the research and analysis of the tools the most important Hilti competitors and other big companies in manufacturing industry use, but since the research did not show any relevant results, we expanded our sphere of analysis and we focused on the ones most used by organizations in different application fields.

2.1 Threat modelling

First, a differentiation between the term "threat model" and "threat modelling" is required. The term threat model means "A structured representation of all the information that affects the security of an application. In essence, it is a view of the application and its environment through the lens of security." (Reproduced from [4])

On the other side, with the term threat modelling we are referring to "A process for capturing, organizing, and analyzing all this information. Applied to software, it enables informed decision-making about application security risks. In addition to producing a model, typical threat modeling efforts also produce a prioritized list of security improvements to the concept, requirements, design, or implementation of an application." (Reproduced from [4])

During the threat modeling process, step 1 in the cybersecurity risks analysis, some questions have been identified to help in the analysis conduction:

- What are we working on? - Assess Scope
- What can go wrong? - Identify what can go wrong
- What are we going to do about it? - Identify countermeasures or manage risk
- Did we do a good job? - Assess your work

Based on these questions, some methodologies and frameworks have been published, to create a rigorous guideline for the tools in the market.

We identified the most important five methodologies, inspired by different classifications as the one in article "Threat Modeling: Available Methods" [5], or in the "Threat Modelling and Risk Assessment" thesis work [6] :

- OCTAVE, developed at CMU for the United States Department of Defense
- PASTA, developed by VerSprite CEO and security leader
- STRIDE, developed by two engineers working at Microsoft
- TRIKE, published by Octotrike
- VAST, developed by the Chief Technical Architect of ThreatModel

2.1.1 Threat Modelling Methodologies

OCTAVE

The Operational Critical Threat, Asset and Vulnerability Evaluation "is a risk based strategic assessment and planning technique for security" (Reproduced from [4]), a framework for identifying and managing information security risks, described in [7] and [8]. It is mainly self-directed.

To lead organizations to identify the information assets and the vulnerabilities that may expose them to threats, it defines a method of evaluation flexible and targeted not only to information technology departments but also to business or operational departments.

It is structured in three different phases:

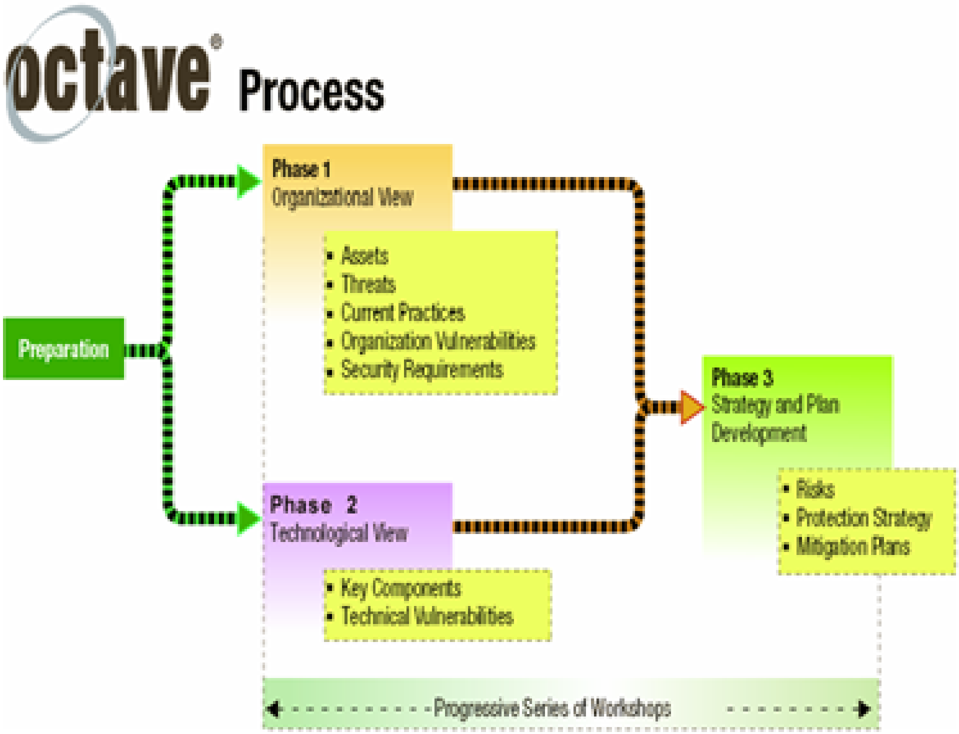


Figure 2.1: OCTAVE Phases. [8]

1. Build Asset-Based Threat Profiles It is an organizational evaluation based on the identification of the most important and critical assets to protect. For each of them the analysis requires to identify the possible threats, creating a threat profile.

2. Identify Infrastructure Vulnerabilities Is it an evaluation of the information infrastructure, analyzing network accesses, information technology components etc. related to each asset identified in the previous phase. Through this phase is possible to determine which components are resistant to network attacks.

3. Develop Security Strategy and Plans This phase leads to the identification of the risks and the creation of a protection strategy and mitigation plans.

The three phases are shown in the figure 2.1.

PASTA

The Process for Attack Simulation and Threat Analysis is a risk-centric methodology, structured as a step-by-step process for aligning business and technical requirements, to provide a dynamic threat identification, enumeration, and scoring. This methodology intent leads to an attacker-centric view of the system in order to develop an asset-centric mitigation strategy. [9] It is used by organizations all over the world [10], Gitlab can be an example as described in GitLab itself [11].

PASTA has seven steps, each of them acting as building block to another one:

1. Define the Objectives

   Objectives can be internally or externally driven, and the purpose is to clearly understand the important parts of the organization, giving the opportunity to include governance into the discussion. The suggestion during this phase is to start understanding the business objectives and then harmonize them with security requirements.

2. Define the Technical Scope

   The purpose of this step is to understand the possible attack surface by defining the technical scope, knowing what the organization is protecting.

   This phase can lead the organization to identify dependencies with third party services and highlights the importance of the collaboration effort of the different teams/departments.

3. Decompose the Application

   If step two takes to build the context around the organization, this stage is crucial to understand if you have implicit trust models, as an IoT device talking to the cloud. In this stage data flow datagram production is needed, to better understand the calls and the integrations discovered in the previous step. The data flow diagram only provides a map for analysis, without identifying what engineers should be worried about.

4. Analyze the Threats

   This stage is based upon the technology selection of the step 2, considering data type, data models, data consumption models. The purpose is to identify what kind of threats are affecting the already defined attack surface, considering how data is consumed.

5. Vulnerability Analysis

   It deals with the correlation between the system's vulnerabilities and the system's assets, considering combination between tools and best practices. Examples can be volume management, volume assessment, static/dynamic analysis etc. The key differentiator with this methodology is the focus on the risks that will have the biggest impact to the business. The scope is to identify what is wrong.

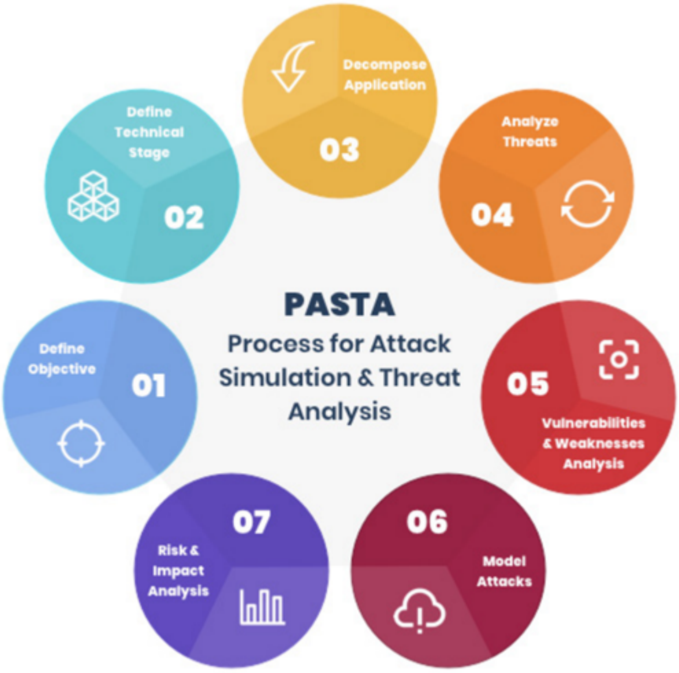


Figure 2.2: PASTA Process. [12]

6. Attack Analysis

   The purpose of this stage is to map known vulnerabilities to a node on the attack tree to determine its likelihood. Typically, the parent node is the threat objective. Attack trees can be of different dimensions, focusing on the entire system or on a small asset.

7. Risk and Impact Analysis

   The last step is related to the risk reduction, building countermeasures that mitigate the important threats. Using the information found in previous blocks the impact of threats can be identified through simulated attacks.

A schema of the steps can be found in the Figure 2.2.

STRIDE

STRIDE "evaluates the system detail design. It models the in-place system. By building data-flow diagrams, STRIDE is used to identify system entities, events, and the boundaries of the system. STRIDE applies a general set of known threats based on its name, which is a mnemonic." [Reproduced by [5]] The six categories applied are:

- Spoofing Identity

  Pretending to be something or someone you are not.

- Tampering

  Modifying something you are not supposed to modify. This can be on disk, in memory, and/or in transit.

- Repudiation

  Claiming you did not do something, whether or not you actually did.

- Information Disclosure

  Exposing information to people who are not authorized to see it.

- Denial of Service (DoS)

  Provide services to not legitimate users; this can include crashing the service, making it unusably slow, consuming all its storage (memory and/or disk).

- Elevation of Privilege

  Being able to perform operations you aren't supposed to be able to do.

[Reproduced from [13]]

The use of STRIDE is mentioned and use to build secure process of threat modeling, for example in the "Threat Modeling Process" [4], to be used to determine and rank threats.

TRIKE

This methodology is focused on the risk-management, using threat model to satisfy the security auditing process. It is based on four specific models:

1. Requirements Model

   a. Actors

   b. Assets

   c. Intended Actions

   d. Rules

   e. Actor-Asset-Action Matrix

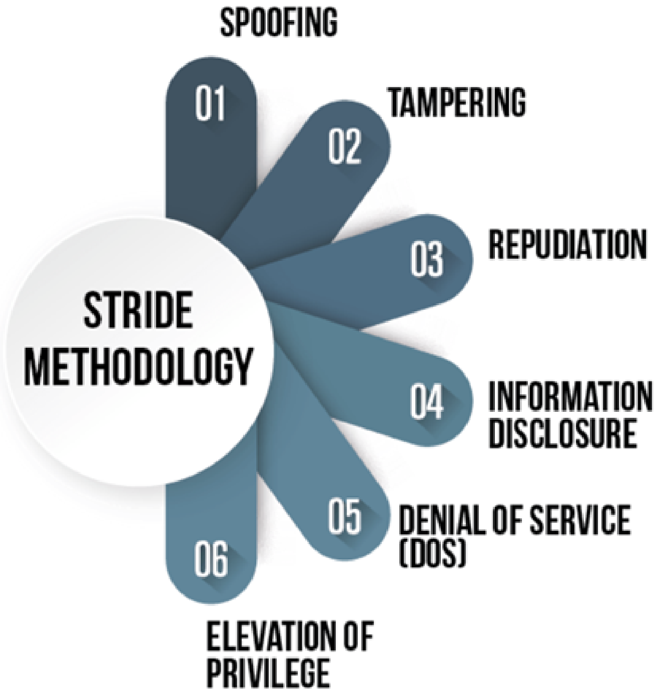


Figure 2.3: Stride methodology. [14]

2. Implementation Model
    a. Intended Actions vs Supporting Operations and the State Machine
    b. Data Flow Diagrams
    c. Use Flows
3. Threat Model
    a. Threat Generation
    b. Attacks, Attack Trees and the Attack
    c. Weakness
    d. Vulnerabilities
    e. Mitigations
    f. Attack Libraries
4. Risk Model
    a. Asset Values, Role Risks, Asset-Action Risks and Threat Exposures

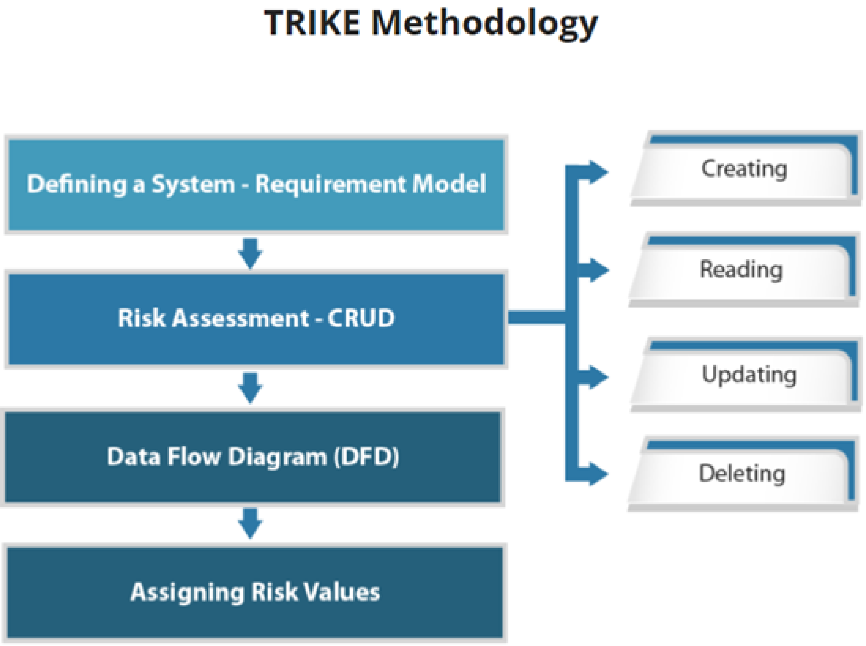


Figure 2.4: TRIKE methodology. [14]

b. Weakness Probabilities and Mitigations

c. Vulnerability Probabilities and Exposures

d. Threat Risks

e. Using the Risk Model

The requirements model establishes the stakeholder-defined "acceptable" level of risk assigned to each asset class. As described in the paper [15], while generating the threat model, it is important to make sure that all the stakeholders understand the risks that are apparent to the system and educate them in recognizing risks, threats and relative mitigations.

VAST

The Visual, Agile and Simple Threat methodology is based on ThreatModeler, a commercial automated threat-modeling platform. It is founded on the idea that modeling is only useful if it encircles the entire software development life cycle (SDLC).

It requires creating three types of models:

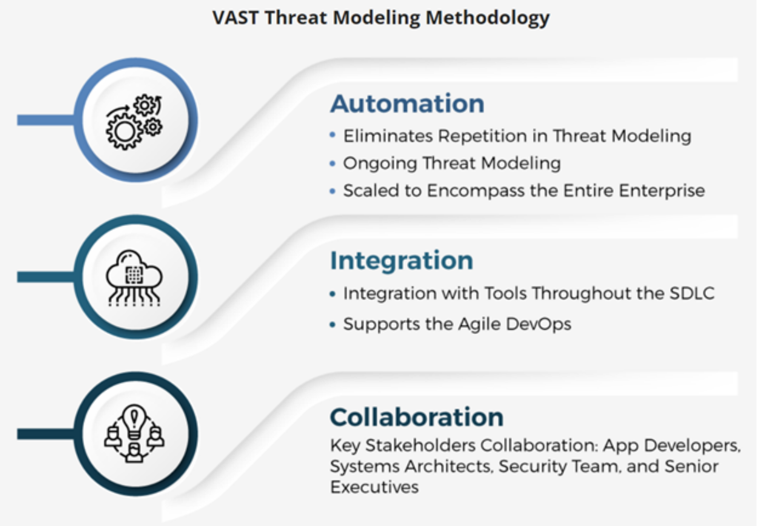

VAST Threat Modeling Methodology
Automation
• Eliminates Repetition in Threat Modeling
• Ongoing Threat Modeling
• Scaled to Encompass the Entire Enterprise
Integration
• Integration with Tools Throughout the SDLC
• Supports the Agile DevOps
Collaboration
Key Stakeholders Collaboration: App Developers, Systems Architects, Security Team, and Senior Executives

Figure 2.5: VAST methodology. [14]

1. Automation

   In order to eliminate the repetitive portion of threat modeling, taking the time needed to update a model from hours to minutes

2. Integration

   The process must integrate with the used tools throughout the SDLC to provide consistent results for evaluation. At this purpose, the methodology was created with the principles of Agile DevOps to support scalability and sustainability.

3. Collaboration

   It requires collaboration between stakeholders, software developers, system architects, security managers and senior executives throughout the organization.

   Using VAST, ThreatModeler provides a holistic view of the entire attack surface, enabling enterprises to minimize the overall risk, as described in [16].

2.1.2 Threat Modelling Tools

Based on these methodologies described above, a lot of tools are existent in the market, to perform the threat modelling process in the easiest and best way. We have identified the most important ones:

- OWASP Threat Dragon published in 2022 by OWASP
- Microsoft Threat Modeling tool, published in 2020 by Microsoft
- MITRE ATT&CK, published in 2013 by MITRE

OWASP Threat Dragon

It is a modeling tool used to create threat model diagrams as part of a secure development lifecycle, as reported in [17]. It can be used to record possible threats and decide on their mitigations, as well as giving a visual indication of the threat model components and threat surfaces. Threat Dragon runs either as a web or a desktop application and supports STRIDE and CIA.

A demo is available directly in the OWASP website.

The tool is based on the usage of elements/blocks to create a diagram, combining the workflow and the architecture diagram, representing the structure of the solution. Steps to follow are:

1. Definition of

    a. Title

    b. Owner

    c. Receiver

    d. High Level description

    d. Contributors

2. Creation of the diagram element by element. The user can give at each element the preferred name, choosing by a fixed list of categories:

    a. Actors

    b. Processes

    c. Boundaries

    d. Connections.

    Definition of flows between elements

3. Addition of threats to each element defining

    a. Title

    b. STRIDE element type

    c. Threat Status (open or mitigated)

   d. Severity
   e. Description
   f. Mitigations

4. Optional definition of "Out of scope" for each element. Some checkboxes depending on the type of the element can be set:

   a. Is a log
   b. Stores confidential data
   c. Is encrypted
   d. Is signed
   e. Is over a public network

5. Visual list of the threats and mitigations comments below the diagram

In this tool the definition of the structure, threats and mitigations of the solution are self-inserted, and in case of changes/integrations a new diagram is needed.

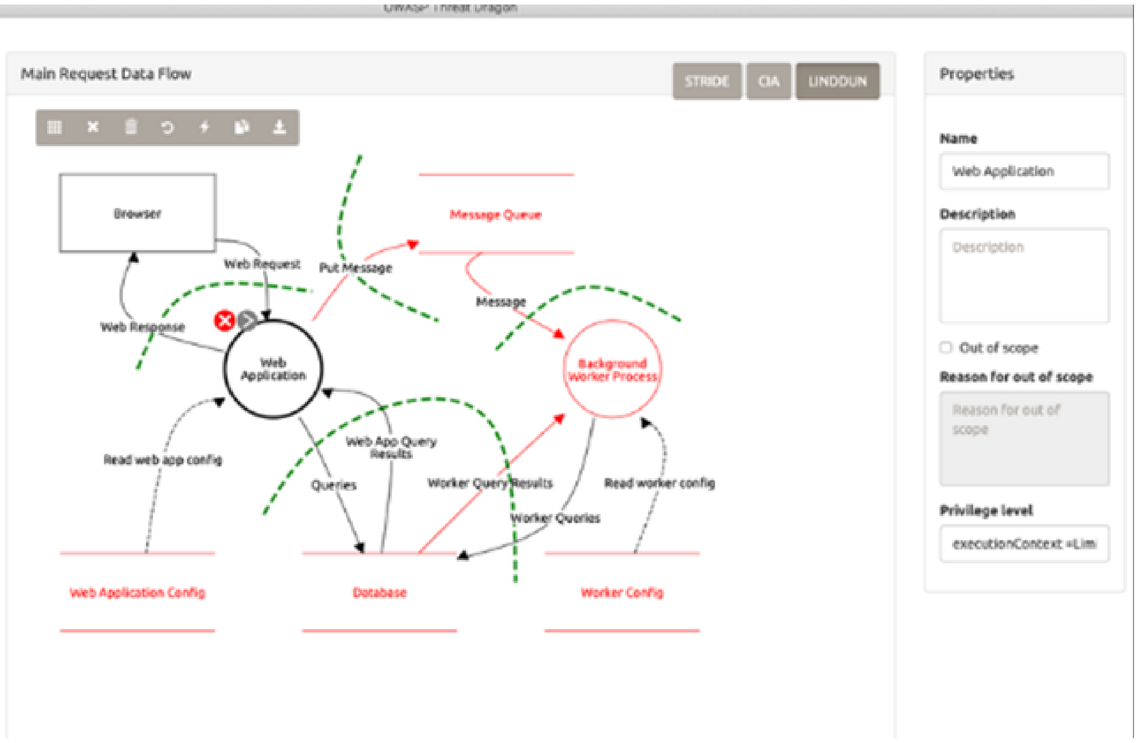
Microsoft Threat Modelling Tool

This tool is a core element of the Microsoft SDLC and allows architects to identify and mitigate potential issues as described in [19]. It enables anyone to:

- Communicate about security design of their systems
- Analyze those designs for potential security issues using a proven methodology
- Suggest and manage mitigations for security issues

As the previous Threat Dragon, the Microsoft Threat Modeling Tool is based on the creation of a diagram representing the structure of the solution, element by element, following these steps:

1. Choice of a template from a fixed list of:
   - Azure Threat Model Template
   - SDL TM Knowledge Base
   - Medical Device Template
2. Creation of the diagram, element by element, choosing from a fixed list of elements categories. Example of Generic Process category:
   - OS Process



Name: Web Application

Description: Description

□ Out of scope

Reason for out of scope: Reason for out of scope

Privilege level: executionContext =Lim

Figure 2.6: OWASP Threat Dragon tool view. [18]

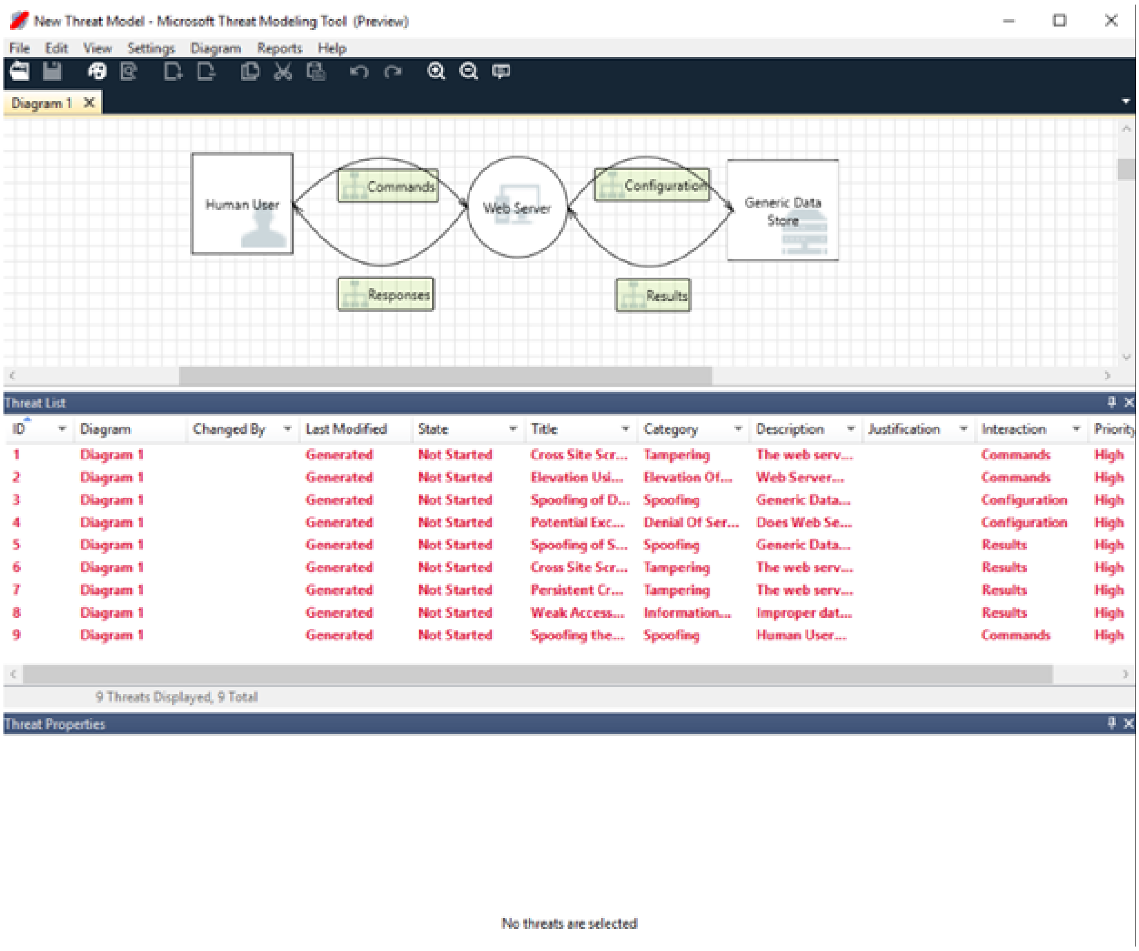
- Thread
- Kernel Thread
- Native Application
- Web Application
- Etc.

3. The tool generates a predictable list of threats based on STRIDE and proposes a generic description and suggestions to mitigate them as examples or hyperlinks to Azure documentation

4. Some properties can be chosen:

   - Not Applicable
   - Needs investigation
   - Mitigated

5. A report with threats and mitigations can be visualized



ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
1	Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Commands	High
2	Diagram 1		Generated	Not Started	Elevation Usi...	Elevation Of...	Web Server...		Commands	High
3	Diagram 1		Generated	Not Started	Spoofing of D...	Spoofing	Generic Data...		Configuration	High
4	Diagram 1		Generated	Not Started	Potential Exc...	Denial Of Ser...	Does Web Se...		Configuration	High
5	Diagram 1		Generated	Not Started	Spoofing of S...	Spoofing	Generic Data...		Results	High
6	Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Results	High
7	Diagram 1		Generated	Not Started	Persistent Cr...	Tampering	The web serv...		Results	High
8	Diagram 1		Generated	Not Started	Weak Access...	Information...	Improper dat...		Results	High
9	Diagram 1		Generated	Not Started	Spoofing the...	Spoofing	Human User...		Commands	High

9 Threats Displayed, 9 Total

Figure 2.7: MS Threat Modeling tool. [19]

Differently from OWASP Threat Dragon, in Microsoft Threat Modeling Tool the list of threats and possible mitigations are suggested by the tool itself, but the structure of the solution is still self-created.

MITRE ATT&CK

This tool is "a globally-accessible knowledge base of adversary tactics and techniques based on real-world observation" [20]. It is open and available to any person or organization.

It goes throughout different stages:

1. Choice of the matrix between predefined Enterprise, Mobile or ICS lists:

   ICS:

   - PRE
   - Windows
   - macOS
   - Linux
   - Cloud
   - Network
   - Containers

   Mobile:

   - Android
   - iOS

2. Choice of the tactic from lists depending on the matrix. Example:

   - Reconnaissance for Enterprise only
   - Initial Access for Enterprise, Mobile, ICS

3. A - Choice of a technique from a table containing name and description, based on the tactic chosen in the previous step. For example, Reconnaissance contains:

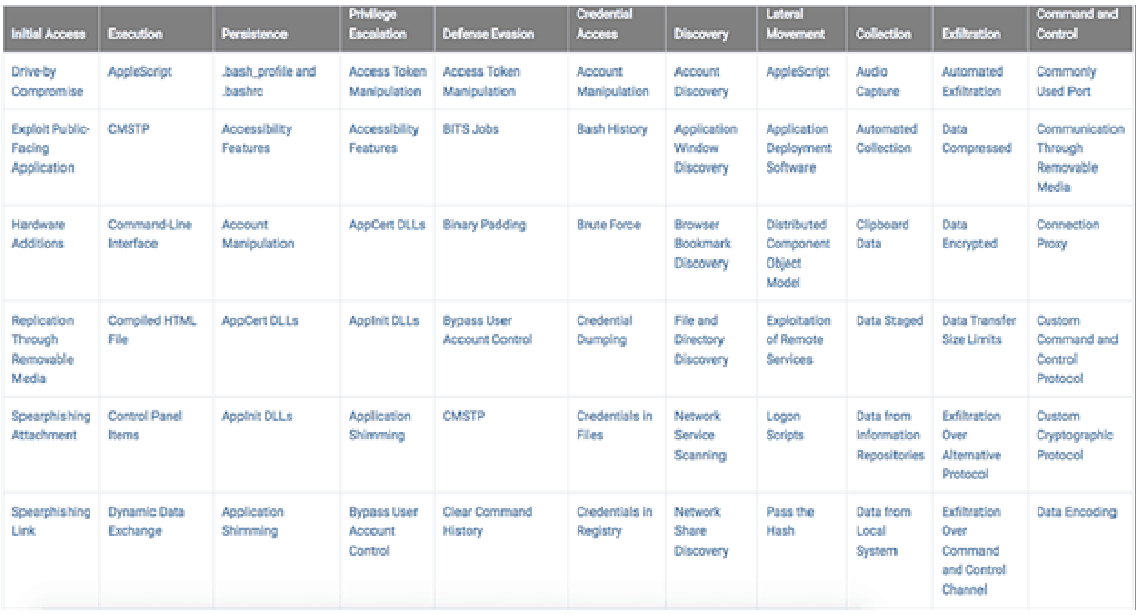
   - Active Scanning
   - Gather Victim Host Information
   - Etc.

   B - Each technique contains:

   - Name
   - Description
   - Mitigation Table
   - Detection Table

4. Choice of a sub-technique and see threat groups. For example, Active scanning contains:

   - Scanning IP blocks
   - Vulnerability Scanning
   - Wordlist Scanning



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data Staged	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Control Panel Items	AppInit DLLs	Application Shimming	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Information Repositories	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Local System	Exfiltration Over Command and Control Channel	Data Encoding

Figure 2.8: MITRE initial table. [21]

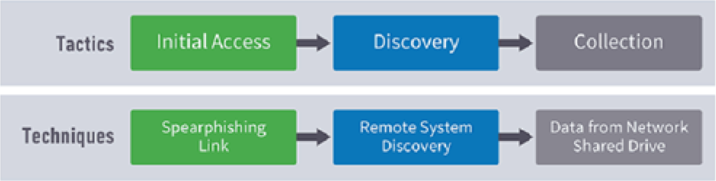


Figure 2.9: MITRE tactics and techniques. [21]

5. 5. By clicking on a threat group, the reverse procedure is put in place, and a table with threat groups and used techniques is showed.

   By clicking on a mitigation, the reverse procedure is put in place, and a table containing the mitigations and the techniques to which they can be applied, is showed.

   By clicking on detection, a table with all the data components is showed.

Differently from the previous two tools, the MITRE ATT&CK is not automatized, the user must construct the solution and the issues going through the tables until the right path is reached.

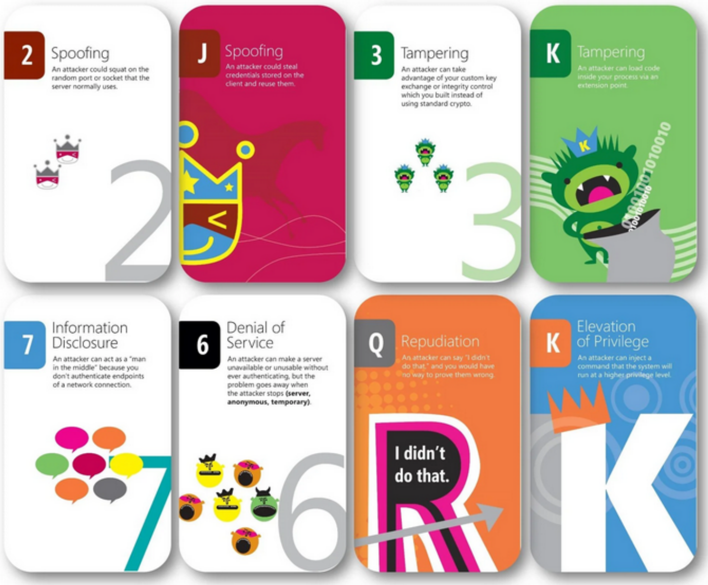

2
Spoofing
An attacker could squat on the random port or socket that the server normally uses.
J
Spoofing
An attacker could steal credentials stored on the client and reuse them.
3
Tampering
An attacker can take advantage of your custom key exchange or integrity control which you built instead of using standard crypto.
K
Tampering
An attacker can load code inside your process via an extension point.
7
Information Disclosure
An attacker can act as a "man in the middle" because you don't authenticate endpoints of a network connection.
6
Denial of Service
An attacker can make a server unavailable or unusable without ever authenticating, but the problem goes away when the attacker stops (server, anonymous, temporary).
Q
Repudiation
An attacker can say "I didn't do that," and you would have no way to prove them wrong.
I didn't do that.
K
Elevation of Privilege
An attacker can inject a command that the system will run at a higher privilege level.

Figure 2.10: Elevation of Privileges cards. [22]

EoP Threat Modeling Game

The Elevation of Privilege(EoP) "is a card game designed to introduce developers who are not information security practitioners or experts to craft of threat modeling". [22] It is a set of 74 cards, based on the six STRIDE categories, used to build a game for 3-6 players. A diagram of the solution that requires a threat model is drawn, and at every round everyone has to play one card (reading it, announcing the threat and recording it). Each round is won by the highest card, unless the Elevation of Privile card is played. According to the description in [22], "you'll use the game to find threats in a sample system architecture". Even if it cannot be a tool upon which an organization can build its risk management strategy, it can be a good opportunity to train the employees to recognize, and be aware possible threats that can harm the system.

An example of the cards is shown in the Figure 2.10.

2.1.3 Consideration

Through the analysis of the most used threat modelling methodologies, the main phases to follow in order to reach the scope of this thesis work, improving the Guided Risk Assessment tool described in 3, have been identified.

In particular, the OCTAVE methodology and the STRIDE methodology can be combined to create a consistent strategy based on the identification of the threats, of the vulnerabilities and, according to that, of the risks of a solution to be implemented, taking into consideration the STRIDE categories to identify and to group the risk effects, directly consequences of the combination of the components classified and identified.

On the other hand, however, the main focus of this technique on the threat side, with the neglect in particular of the risk calculation, bring the usage of this technique only, to be out of scope, not suitable to be considered as main basis of the tool into examination.

The analysis of the threat modelling tools, in particular, highlighted how the only consideration and usage of the *Threat Modelling* technique, is not enough to perform a complete and consistent analysis, in which the automation has a crucial role.

For this reasons, the analysis of the other techniques described below became necessary.

2.2 Risk assessment

Different definitions about risk assessment and management can be found, since it is a very important phase of the cybersecurity assessment.

As reproduced from [6] "It is normally done after the threat modeling process in order to map each threat to either a mitigation mechanism or to an assumption that is not worth worrying about in certain contexts."

We refer to the risk management as the definition provided by NIST: "Risk management is the ongoing process of identifying, assessing, and responding to risk. To manage risk, organizations should understand the likelihood that an event will occur and the potential resulting impacts. With this information, organizations can determine the acceptable level of risk for achieving their organizational objectives and can express this as their risk tolerance."

(Reproduced from [23]

A lot of risk assessment frameworks exist in the market, in order to lead organizations to manage the risk appropriately.

We decided to focus on some of them, classified in sources as [24]:

- NIST Risk Management Framework, published in 2021 by NIST
- ISO 27005, published in 2018 by ISO
- FAIR, published in 2010 by OPENGROUP
- CIS CONTROLS V8, published in 2022 by CIS
- IRAM2, published in 2014 by ISF

2.2.1 Risk Assessment Methodologies

NIST Risk Management Framework

It "provides a process that integrates security, privacy and cyber supply chain risk management activities into the SDLC" (Reproduced from [23]).

It is structured in seven stages, as shown in 2.11:

1. Prepare: activities that are crucial to prepare the organization to manage security

2. Categorize: categorization of the system and information stored, processed and transmitted based on an impact analysis

3. Select: selection of the controls taken from NIST SP 800-53 to protect the system

4. Implement: implementation of the controls and documentation of deployment of them

5. Assess: assessment to determine if the controls are in place, producing the right result

6. Authorize: risk-based decision to authorize the system (to operate)

7. Monitor: monitorization continuously of the control implementation and risks to the system

ISO 27005

This framework follows an iterative process for risk assessment and/or treatment activities. An iterative approach can lead to a depth increasement providing a good balance between time and effort spent in identifying appropriately controls, as described in the official documentation [26].

It divides the risk assessment into two different risk management cycles 2.12:

1. Strategic Cycle: conducted at longer time basis or in case of changes and applies to the environment

2. Operational Cycle: conducted in a shorter time basis and applies to all risk assessments considering the context of the process

At the end of the document of the framework, tables with examples of attack methods, target objectives, threats, vulnerabilities, risk scenarios etc. are given.

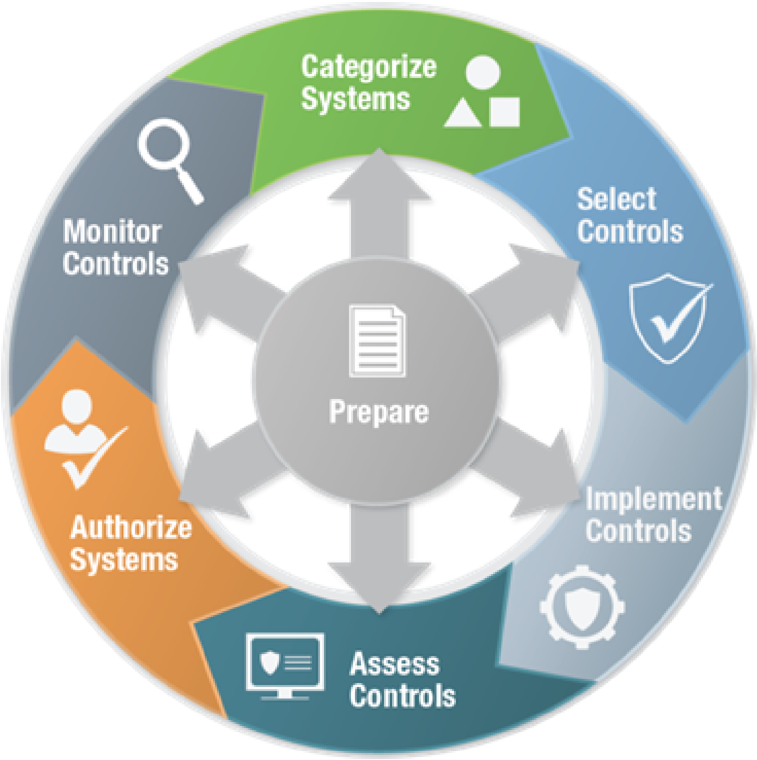


Figure 2.11: NIST Risk Management Framework Process. [25]

FAIR

The Factor Analysis of Information Risk framework [27], based on ISO 27002, provides standards and best practices to help organization management, measurement and reporting following a business perspective. It is structured in 4 stages 2.13:

1. Identify Scenario Components
2. Estimate Loss Event Frequency
3. Evaluate Probable Loss Magnitude
4. Derive and Articulate Risk

CIS Controls v8

More than a framework it is a set of safeguards to mitigate the most common attacks against systems and networks. It is structured in four objectives followed by different principles 2.14:

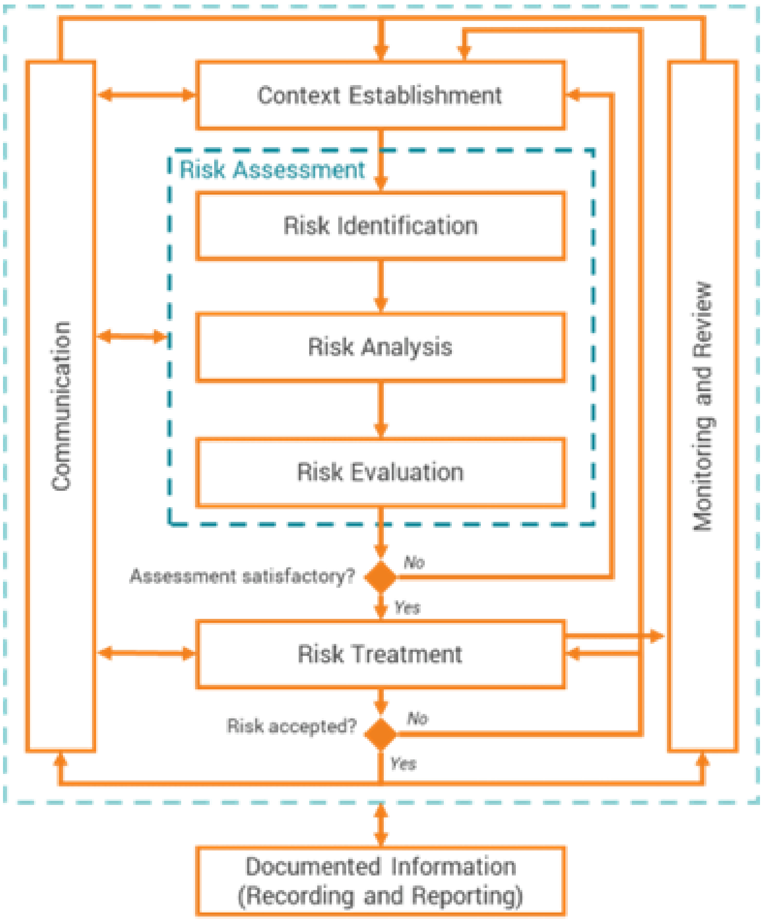


Figure 2.12: ISO 27005 Process. [26]

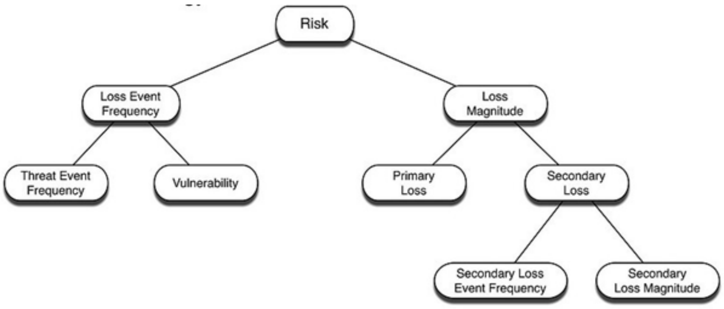


Figure 2.13: Fair Process. [27]

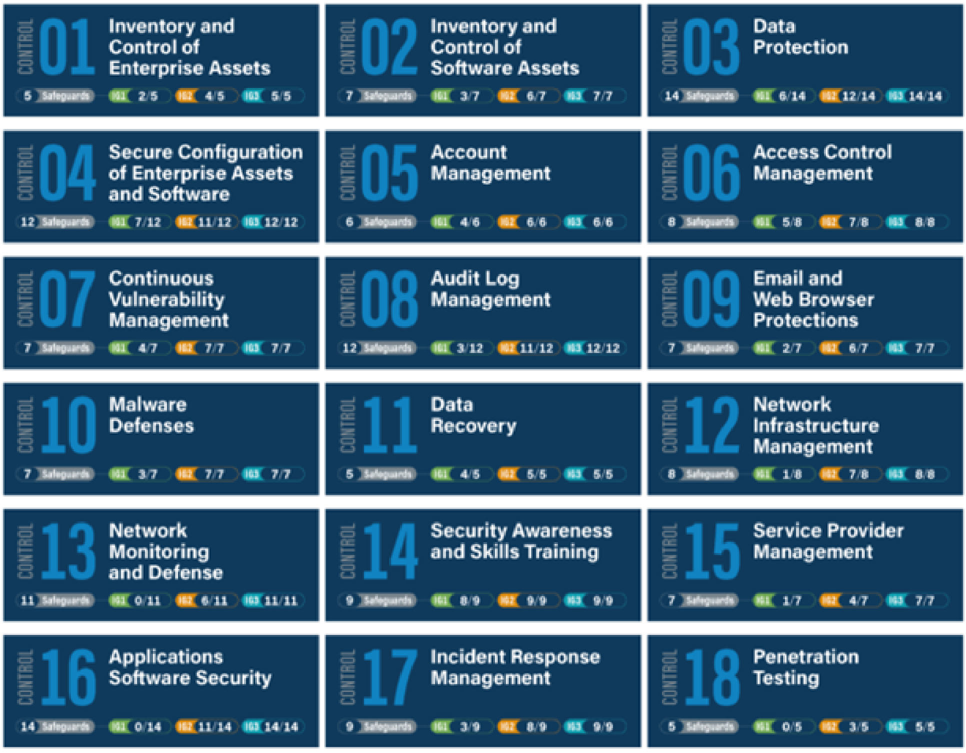
A. Managing security risk
   - Governance
   - Risk Management
   - Asset Management
   - Supply Chain
B. Protect against cyber-attacks
   - Service Protection Policies and Processes
   - Identity and Access Control
   - Data Security
   - System Security
   - Resilient Networks and Systems
   - Staff Awareness and Training
C. Detecting cyber security events
   - Security Monitoring
   - Proactive Security Event Discovery
D. Minimising the impact of cyber security incidents
   - Response and Recovery Planning
   - Lessons Learned

Tables containing the controls, and the relative tool are open source. The methodology is used in real world, and an explanation of the different case studies is given by [28].

IRAM2

This framework is structured into six different steps, described in the documentation [30] and shown 2.15:

- Scoping
  - Understand the environment to be assessed and the organization characteristics
  - Define the purpose of the environment to be assessed
- Business Impact Assessment



CONTROL 01 Inventory and Control of Enterprise Assets
5 Safeguards IG1 2/5 IG2 4/5 IG3 5/5

CONTROL 02 Inventory and Control of Software Assets
7 Safeguards IG1 3/7 IG2 6/7 IG3 7/7

CONTROL 03 Data Protection
14 Safeguards IG1 6/14 IG2 12/14 IG3 14/14

CONTROL 04 Secure Configuration of Enterprise Assets and Software
12 Safeguards IG1 7/12 IG2 11/12 IG3 12/12

CONTROL 05 Account Management
6 Safeguards IG1 4/6 IG2 6/6 IG3 6/6

CONTROL 06 Access Control Management
8 Safeguards IG1 5/8 IG2 7/8 IG3 8/8

CONTROL 07 Continuous Vulnerability Management
7 Safeguards IG1 4/7 IG2 7/7 IG3 7/7

CONTROL 08 Audit Log Management
12 Safeguards IG1 3/12 IG2 11/12 IG3 12/12

CONTROL 09 Email and Web Browser Protections
7 Safeguards IG1 2/7 IG2 6/7 IG3 7/7

CONTROL 10 Malware Defenses
7 Safeguards IG1 3/7 IG2 7/7 IG3 7/7

CONTROL 11 Data Recovery
5 Safeguards IG1 4/5 IG2 5/5 IG3 5/5

CONTROL 12 Network Infrastructure Management
8 Safeguards IG1 1/8 IG2 7/8 IG3 8/8

CONTROL 13 Network Monitoring and Defense
11 Safeguards IG1 0/11 IG2 6/11 IG3 11/11

CONTROL 14 Security Awareness and Skills Training
9 Safeguards IG1 8/9 IG2 9/9 IG3 9/9

CONTROL 15 Service Provider Management
7 Safeguards IG1 1/7 IG2 4/7 IG3 7/7

CONTROL 16 Applications Software Security
14 Safeguards IG1 0/14 IG2 11/14 IG3 14/14

CONTROL 17 Incident Response Management
9 Safeguards IG1 3/9 IG2 8/9 IG3 9/9

CONTROL 18 Penetration Testing
5 Safeguards IG1 0/5 IG2 3/5 IG3 5/5

Figure 2.14: CIS Controls v8 Process. [29]

  - Identify the information asset and assess the business impact

- Threat Profiling
  - Identify the relevant threats and prioritise them
  - Identify the ways that the highest priority threats can manifest to cause damages to the environment

- Vulnerability Assessment
  - Identify the controls that can be put in place and are relevant for the purpose of the environment
  - Identify the extent to which every control has been implemented
  - Understand the strength of the controls put in place

- Risk Evaluation
  - Calculate the residual risk rating for each risk

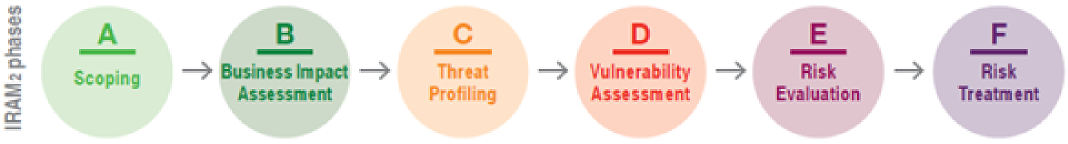


Figure 2.15: IRAM2 Process. [30]

- Risk Treatment
  - Define a risk treatment approach for each risk

2.2.2 Consideration

If the analysis of the *Threat Modelling methodologies* allowed to plan a strategy in the identification of the phases to follow, the study of the *Risk Assessment methodologies* helped in reaching the final goal of the result to be provided by the tool through the focus on the risk identification and calculation. What these methodologies highlight, in fact, is the definition of the risk as the combination between a threat and a vulnerability, defining the scores and a classification, that allows the identification of the problems in the process followed by the tool.

The failed research of risk assessment tools already implemented shown that these technique is more based on the classification and the identification of the general formulas and scores for the calculation of the risk, and for this reason the greatest contribution of these techniques to this thesis work is done by the classification of the components identified.

The vulnerabilities tables listed in the IRAM2 and in ISO 27005, in fact, helped me in the identification and categorization of the vulnerabilities, together with the CIS v8, from which the major vulnerabilities have been extracted, starting from the controls and the mitigations.

Not only in the vulnerability perspective these techniques resulted crucial in obtaining the main improvements of the tool in examination, but also in the definition of the threats and the threats groups, for which the IRAM2 classification has been choosen as major source.

Since the techniques of *Threat Modelling* and *Risk Assessment*, were taken into consideration, bringing to light the crucial phase of *Vulnerability Assessment*, also an analysis of this technique, and consequent research of possible tools already implemented, became necessary.

2.3 Vulnerability assessment

As we have seen above, the threat modelling require to have a look outside the organization to determine threats already existing that could lead to organizational damages, while vulnerability assessment looks inside the organization for structural flaws and weaknesses.

Through the analysis of the *Threat Modelling* and *Risk Assessment* techniques, the identification of the main phases focused on threats, vulnerabilities, and consequent risk, have been highlighted as crucial during this process.

For that reason, an analysis on the *Vulnerability Assessment* has been performed.

If the methodologies listed above present consistent frameworks describing the strategy to be followed, during the research became clear that if the general concept of vulnerability assessment can be useful in this work perspective, the real implementations already present in the market, through automatic scanning, is out of the scope.

The definition of *Vulnerability Assessment*, in fact, is "Formal description and evaluation of the vulnerabilities in an information system." (Reproduced by [2]), phase that is followed in the process into examination, but since the main implementations are performed through the usage of automatic scanners (e.g., Nessus as described in [31]), and the purpose of this work is a process automatized in the calculation of the risk, but manually in identifying the assets, the applications of this techniques has been considered as out of scope.

2.4 Final Consideration

Through the analysis of the main techniques of the *Threat Modelling*, *Risk Assessment* and *Vulnerability Assessment*, the possible application of the methodologies to reach the goal of the tool into examination has been considered.

If on one hand the *Threat Modelling* technique could be considered at the beginning as more relevant for the purpose of the tool, the study of the methodologies and the tools already published brought to light a main focus on the threats side, with a neglect of the risk calculation, which made the analysis of the other techniques necessary.

On the other hand, the *Vulnerability Assessment* technique has been considered out of scope because of the main focus of the vulnerability side only, and the main implementations through automated scanners.

At the same time, the *Risk Assessment* analysis highlighted how the phases of threats and vulnerabilities identification can be combined, without neglecting the calculation of the risk, establishing a complete process in which the different areas have been considered to create a consistent and complete analysis that culminates in the identification of the threats that, combined with exploitable vulnerabilities on an existent asset, create a concrete possible risk.

Since the phases of the threats and vulnerabilities identification are crucial parts of the risk assessment process, important insights can be used and taken into consideration from the two techniques, even if they were considered mainly out of scope in the strategy definition. If the STRIDE methodology, for example, is not used to model and categorize the threats, and it is not used as base of the strategy, it can be used to categorize the risk effects, in order to combine different techniques to obtain a complete and consistent result, which provides an identification of threats and vulnerabilities through pre-defined lists as suitable as possible to the organization in which the solution has to be implemented, but providing automation in the identification of the relevant vulnerabilities for the solution into examination, and a consequent calculation of the risk as accurate as possible.

Chapter 3

The Guided Risk Assessment tool

During this chapter we will go through the meaning of the concepts described in the second chapter, applying them specifically to Hilti organization.

Following the perspective of caring about the risk management, the Guided Risk Assessment tool was implemented in May 2018 by Hilti, in order to support both project managers and IT risk & Cybersecurity team in assessing the risk of the solutions, preventing negative impacts for the company.

It was created to be a tool for the purpose of self-assessment of the risk of a solution that has to be developed/implemented/integrated, and it was conceived to be used during the entire development lifecycle.

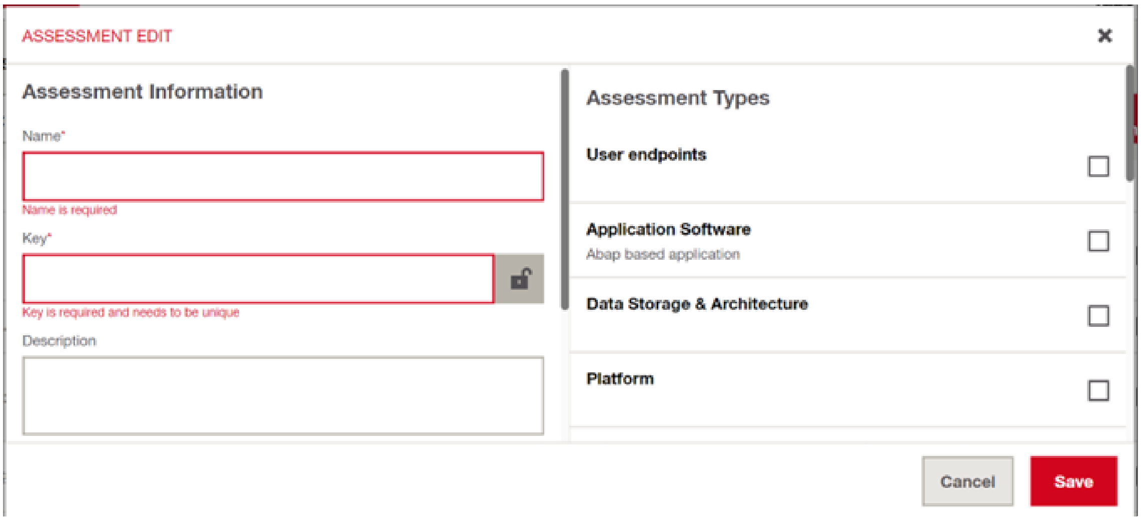
It is focused on four different principles:

1. Identification : What can go wrong?
2. Prioritization : What would be the damage and how likely would it happen?
3. Mitigation : How can we reduce the risks?
4. Communication & Acceptance : How can we communicate risks and ensure go-live?

3.1 GRA overview

3.1.1 How to perform an analysis

The "Assessment Life-cycle" begins when a solution need to be implemented, and the new assessment is created by a user with an *Admin* role only, gained by members of the Cybersecurity team.



ASSESSMENT EDIT

Assessment Information

Name*

Name is required

Key*

Key is required and needs to be unique

Description

Assessment Types

User endpoints □

Application Software □
Abap based application

Data Storage & Architecture □

Platform □

Figure 3.1: Creation of an assessment.

The admin defines:

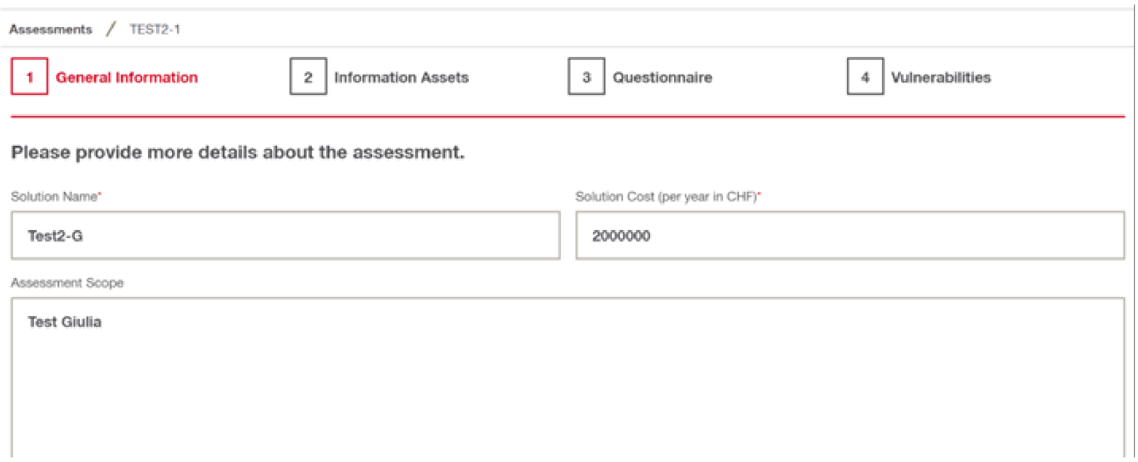
- Title: every assessment has a title, used automatically by the tool itself to define a unique ID;
- Editor(s): every user defined with this role for this specific assessment, can access the assessment from the list in the HomePage, using the search bar and/or the filters;
- Assessment Types: through a checkbox list, the admin can select one or more types the solution to be implemented belongs to (e.g., Platform, Application Software etc.), identified for that specific solution. The Assessment Types cannot be modified after the assessment is created.

You can find an overview of the creation pop-up in the Figure 3.1.

After the new assessment is created, the user has to provide more information about the solution, in order to give the tool all the elements needed to perform an analysis, through three different steps, plus a fourth one in which the result of the analysis is provided by the tool. In order to come to the end of the "Assessment Life-cycle", the user has to take into consideration the result of the analysis, and put in place the mitigations needed to make the risk as low as possible.

Going into details of each step to be performed:

1. General Information: the first step of the analysis requires the user to provide the general information about the solution, as the *Assessment Scope* and the



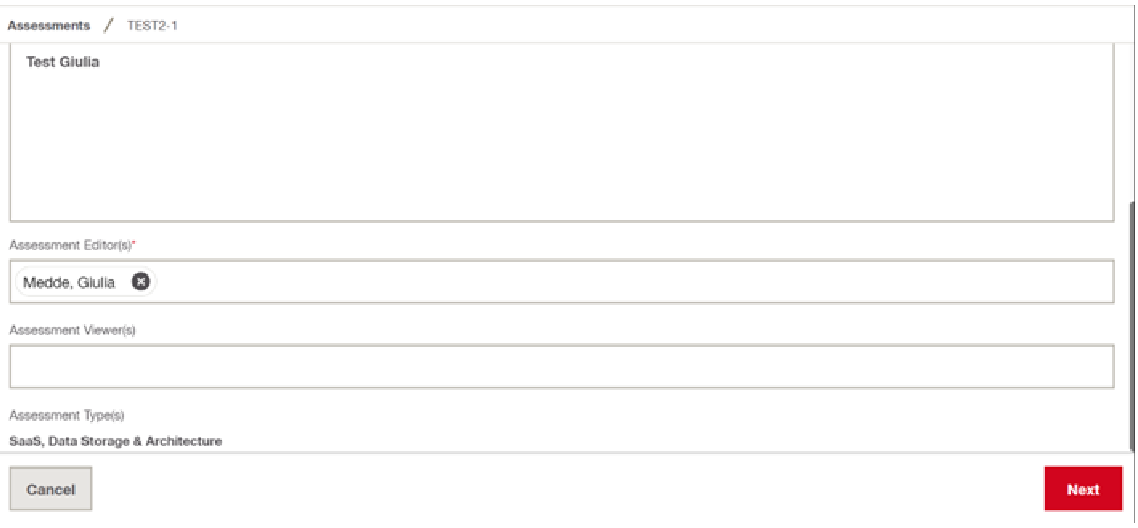
Please provide more details about the assessment.

Solution Name*: Test2-G

Solution Cost (per year in CHF)*: 2000000

Assessment Scope: Test Giulia

Figure 3.2: Creation of an assessment - Step 1.



Test Giulia

Assessment Editor(s)*: Medde, Giulia

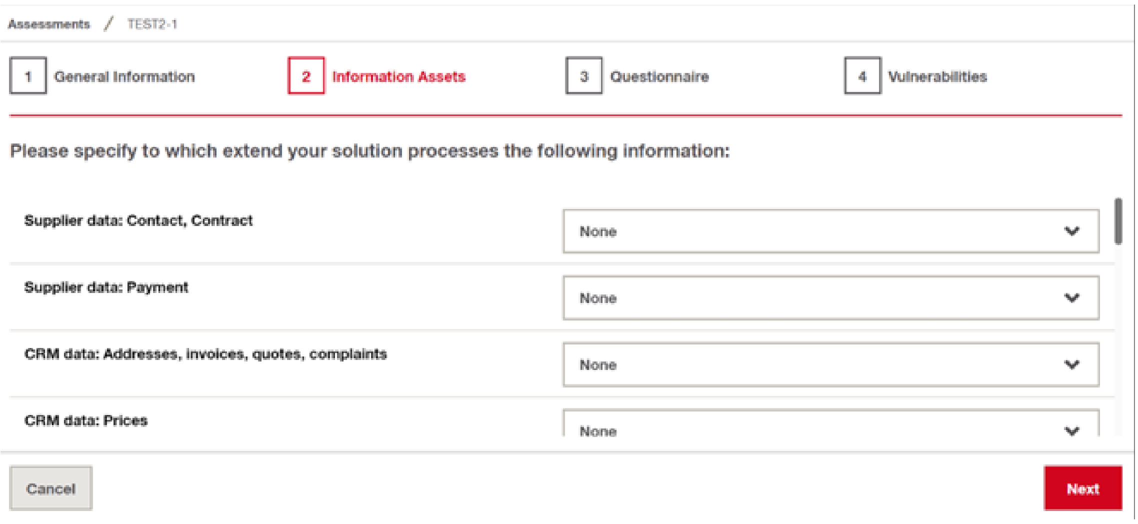
Assessment Viewer(s):

Assessment Type(s): SaaS, Data Storage & Architecture

Figure 3.3: Creation of an assessment - Step 1.

Solution Cost (per year, in CHF). This last field is mandatory, and it will be used in the calculation of the risk. An example of the *Step 1* is shown in the Figures 3.2 and 3.3

2. Information Assets: the second step of the analysis requires the user to provide information about the data stored/processed/transferred within the solution. A total of 18 questions are provided, each one with a multiple-choice answer dealing with the rights given by the solution to that specific category of data.



Please specify to which extend your solution processes the following information:

Supplier data: Contact, Contract — None

Supplier data: Payment — None

CRM data: Addresses, invoices, quotes, complaints — None

CRM data: Prices — None

Figure 3.4: Creation of an assessment - Step 2.

The possible answers are *None*, *Read* and *Read & Write.* The questions cover different kind of data, sensitive and/or public. An example of a Step 2 is shown in the Figure 3.4

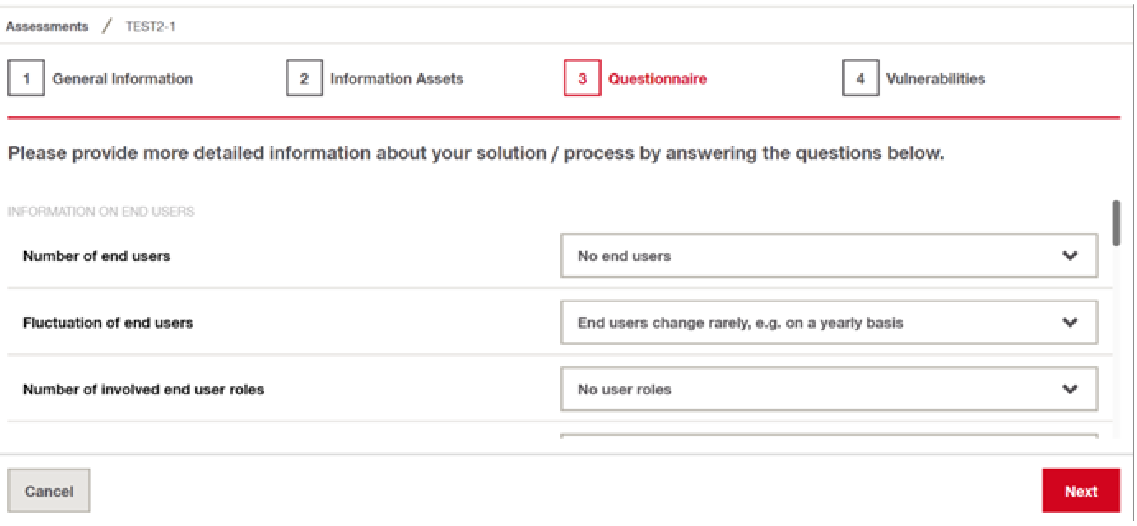
3. Questionnaire: this third step of the analysis requires the user to provide information about different aspects of the solution. A total of 17 questions are grouped into 4 categories (e.g., Information of end users, Information on administrators and vendors, Information on system access and usage, Information on technological and operational change).

   Each question has a multiple-choice answers, and it will be used during the calculation. An example of this *Step 3* is shown in the Figure 3.5.

4. Vulnerabilities: this fourth and last step contains the recap of the information provided in the *Step 1* (e.g, *Solution Cost* etc.), and the result of the analysis by the tool itself. As you can notice in Figure 3.6, on the left side of the page a heatmap (a cartesian diagram Likelihood x Impact), provides the position of the risk scenarios associated to the solution, through bubbles in different areas of the map.

   The color of the areas inside of the map represent the level of the risk, mapping the *High Risk* with the red color, the *Middle Risk* with yellow, and *Low Risk* with the green color.

   Every bubble is clickable, in order to provide a name and a description of the risk scenario that it represents.

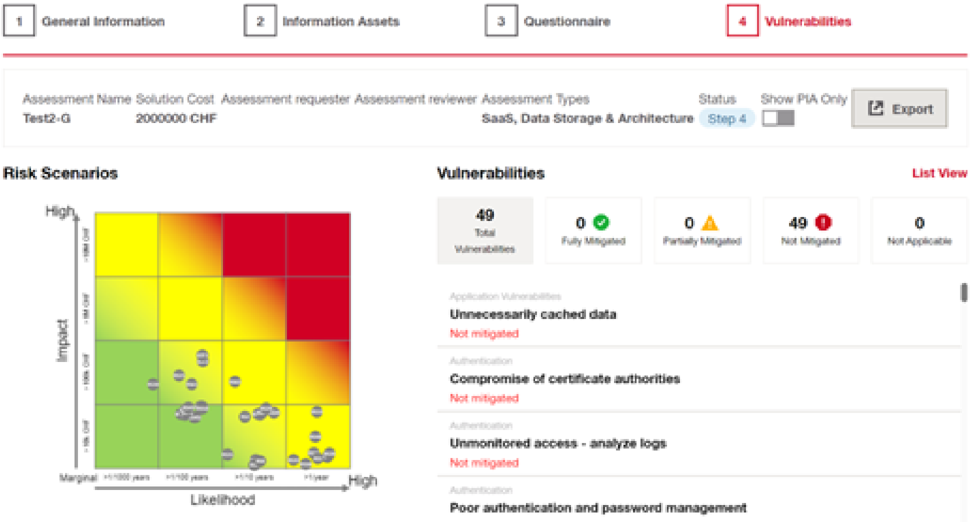


Please provide more detailed information about your solution / process by answering the questions below.

INFORMATION ON END USERS

Number of end users	No end users
Fluctuation of end users	End users change rarely, e.g. on a yearly basis
Number of involved end user roles	No user roles

Figure 3.5: Creation of an assessment - Step 3.



Assessment Name	Solution Cost	Assessment requester	Assessment reviewer	Assessment Types	Status	Show PIA Only
Test2-G	2000000 CHF			SaaS, Data Storage & Architecture	Step 4	

49	0	0	49	0
Total Vulnerabilities	Fully Mitigated	Partially Mitigated	Not Mitigated	Not Applicable

Application Vulnerabilities
Unnecessarily cached data
Not mitigated

Authentication
Compromise of certificate authorities
Not mitigated

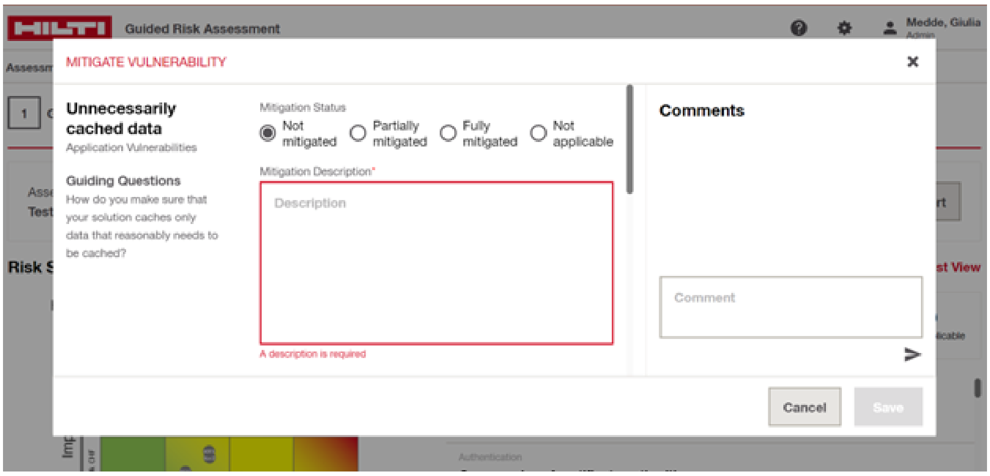
Authentication
Unmonitored access - analyze logs
Not mitigated

Authentication
Poor authentication and password management

Figure 3.6: Creation of an assessment - Step 4.

On the right side of the page, a list of vulnerabilities identified for that solution is provided, with a *Not Mitigated* status for all of them at the beginning. The vulnerabilities can be filtered by the status, *Not Mitigated, Partially Mitigated, Fully Mitigated*, and are clickable, in order to provide the details.

As shown in Figure 3.7 and 3.8, by clicking on a single vulnerability, a pop-up appears, providing the *Guiding Questions*, useful to understand which



MITIGATE VULNERABILITY ×

Unnecessarily cached data

Application Vulnerabilities

Guiding Questions

How do you make sure that your solution caches only data that reasonably needs to be cached?

Mitigation Status

● Not mitigated ○ Partially mitigated ○ Fully mitigated ○ Not applicable

Mitigation Description*

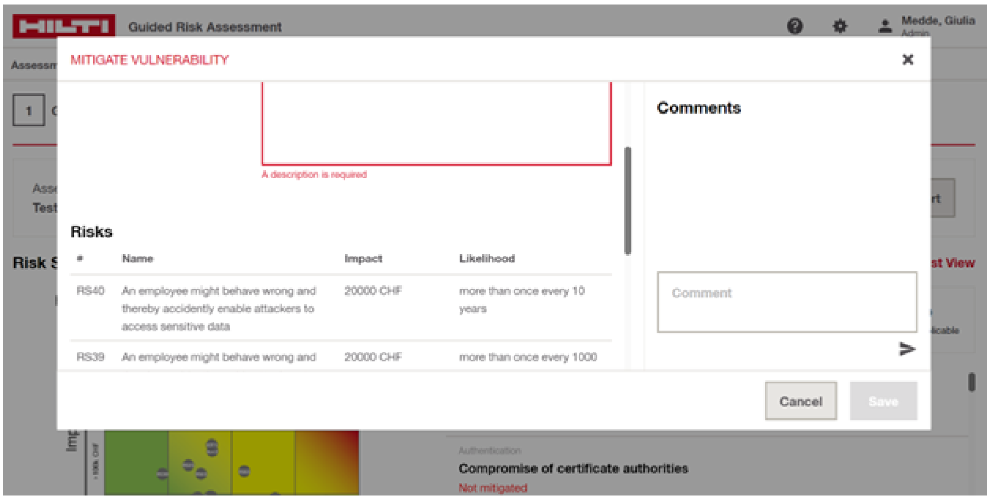
Description

A description is required

Comments

Comment

Figure 3.7: View of vulnerabilities in Step 4.



MITIGATE VULNERABILITY ×

A description is required

Comments

Risks

#	Name	Impact	Likelihood
RS40	An employee might behave wrong and thereby accidently enable attackers to access sensitive data	20000 CHF	more than once every 10 years
RS39	An employee might behave wrong and	20000 CHF	more than once every 1000

Comment

Authentication

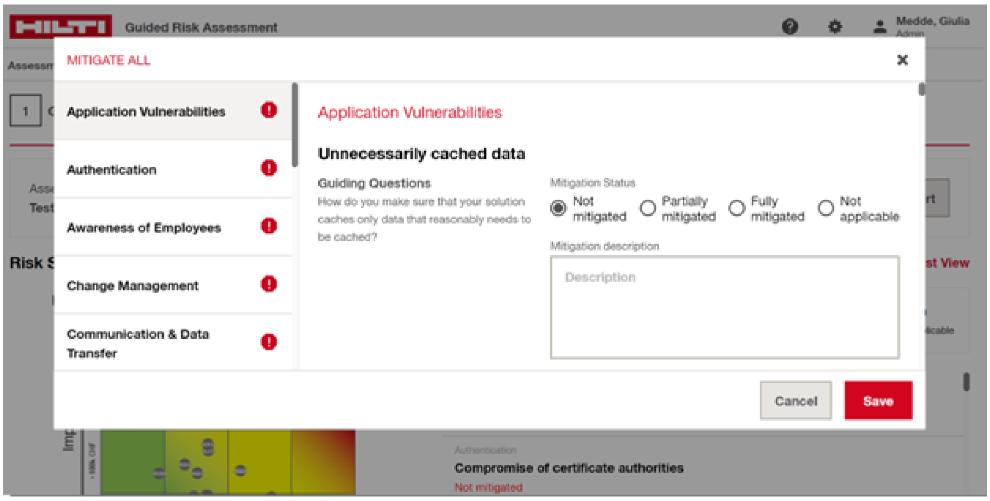
Compromise of certificate authorities

Not mitigated

Figure 3.8: View of each vulnerability in Step 4.

mitigation(s) should be applied before change the status of the vulnerability from "Not Mitigated" to "Partially" or "Fully Mitigated". The section *Mitigation Description* is necessary to describe the mitigation(s) put in place. On the right section, comments can be added, while on the bottom, a list of risk scenarios associated to that specific vulnerability are provided. The collection of the risk scenarios associated to the vulnerabilities is the one shown in the heatmap mentioned above.

Coming back to the main page of the *Step 4* of the analysis, another button



Application Vulnerabilities

Unnecessarily cached data

Guiding Questions

How do you make sure that your solution caches only data that reasonably needs to be cached?

Mitigation Status

● Not mitigated ○ Partially mitigated ○ Fully mitigated ○ Not applicable

Mitigation description

Description

Authentication

Compromise of certificate authorities

Not mitigated

Figure 3.9: List view of each vulnerability in Step 4.

List View is clickable. As you can see in the Figure 3.9, it opens a pop-up with all the vulnerabilities already listed, but grouped into categories, with an indication of the status, the related guided question(s), and the section where describe the mitigation(s) put in place.

Through the *Show PIA only* button, is also possible filter the vulnerabilities, visualizing only the ones PIA related (e.g., *Integrity* etc.), while the export is possible via PDF Report ot Audit Log, by clicking on the button *Export* on the right side.

The purpose of this step is to identify the vulnerabilities related to the solutions, visualizing the risk scenarios in the heatmap, and to take into consideration each of them, putting in place a mitigation (documented in the relative section).

When a mitigation is applied, the user can change the status of the related vulnerability, and the bubble representing the related risk scenario will change the position, giving the user an overview of the risky level of the solution.

A vulnerability can also be set as "Not Applicable" status, when it is not relevant for the solution.

Through this steps, the user can be aware about the vulnerabilities, the risk scenarios, and the required changes, to make the solution as secure as possible.

At the end of the process, the user can click on the *Request Review* button, on the bottom panel, and the status of the assessment will change to *Pending Review*, since the assessment need to be reviewed by the Cybersecurity team, before the solution can obtain the approval and be implemented.

3.1.2 Comparison with other tools

After the overview of the Guided Risk Assessment tool, the main differences between the Hilti internal tool, and the tools presented in the market (described here, 2.1.2) should be clearer.

1. Solution definition On one side, the other tools require to build a solution diagram or to navigate through the different tables to build the solution, while in the Hilti tool the solution is built choosing from a fixed list of types and answering to a set of questions.

2. Vulnerability definition As threat modelling tool, the other tools don't provide a set of weaknesses of the solution, while the GRA automatically provides, according to the solution deliverable type, a set of vulnerabilities to be mitigated.

3. Risk calculation For the same reason that the other tools don't provide a set of vulnerabilities, they don't even calculate the associated risk.

   On the other side, the heatmap inserted in the GRA, gives an overview of the risk scenarios according to the elements inserted in the questionnaires, elements that is not present in the other tools.

4. Mitigation suggestions As mentioned above, the MITRE ATT&CK is the only tool in which the suggestions are provided directly by the tool, according to the used techniques.

   Since in the GRA the mitigation suggestions are directly connected to the related vulnerability, and provided directly by the tool, we can say that in that sense the tool is similar to the MITRE one.

As conclusion of this comparison, we can notice how the GRA tool is different from the previous ones since it is used not only as threat modelling tool, but also at the scope of calculating the risk and identifying the vulnerabilities with relative mitigations, provided automatically by the tool.

Being these tools, in fact, application of the *Threat Modelling* technique, don't provide a calculation of the risk, nor a list of vulnerabilities and possible mitigations. If on one side, the *OWASP Threat Dragon* can be taken into consideration for the usage of the *STRIDE* categories for the threat categorization, in all the other aspects is completely to be considered not enough automatized. The Microsoft Threat Modeling Tool, indeed, even if is similar to the previous one, has the characteristic that generates a predictable list of threats, with a level up in the automation. Despite this, is still not enough automatic, and not enough focused on the risk and vulnerabilities side. On the contrary, the MITRE ATT&CK provides an analysis not only based on the threats, but also on the mitigation suggestion, useful to be considered in the scope of the GRA. At the same time, however, even in this case the tool is completely manual (the user has to navigate inside of the tables provided by the application), and the risk calculation is not considered at all.

Through this comparison, is even more clear how the application of the *Threat Modelling* technique only results out of the scope of the tool into examination, and a combination between the different techniques is needed in order to build a complete, consistent and robust strategy to provide an analysis focused not only on the threat side, but also on the vulnerability and the risk side.

If on one side, in fact, the *Threat Modelling* technique and tools, can be useful in identifying the threats, and the *Vulnerability Assessment* can be considered in its general meaning to be focused in identifying the weaknesses of the system, on the other side the *Risk Assessment* is crucial for the combination between the two components and the real calculation of the risk, the main goal to be achieved with this tool.

3.1.3 GRA components and connections

Before going into details of the calculation of the risk, is important to understand the connections between the components involved. Components involved are:

- Assessment: the assessment is the single analysis performed on a solution. Its lifecycle starts with the creation by an admin, and ends (potentially) with the mitigations of all the vulnerabilities, which determines the final calculation of the risk;

- Types: they correspond to the types which the solution belongs to (e.g., Platform, SaaS, IoT etc.) and are pre-defined.

- Questions: it is the set of questions provided to the user in the Step 3 *Questionnaire* of the analysis in order to collect more information about the solution.

- Options: is the set of the possible answers of each question in the *Questionnaire*. Every answer is characterized by a different weight, assigned during the option definition.

- Vulnerability: they are pre-defined and associated to the solution through the choice of the types. They are shown to the user in the Step 4 *Vulnerabilities* of the analysis, and have to be mitigated providing a description of the mitigation put in place.

- Threat Groups: they are pre-defined, and never shown to the user, but crucial for the calculation of the likelihood.
- Risk Scenarios: they are represented through the bubbles shown in the heatmap and are strictly related with the likelihood concept.
- Risk Effect: general consequence of a risk scenario situation. They are pre-defined, and crucial for the impact calculation.

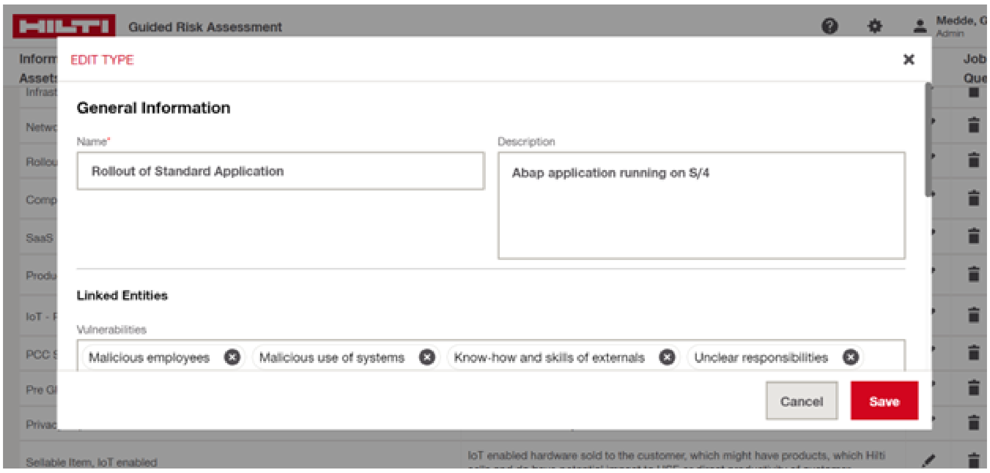
Considering all these components described above, they are connected to each other in order to be used during the risk calculation. Since the reasons behind the connections are strictly related with the concepts of likelihood and impact calculation, for a better explanation refer to 3.2.

3.1.4 GRA lists

The description of the usage of the tool, and the comparison with other tools in the market, have highlighted the importance of the components and the connections between them. In user perspective, in fact, the accuracy of the lists lead to a better experience, in terms of usability, but also in terms of usefullness, giving to the user the possibility to really understand the weaknesses of the solution to be implemented and the best way to solve them, building secure and robust systems in Cybersecurity perspective. Have pre-defined lists of components adapted as much as possible to the needs of the solution into analysis, improves not only the user experience, but highlights the vulnerabilities of the system in the easiest way, giving the opportunity to make the mitigations as clear as possible. For this reason, and being the pre-defined lists crucial part of the risk calculation, as we will see in the next paragraph, a lot of effort was spent during this phase to study and analyse deeply the content and the structure of the tables, to find the best way in which improve them in a user perspective, but also focusing on the principle of build solutions as secure as possible.

- Solution Deliverable Type list

  In the current GRA 15 different solution deliverable types are defined, each one characterized by the "Name", "Description", and a list of connected vulnerabilities. Since the types for the solution are defined by the admin during the creation of the assessment, the list of the vulnerabilities related to that solution is defined according to the associated types. The list of the types can be handled by an admin in the relative section of the *Admin Panel.* As shown in Figure 3.10, each type is editable.



EDIT TYPE

General Information

Name*: Rollout of Standard Application

Description: Abap application running on S/4

Linked Entities

Vulnerabilities: Malicious employees ⊗ Malicious use of systems ⊗ Know-how and skills of externals ⊗ Unclear responsibilities ⊗

Sellable Item, IoT enabled

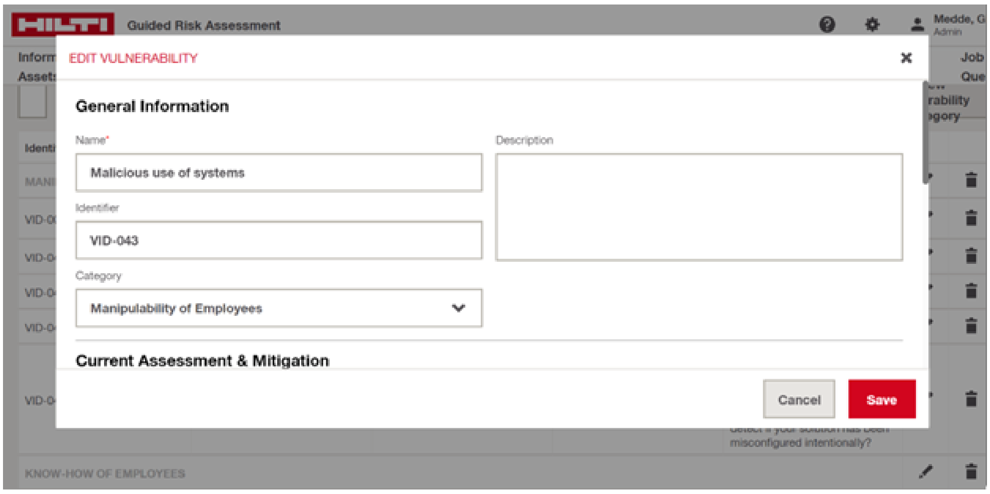
IoT enabled hardware sold to the customer, which might have products, which Hilti

Figure 3.10: Edit a Solution Deliverable Type.

- Vulnerabilities list

  Vulnerabilities are grouped into 18 categories:

  1. Manipulability of employees
  2. Know-how of employees
  3. Awareness of employees
  4. Scale of access
  5. Authentication
  6. Physical access
  7. Device Connectivity
  8. Network Architecture
  9. Communication & Data Transfer
  10. Vendor Vulnerabilities
  11. Application Vulnerabilities
  12. Incident Management
  13. Response Management
  14. System Configuration
  15. Hardware
  16. Change Management
  17. Privacy Impact Assessment



General Information

Name*: Malicious use of systems

Description:

Identifier: VID-043

Category: Manipulability of Employees

Current Assessment & Mitigation

misconfigured intentionally?

Figure 3.11: Edit a vulnerability.

18. Production Process Vulnerability

The list is filled with a total number of 66 vulnerabilities, each of them characterized by a "Name", "Description", "Hilti Score", "Mitigation Suggestion", a boolean "Affects impact", a boolean "PIA relevant", and a list of linked entities. For a better description of the parameters defined, refer to 3.2.

The linked entities are the connected components, in details the "Questions", "Threat Groups", "Types". The link between the vulnerability and one or more questions means that the weight of that specific vulnerability will change according to the answer to the relative question(s), affecting the likelihood of the related risk scenario(s).

At the same time, a vulnerability is correlated to one or more threat groups, to be connected to the risk scenarios.

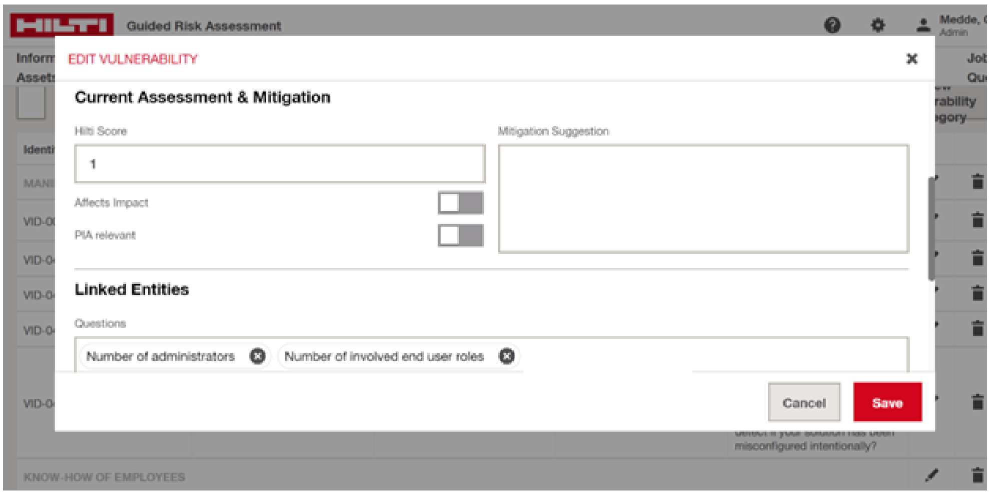
In the meantime, the link between vulnerability and types means that a specific vulnerability will be shown for the solution when those specific types will be chosen.

The list of the vulnerabilities can be handled by an admin in the relative section of the *Admin Panel*. As shown in Figures 3.11,3.12, 3.13, each vulnerability is editable.

- Threat Groups list

  Threat Groups are grouped into four categories:

  1. Availability



Current Assessment & Mitigation

Hilti Score: 1

Affects Impact

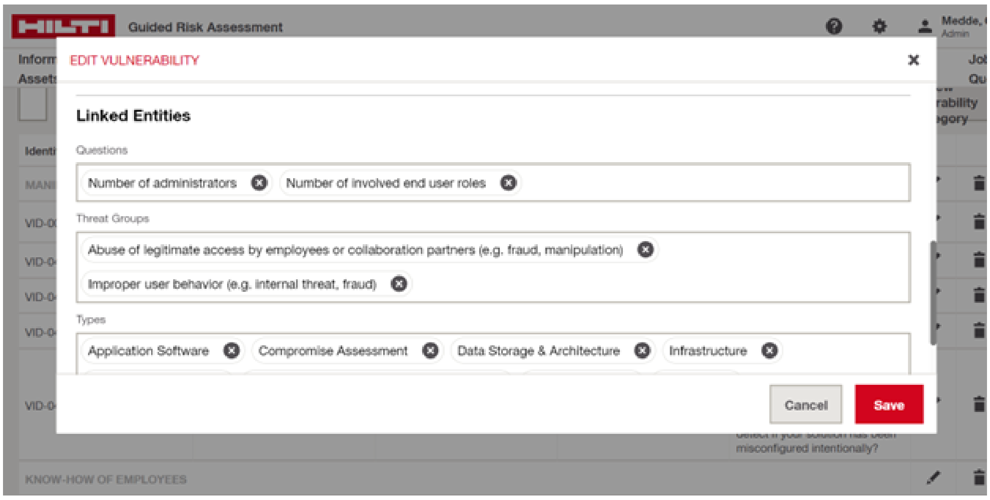
PIA relevant

Mitigation Suggestion

Linked Entities

Questions: Number of administrators, Number of involved end user roles

Figure 3.12: Edit a vulnerability.



Linked Entities

Questions: Number of administrators, Number of involved end user roles

Threat Groups: Abuse of legitimate access by employees or collaboration partners (e.g. fraud, manipulation), Improper user behavior (e.g. internal threat, fraud)

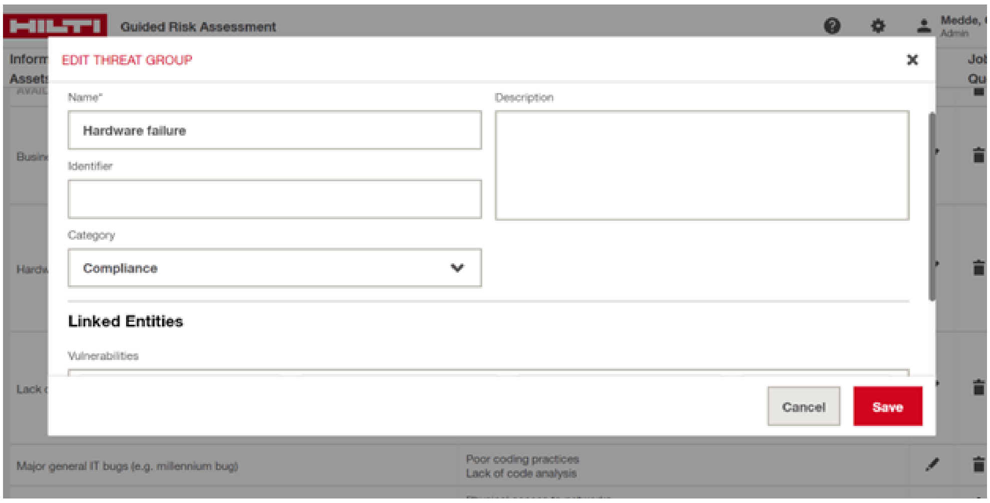
Types: Application Software, Compromise Assessment, Data Storage & Architecture, Infrastructure

Figure 3.13: Edit a vulnerability.

2. Compliance
3. Reliability
4. Security

The list is filled by 36 threat groups, each of them characterized by "Name", "Description" and linked entities.

In this case the linked entities are only the vulnerabilities correlated to that threat group.



Name*: Hardware failure

Identifier:

Description:

Category: Compliance

Linked Entities

Vulnerabilities

Major general IT bugs (e.g. millennium bug) — Poor coding practices, Lack of code analysis

Figure 3.14: Edit a threat.

The list of the threat groups can be handled by an admin in the relative section of the *Admin Panel*. As shown in Figure 3.14, each threat group is editable.

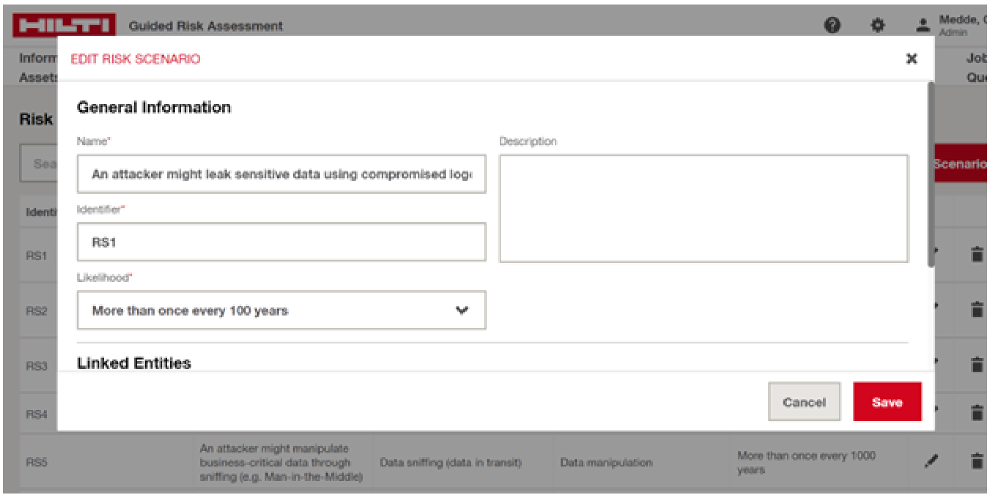
- Risk scenarios list

  The list is filled by 67 risk scenarios, and each of them is characterized by a "Name", "Description", "Likelihood" and linked entities. For a definition of the likelihood, refer to 3.2.

  The field "Likelihood" can be defined between eight choices:

  1. More than once every year
  2. Approximately once every year
  3. More than once every 10 years
  4. Approximately once every 10 years
  5. More than once every 100 years
  6. Approximately once every 100 years
  7. More than once every 1000 years
  8. Approximately once every 1000 years

  In this case, the linked entities are the risk effect which the risk scenario is related to, and the threat group. Since every risk scenario is characterized by only one threat group and one risk effect, we will have different risk scenarios based on these correlations.



EDIT RISK SCENARIO

General Information

Name*: An attacker might leak sensitive data using compromised logi

Description:

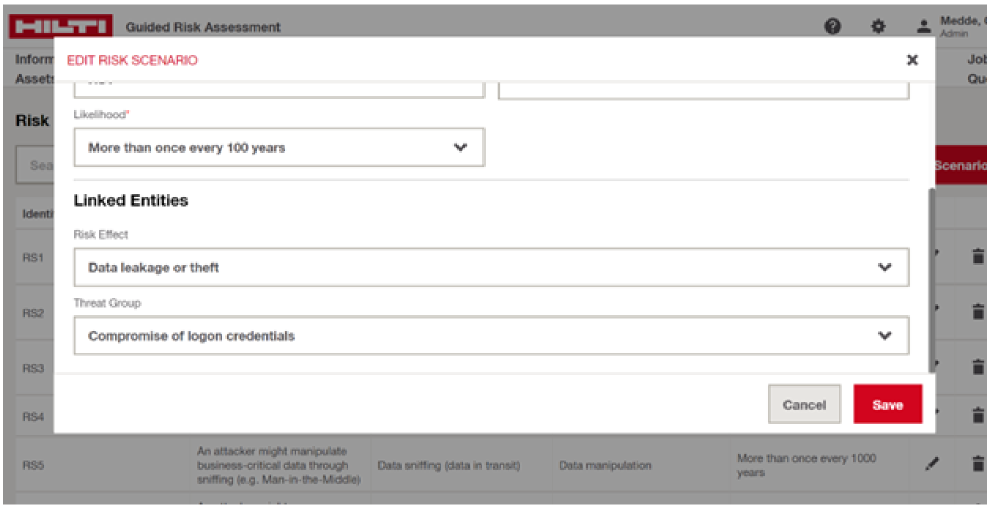
Identifier*: RS1

Likelihood*: More than once every 100 years

Linked Entities

RS5	An attacker might manipulate business-critical data through sniffing (e.g. Man-in-the-Middle)	Data sniffing (data in transit)	Data manipulation	More than once every 1000 years

Figure 3.15: Edit a risk scenario.



EDIT RISK SCENARIO

Likelihood*: More than once every 100 years

Linked Entities

Risk Effect: Data leakage or theft

Threat Group: Compromise of logon credentials

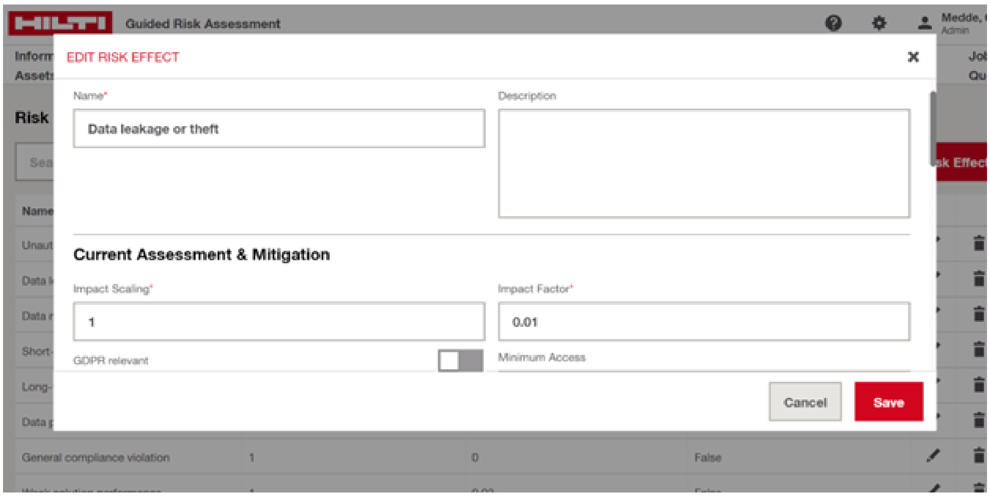
RS5	An attacker might manipulate business-critical data through sniffing (e.g. Man-in-the-Middle)	Data sniffing (data in transit)	Data manipulation	More than once every 1000 years

Figure 3.16: Edit a risk scenario.

The list of the risk scenarios can be handled by an admin in the relative section of the *Admin Panel*. As shown in Figures 3.15, 3.16, each risk scenario is editable.

- Risk effects list

  This list is filled by 12 risk effects, each of them characterized by "Name", "Description", "Impact Scaling", "Impact Factor", a boolean "GDPR relevant", "Minimum Access", and the impact mitigations.



Name* Data leakage or theft

Description

Current Assessment & Mitigation

Impact Scaling* 1

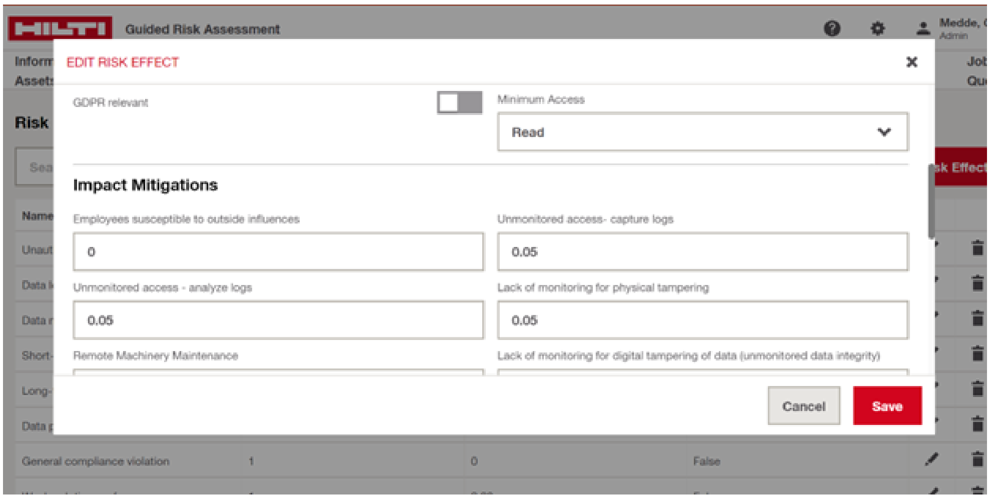
Impact Factor* 0.01

GDPR relevant

Minimum Access

General compliance violation 1 0 False

Figure 3.17: Edit a risk effect.



GDPR relevant

Minimum Access Read

Impact Mitigations

Employees susceptible to outside influences 0

Unmonitored access- capture logs 0.05

Unmonitored access - analyze logs 0.05

Lack of monitoring for physical tampering 0.05

Remote Machinery Maintenance

Lack of monitoring for digital tampering of data (unmonitored data integrity)

General compliance violation 1 0 False

Figure 3.18: Edit a risk effect.

The "Minimum Access" can be defined choosing between "None", "Read", "Read & Write", while the impact mitigations are the vulnerabilities affecting the impact of that risk effect, with relative impact factor, used in the impact calculation. For a better explanation of the parameters, refer to 3.2.

The list of the risk effects can be handled by an admin in the relative section of the *Admin Panel*. As shown in Figures 3.17, 3.18, each risk effect is editable.

- Questions list

As part of the assessment creations, different steps of surveys need to be filled by the user, in order to have a more accurate likelihood and impact calculation.

The survey in the step 3 is composed by 19 questions grouped into 4 categories:

1. Information on end users
2. Information on administrators and vendors
3. Information on system access and usage
4. Information on technological and operational change

Each question is connected to multiple options, each one having a different weight which will impact the likelihood of the risk scenario associated to the vulnerability associated to the specific question (e.g., the question "Number of end users" has one option to be answered "No end users", characterized by a weight 0.

Since the question is connected to the vulnerabilities "Poor authentication and password management", "IoT enable item not protected against reverse engineering", "Use of compromised components", the weight 0 will affect the calculation of each risk scenario related to these vulnerabilities).

This means that, apart for the field "Name", "Question", "Description", "Position", each question will be characterized by a list of vulnerabilities as linked entities, and a list of options, each one with a different value.

The list of the questions can be handled by an admin in the relative section of the *Admin Panel.* As shown in Figures 3.19, 3.20, each question is editable.

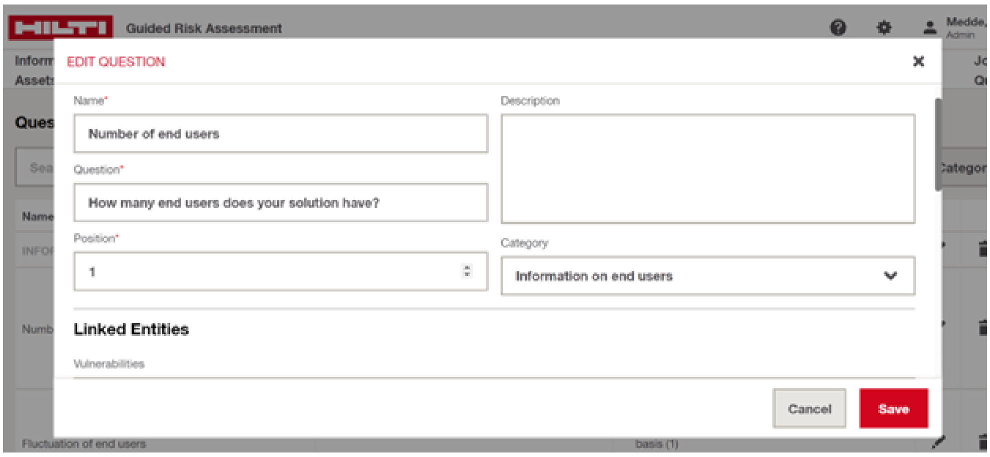
- Information Assets

  If on one side the questions of the step 3 affect the likelihood, on the other side the survey of the step 2 affects the impact.

  In this step there are 18 questions characterized by "Name", "Description", "Position", "Personal Data Usage", "Impact Assessment".

  If on one hand the Personal Data Usage can be defined as "None", "Personal data (dependent on add. Information)", "Personal data (direct identifiable)", "Special categories of personal data (dependent on add. Information)", on the other hand the Impact Assessment section is composed by the risk effects correlated to the question, with relative impact factors.

  The list of the information assets can be handled by an admin in the relative section of the *Admin Panel.* As shown in Figure 3.21, each question is editable.



Name*: Number of end users

Question*: How many end users does your solution have?

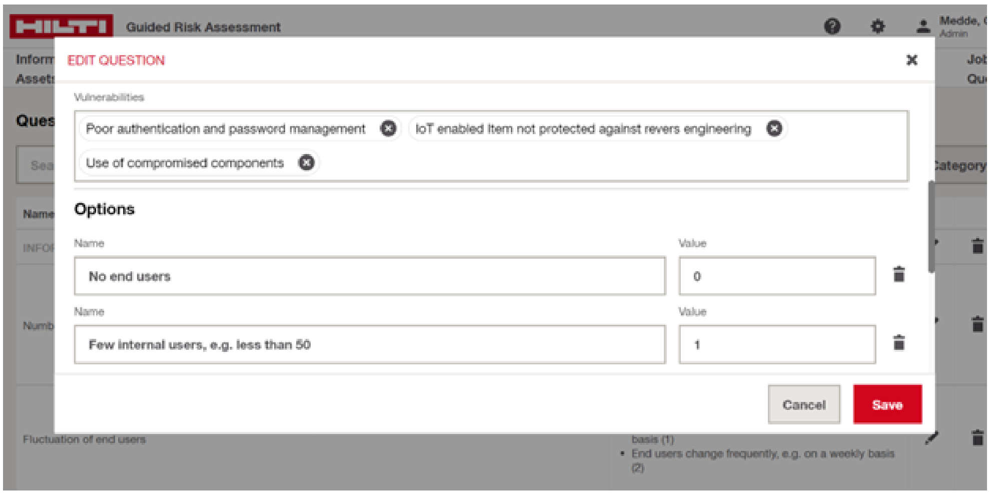
Position*: 1

Description:

Category: Information on end users

Vulnerabilities

Figure 3.19: Edit a question.



Vulnerabilities: Poor authentication and password management; IoT enabled Item not protected against revers engineering; Use of compromised components

Name	Value
No end users	0
Few internal users, e.g. less than 50	1

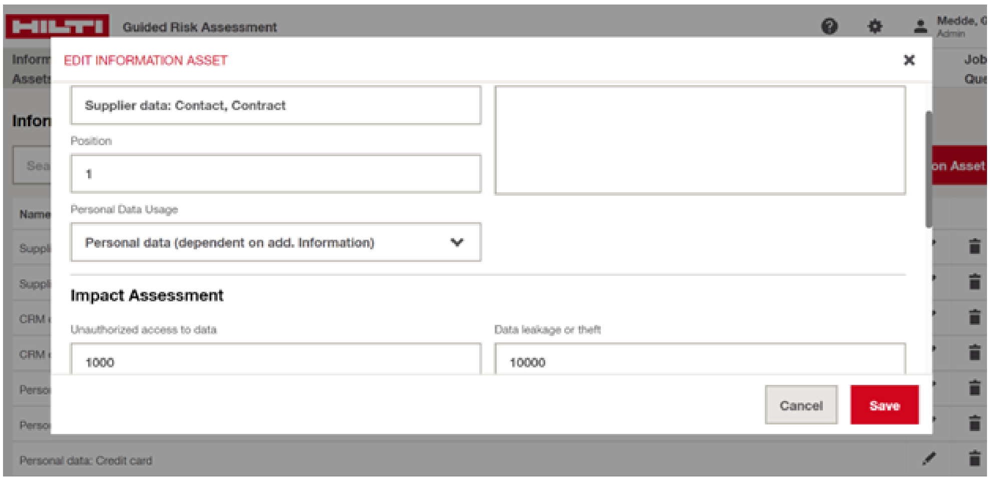
Figure 3.20: Edit a question.

3.2 The risk in the GRA

In Hilti perspective we can define the risk as "effect of uncertainty on objectives" (Reproduced from [26])

The terms *threat* and *risk* are often interchangeable in common life, but in the context of IT Risk Management is important to clearly distinguish between them. On one side we can recognize a *threat* as potential danger that can lead to a risk if a related vulnerability is exploited.

If we consider, for example, a "Physical access to the system" as a threat, it can



EDIT INFORMATION ASSET

Supplier data: Contact, Contract

Position: 1

Personal Data Usage: Personal data (dependent on add. Information)

Impact Assessment

Unauthorized access to data: 1000

Data leakage or theft: 10000

Personal data: Credit card

Figure 3.21: Edit an information asset.

lead to an effective risk scenario only if combined with an exploitable vulnerability, as for example "Servers room always open". That's the reason why the identification and mitigation process is so important. Throughout the controls put in place to mitigate the vulnerability, the threats will have nothing to exploit, and they cannot lead to an effective risk scenario and effect.

Considering again the previous example, if we insert a badge checker to handle the access to the servers' room, the control put in place will avoid the physical access caused by the always open door.

Every risk is characterized by two aspects:

- Impact: worst case effect on Hilti if the scenario would instantiate
- Likelihood: frequency that a threat may exploit a vulnerability in a Hilti specific scenario. This requires vulnerabilities on Hilti's side allowing the threat to act on Hilti.

(Reproduced from [32])

The IT Risk Framework followed by Hilti classifies four types of IT Risks as described in [32]:

1. Security: Prevention of disclosure, corruption or misused of electronic information to unauthorized individuals, entities, or processes
2. Compliance: Set of rules, standards, laws, and regulations to which IT must conform

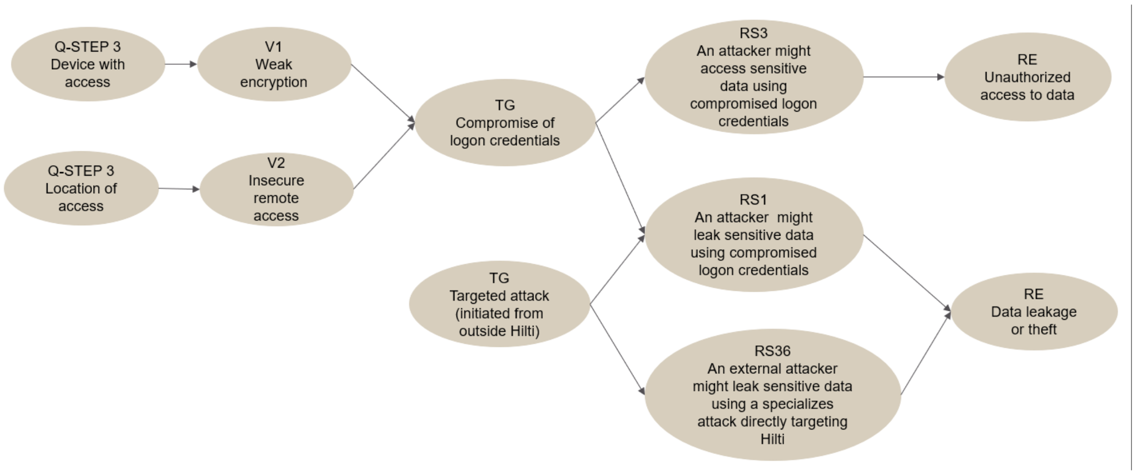

Q-STEP 3
Device with access
V1
Weak encryption
Q-STEP 3
Location of access
V2
Insecure remote access
TG
Compromise of logon credentials
TG
Targeted attack (initiated from outside Hilti)
RS3
An attacker might access sensitive data using compromised logon credentials
RS1
An attacker might leak sensitive data using compromised logon credentials
RS36
An external attacker might leak sensitive data using a specializes attack directly targeting Hilti
RE
Unauthorized access to data
RE
Data leakage or theft

Figure 3.22: Connections between components - Likelihood perspective.

3. Reliability: It should enhance value creation and business productivity

4. Availability: Accessibility and usability of electronic information and related processing systems by authorized entity

3.2.1 Likelihood calculation

Connections between components

As shown in the Figure 3.22, each vulnerability that affects the likelihood, is connected to one or more threat groups (if the vulnerability is not connected to at least one threat group, it doesn't affect the likelihood).

Since each threat group is connected to one or more risk scenarios, the vulnerabilities are connected to the risk scenarios through the threat groups.

Based to these connections, the affection of the likelihood of a risk scenarios by a vulnerability, is due to the flow:

1. Question Step 3 -> Vulnerability

2. Vulnerability -> Threat Group

3. Threat Group -> Risk Scenario

Parameters Involved

- Weight: the value defined for each answer of each question in the Step 3 *Questionnaire*. In the likelihood calculation only the weight of the answers chosen by the user during the analysis are considered.

- NormalizedIssueScore: the normalization between 0 and 1 of the result of the answers filled by the user in the Step 3 *Questionnaire*.

- Score: the issue score mapped from the *normalizedIssueScore*.

- MitigatedScore: the result of the mitigation of the vulnerabilities identified for the solution.

- HiltiScore: the value defined for a vulnerability during the vulnerability creation.

- RelevantVulnerabilities: the vulnerabilities identified for the Solution Type(s).

- Likelihood: the new value for the likelihood.

- HiltiMaxScore: the maximum HiltiScore of the relevant vulnerabilities.

- AssessmentMaxScore: the maximum value of the maximum between the *MitigatedScore* and the *HiltiScore* of the relevant vulnerabilities.

- HiltiMaxScoreCount: the number of the relevant vulnerabilities with *HiltiScore* equal to *HiltiMaxScore*.

- AssessmentMaxScoreCount: the number of the relevant vulnerabilities with *AssessmentMaxScore* equal to the maximum value of the maximum between the *MitigatedScore* and the *HiltiScore* of the relevant vulnerabilities.

Calculation process

At the creation of the assessment, each Risk Scenario has a specific likelihood value (represented by the parameter *CustomLikelihood*), defined during the insertion of the Risk Scenario in the Admin panel.

During the analysis, and after each change (change of the mitigation status or in the *Questionnaire* survey), the value of the likelihood of each Risk Scenario will be modified according to the parameters described in the 3.2.1.

The different stages of the calculation are show at high level in the diagram 3.23, and in details below:

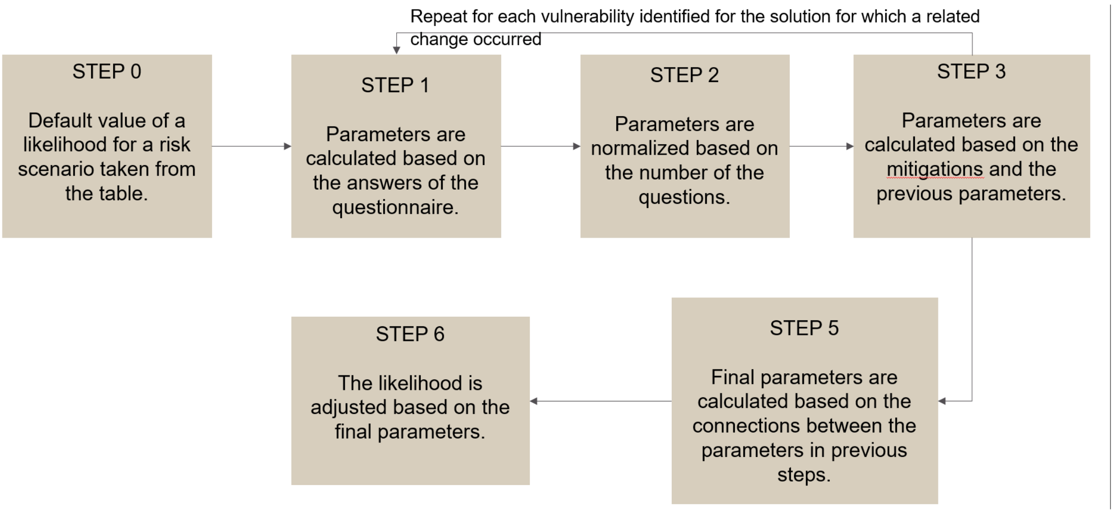

Repeat for each vulnerability identified for the solution for which a related change occurred
STEP 0
Default value of a likelihood for a risk scenario taken from the table.
STEP 1
Parameters are calculated based on the answers of the questionnaire.
STEP 2
Parameters are normalized based on the number of the questions.
STEP 3
Parameters are calculated based on the mitigations and the previous parameters.
STEP 5
Final parameters are calculated based on the connections between the parameters in previous steps.
STEP 6
The likelihood is adjusted based on the final parameters.

Figure 3.23: Likelihood calculation - High Level.

1. The first stage filters all the vulnerabilities (identified for the solution by the choice of the Type(s)), for which the mitigation status is changed, or the answer of a correlated question in the *Questionnaire* is changed. This means that during the first calculation (the first time the user access to the *Step 4* of the analysis), all the vulnerabilities are taken into consideration.

   At this point only the questions correlated to the vulnerabilities and for which the answer is changed are taken into consideration, with a loop in which the weight of the answer is considered.

   If the user provides an answer to the question, the relative value is considered, otherwise the maximum of the answer's wight is taken.

   Each weight is used to calculate the parameter *normalizedIssueScore*, following the formula:

   $$\text{normalizedIssueScore} = \text{weight}/\text{maxOptionWeight}$$

2. In the second stage, the parameter *normalizedIssueScore*, calculated in the previous stage, is normalized between 0 and 1, and the parameter *score* is calculated.

   The value of *normalizedIssueScore* has value 1 if no answer is changed for the questions connected to the vulnerability in the loop, otherwise is calculated considering the number of questions for which the answer is changed, following the formula:

   $$\text{normalizedIssueScore}/ = \text{numberOfQuestions}$$

The value of the parameter *score* is calculated according to a method which maps the normalized score for a vulnerability to an issue score.

For example, if n1 is one of the thresholds defined in the method, the score calculation follow: 1,

Algorithm 1 Score calculation

```
**if** $normalizedIssueScore > n1$ **then**
    $score = 0$
**end if**
```

3. The third stage takes into consideration the eventual mitigation of the vulnerability considered in that phase of the loop, with the purpose of calculate the *mitigatedScore* parameter.

   If the vulnerability has status *Not Mitigated*, the parameter *score* (calculated in the previous stage) is considered, while if the status is *Fully Mitigated*, the parameter *HiltiScore* (defined during the vulnerability definition in the Admin panel and specific for each vulnerability) is considered. Lastly, if the status is *Partially Mitigated*, the parameter is calculated following the formula:

   $$\text{mitigatedScore} = \text{HiltiScore} + \text{abs}(\text{score} - \text{HiltiScore})/2.0$$

4. We can consider as Stage 4 the continue of the loop of the vulnerabilities for which a change is detected in the mitigation status, or in the answer of the relative questions of the *Questionnaire*.

   At the end of this stage we will have defined the following parameters for each vulnerability: *HiltiScore, normalizedIssueScore, score, mitigatedScore*

5. In this fifth stage, the risk scenarios are considered into a loop, filling the parameter *relevantVulnerabilities* with all the vulnerabilities connected to the threat group already connected to that specific risk scenario, and connected to the solution type(s) of the specific assessment into analysis.

   At this point, considering the *likelihoodPreDefined* as the value of the likelihood before the new calculation, the final parameters are calculated based on:

   - *likelihood*: 2

     Please, note that the value of the parameter *other-likelihood* is not specified, because the work of reverse engineering did not clarify its meaning and origin.
   - *hiltiMaxScore*: 3
   - *assessmentMaxScore*: 4

Algorithm 2 Likelihood calculation.

```
if $likelihoodPreDefined > 0$ then
    likelihood = 1/likelihoodPreDefined
else
    likelihood = 1/otherLikelihood
end if
```

Algorithm 3 Hilti Max Score calculation.

```
if $numberOfRelevantVulnerabilities > 0$ then
    hiltiMaxScore = max(HiltiScore)
else
    hiltiMaxScore = 0
end if
```

Algorithm 4 Assessment Max Score calculation.

```
if $numberOfRelevantVulnerabilities > 0$ then
    assessmentMaxScore = max(mitigatedScore, HiltiScore)
else
    assessmentMaxScore = 0
end if
```

- *hiltiMaxScoreCount*: if n is the number of relevant vulnerabilities for which

  $$\text{HiltiScore} == \text{hiltiMaxScore}$$

  the parameter *hiltiMaxScoreCount* is calculated as follow: 5

Algorithm 5 Hilti Max Score Count calculation.

```
if $numberOfRelevantVulnerabilities > 0$ then
    hiltiMaxScoreCount =
    = n
else
    hiltiMaxScoreCount = 0
end if
```

- *assessmentMaxScoreCount*: if m is the number of relevant vulnerabilities for which

  $$\max(\text{mitigatedScore}, \text{HiltiScore}) == \text{assessmentMaxScore}$$

  the parameter is calculated as follow: 6

Algorithm 6 Assessment Max Score Count calculation.

```
if $numberOfRelevantVulnerabilities > 0$ then
    assessmentMaxScoreCount = m
else
    assessmentMaxScoreCount = 0
end if
```

6. In the sixth and last stage, the likelihood of the risk scenario is calculated based on the previous parameters. In particular:
   - If the difference between the issue scores *assessmentScore, hiltiMaxScore* is greater than 0.5, the consequence is a big increase of the likelihood.
   - If the difference is just a small increase, the consequence depends on the value of the current likelihood.
   - Otherwise the likelihood is inherited.

Practical Example

1. As shown in the Figure 3.24, consider a vulnerability V1 with *HiltiScore* 1, related to 2 questions in the *Questionnaire*, and each of the 2 questions has 2 options (possible answers), with weight respectively 0 and 1.

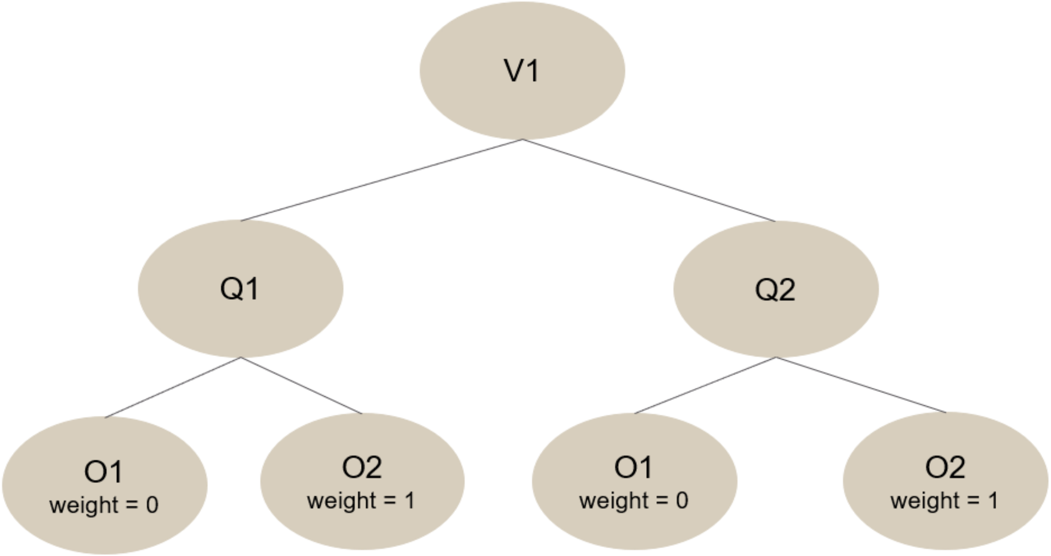


Figure 3.24: Stage 1 - Likelihood calculation

If the user, during the filling of the *Questionnaire*, select the option O1 for the question Q1, and the option O2 for the question Q2, after the first selection:

$$\text{normalizedIssueScore} = \text{weightO1}/\text{maxWeightQ1} = 0/1 = 0$$

After the second selection:

$$\begin{aligned}&\text{normalizedIssueScore} =\\ &= \text{normalizedIssueScore} + \text{weightO2}/\text{maxWeightQ2} =\\ &= 0 + 1/1 = 1\end{aligned}$$

2. Since the *normalizedIssueScore* value is 1, and we are considering only 1 vulnerability:

$$\begin{aligned}&\text{normalizedIssueScore} =\\ &= \text{normalizedIssueScore}/\text{numberOfQuestions}\\ &= 1/2 = 0.5\end{aligned}$$

According to the method mentioned above 2:

3. We can assume that the vulnerability status is changed to *Partially Mitigated*:

$$\begin{aligned}&\text{abs}(\text{score} - \text{HiltiScore})/2.0 = 0.5/2 = 0.25;\\ &\text{mitigatedScore} = 1 + 0.25 = 1.25\end{aligned}$$

```
if $normalizedIssueScore = 0.5$ then
    score = 1.5
end if
```

4. We can consider now to have in total 2 vulnerabilities for which the mitigation status of the answer of the relative questions in *Questionnaire* is changed, in particular the second vulnerability is V2. Repeating the previous 3 stages, the parameters calculated for the vulnerability V2 are:

$$\begin{aligned}&\text{HiltiScore} = 2;\\&\text{normalizedIssueScore} = 0.5;\\&\text{score} = 1.5;\\&\text{mitigatedScore} = 2\end{aligned}$$

5. In this stage we can calculate the final parameters based on the vulnerabilities previously considered, and considering the likelihood pre-defined for the risk scenario into consideration (first in the loop and last in this example), equal to 5, corresponding to "Approximately once every year":

$$\text{customLikelihood} > 0 \Rightarrow \text{likelihood} = 1/5 = 0.2; \tag{3.1}$$

$$\max(\text{HiltiScoreV1}, \text{HiltiScoreV2}) = \text{HiltiScoreV2} \tag{3.2}$$

$$\text{relevantVulnerabilities} > 0 \land 3.2 \Rightarrow \text{HiltiMaxScore} = 2; \tag{3.3}$$

$$\begin{aligned}&\max(\text{HiltiScoreV1}, \text{HiltiScoreV2}, \text{mitigatedScoreV1}, \text{mitigatedScoreV2}) =\\&= \text{HiltiScoreV2} = \text{mitigatedScoreV2}\end{aligned} \tag{3.4}$$

$$\text{relevantVulnerabilities} > 0 \land 3.4 \Rightarrow \text{assessmentMaxScore} = 2 \tag{3.5}$$

$$\begin{aligned}&\text{relevantVulnerabilities} > 0 \land \text{onlyV2withHiltiScore} = \text{HiltiMaxScore} \Rightarrow\\&\Rightarrow \text{hiltiMaxScoreCount} = 1\end{aligned} \tag{3.6}$$

$$\begin{aligned}&\text{onlyV2withmax}(\text{HiltiScore}, \text{mitigatedScore}) = \text{assessmentMaxScore} \Rightarrow\\&\text{assessmentMaxScore} \Rightarrow \text{count} = 1;\end{aligned} \tag{3.7}$$

$$\text{relevantVulnerabilities} > 0 \land 3.7 \Rightarrow \text{assessmentMaxScoreCount} = 1 \tag{3.8}$$

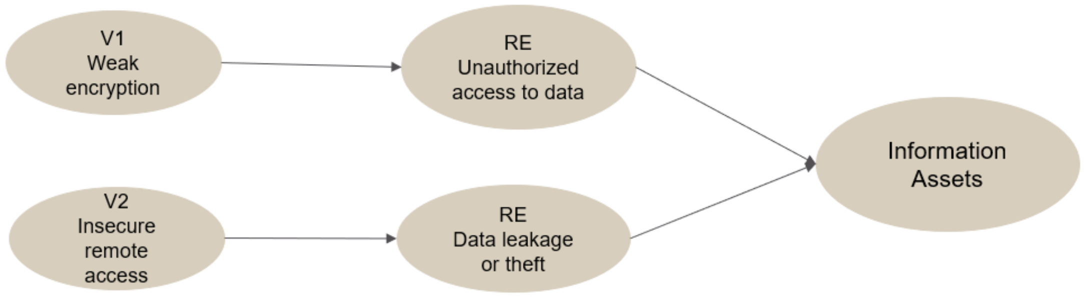


Figure 3.25: Connections between components - Impact perspective.

6. According to the parameters calculated in 3.1, 3.3, 3.5, 3.6, 3.8:

$$\begin{aligned}&\text{assessmentMaxScore} = \text{hiltiMaxScore}\wedge\\&\wedge\text{hiltiMaxScoreCount} = \text{assessmentMaxScoreCount}\wedge\\&\wedge\text{likelihood} = 0.2 \Rightarrow \text{likelihood} = \max(0.001, \text{likelihood}/10) = 0.02\end{aligned}$$

3.2.2 Impact calculation

Connections between components

As shown in the Figure 3.25, each vulnerability that affects the impacts (with the boolean *Affects Impact* set to true), is connected to a risk effect (with an impact factor defined for each of them), and the questions of the Step 2 *Information Assets* are directly connected to the risk effects.

So, the impact calculation is due to the affection of both *Vulnerabilities* and *Information Assets*.

Parameters Involved

- impacts: the list of the impacts defined for the risk effect at which the risk scenario is connected, for which the access of the *Information Asset* is greater than the minimum access defined for the risk effect.
- dataImpact: the impact in terms of data processed/stored/transferred within the solution and involves the risk scenario and effect
- businessImpact: the impact in terms of business

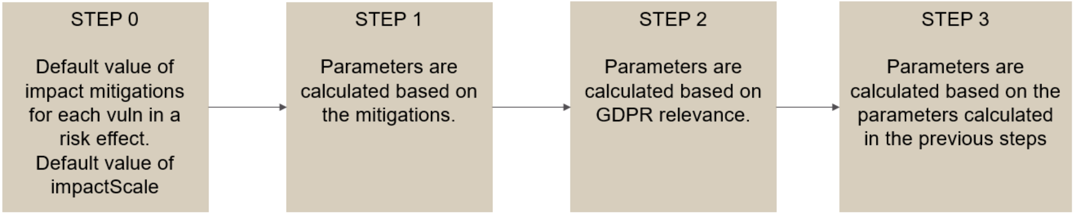

STEP 0
Default value of impact mitigations for each vuln in a risk effect.
Default value of impactScale
STEP 1
Parameters are calculated based on the mitigations.
STEP 2
Parameters are calculated based on GDPR relevance.
STEP 3
Parameters are calculated based on the parameters calculated in the previous steps

Figure 3.26: Impact calculation - High Level.

- max: the number of the impacts (first parameter)
- customImpactFactor: the factor *Impact Scaling* defined for the risk effect
- impactMitigations: the vulnerabilities that affects the impact (with the boolean *Affects Impact* set to true)
- impactMitigationFactor: the total value of the mitigations of the impactMitigations
- impactVulnFactor: the impact value pre-defined for each vulnerability inside each risk effect
- customImpact: related to the risk scenario, the study of the calculation did not clarified where this factor is defined
- PIAcompleted: boolean, true if all the vulnerabilities connected to the risk scenarios are not PIArelevant, or if are all PIA relevant but none of them is fully mitigated

Calculation Process

At the creation of the assessment, each vulnerability that affects the impact has a specific factor of *impact mitigation*, and the value of *customImpactFactor* is defined for the risk effect.

The process of the calculation of the impact, starts at the end of the calculation of the likelihood, into the same loop of the risk scenarios.

The different stages of the calculation are shown at high level in the diagram 3.26, and in details below:

1. In the first stage, the parameters are calculated based on the mitigations of the vulnerabilities affecting the impact. In particular, through a loop of all the vulnerabilities affecting the impact: 7

Algorithm 7

```
if $vulnerabilitystatus = NotMitigated$ then
    impactMitigationFactor = 0
else
    if $vulnerabilitystatus = PartiallyMitigated$ then
        impactMitigationFactor = impactVulnFactor/2
    else
        impactMitigationFactor = impactVulnFactor
    end if
end if
```

2. In this second stage, if the *customImpact* of the risk scenario is already defined, then the parameters become:

$$\begin{aligned}&\text{dataImpact} = \text{customImpact};\\&\text{businessImpact} = 0;\\&\text{customImpactFactor} = 1;\end{aligned}$$

If the *customImpact* is not defined, but the risk effect is GDPR relevant (the boolean *GDPR relevant* is set to true), then:

$$\begin{aligned}&\text{customImpactFactor} = 1;\\&\text{businessImpact} = 0;\end{aligned}$$

If *max* is the maximum impact factor of the impact for which the PrivacyScope is greater than None, the *PIAcompleted* is considered and the *dataImpact* is calculated as follow: 8

Algorithm 8

```
if $PIAcompleted = false \wedge (\forall impactsPrivacyScope > None)$ then
    dataImpact = fixedValue
else
    if $(\forall impactsPrivacyScope > None)$ then
        dataImpact =
    else
        dataImpact = 0
    end if
end if
```

If the risk effect is not GDPR relevant, a geometric formula is used to evaluate the *dataImpact* and *businessImpact*, considering in the first case the impact factors of each impacts, and in the second one the cost and the impact factor of the risk effect connected to the risk scenario.

3. In the third and last stage, the final parameters are calculated, and based on them, the final impact can be calculated:

   - calculatedLikelihood: if *likelihood* is the likelihood defined for the risk scenario, 9
   - calculatedDataImpact

     $$\begin{aligned}&\text{calculatedDataImpact} =\\&= \text{dataImpact} * (1 - \text{impactMitigationFactor}) * \text{customImpactFactor}\end{aligned}$$

Algorithm 9

```
if $riskEffectIsGDPRrelevant$ then
    calculatedLikelihood = 1
else
    calculatedLikelihood =
end if
```

- calculatedBusinessImpact

  $$\begin{aligned}&\text{calculatedBusinessImpact} = \\ &= \text{businessImpact} * (1 - \text{impactMitigationFactor}) * \text{customImpactFactor}\end{aligned}$$

- calculatedExpectedLoss

  $$\begin{aligned}&\text{calculatedExpectedLoss} = \\ &\text{calculatedLikelihood} * (\text{calculatedDataImpact} + \text{calculatedBusinessImpact})\end{aligned}$$

- calculatedRelevance: based on the expectedLoss, an entire value is calculated.

  The impact that will be represented in the heatmap will be the sum between the *dataImpact* and the *businessImpact*, with a random factor called *randomImpact*.

Final Consideration

Through an in-depth analysis of the tool, that led me to apply reverse engineering techniques to understand the behaviour of the tool, and code analysis to make the computation clear, I identified several gaps not only in the content of the list components (e.g., *Vulnerability* list), but also in the structure and strategy, as described in the next chapter.

Chapter 4

GRA gaps

After a general overview of the concepts involved and the tool taken into consideration, the core part of the work of this thesis is more concerning the gaps and the improvements of the Guided Risk Assessment tool.

At the beginning of this thesis work, the several gaps identified by the Cybersecurity team seemed to be related only to the way in which the pre-defined components (e.g., Vulnerabilities, Types etc.) were written and identified for the solution. The documentation of the tool implementation was not sufficient to explain the connections between the components, the risk calculation, the structure and strategy with which the analysis was performed.

For this reason, the following steps during this thesis work helped me to highlight the real reasons behind the inconsistencies and the gaps:

1. Analyse the documentation:

   As first step of this gaps identification, an in-depth analysis of the documentation and the formulas behind the calculation, conducted me to understand the general structure of the tool, giving me an overview of the work performed by the different steps proposed by the tool during the analysis of an assessment. As highlight above, it was not sufficient, and took me through the next steps.

2. Analyse the gaps collected by the Cybersecurity team:

   The various inconsistencies and gaps collected by the Cybersecurity team over time, gave me an idea of the most problematic phases of the analysis, identifying in parallel the needs of the team, and the components that required more effort to be improved.

3. Analyse the DataBase structure:

   After the identification of the gaps in general, the analysis of the DataBase tables and connections became essential to understand the reason behind them. Since two version of the tool were implemented over time, two different DataBase structures were analysed and considered, to highlight the incosistencies not only related to the names of the tables and components, but also related to the way in which the analysis was performed (e.g., Solution Types and Questions connected to Vulnerabilities etc.), leading me to be able to compare how the tool should perform the analysis versus which connections were not being considered and used, with a result completely different from the one supposed and sought.

4. Reverse-engineering of the tool:

   Since inconsistencies in the names of the components, and in the connections, were found in the DataBase structures, an in-depth analysis of the tool became crucial. A reverse engineering effort, including:

   - Analyse the lists in the Admin panel;
   - Create new test assessments;
   - Change the information provided to the tool during test analysis;
   - Analyse the different results;
   - Change the lists in the Admin panel
   - etc.

   brought me to understand the way in which the tool considered the information, how the vulnerabilities were identified for the solution, the role of the different components, and which steps could make the difference during an analysis, highlighting also the reasons behind the inconsistencies, not only related to the form and the content of the lists, but also related to the strategy followed during the process. Since after this step, the calculation of the risk was still impossible to understand, another step became necessary.

5. Code analysis:

   Through the access of the repository, and the analysis of the code, a better understanding of the calculation helped me to really understand the reasons behind the inconsistencies in the heatmap, which parameters were crucial and from where they were derived. Even if some parameters still remain not clear, and other required some assumptions, thanks to these steps a better understanding of the functionalities, the calculation, and the goal that the tool should achieve, helped me to identify not only the problems in the surface (visible to the user), but also the inconsistencies behind them, giving me the opportunity to improve the tool from the root.

Since the gaps have been found both in content (which are the information and how they are managed inside the tool) and container (how information is shown inside of the tool), two separate but deeply interconnected paragraphs will help to better identify them.

4.1 Content Gaps

- Solution deliverable type

  Due to the concept of solution deliverable types itself (the system is represented through one or more *Solution Deliverable Type(s)* that are connecte to the vulnerabilities, limiting the only association between solution and vulnerabilities due to the types), some issues creating a new GRA assessment have been found. In some cases, the fixed list of types doesn't contain the solution needed for the assessment, in other cases is not clear what the type is referred to.

- Connection between risk-scenarios, threat groups, vulnerabilities

  As shown in the diagram 4.1, the structure of the GRA provides multiple risk-scenarios connected to one threat group. Since each threat group is connected to multiple vulnerabilities, we can assume that the connection between risk-scenarios and vulnerabilities affecting the likelihood goes through the threat group table.

  We can see different inconsistencies already having a look at the names.

  Firstly, a threat group should contain a list of threats, not a list of vulnerabilities.

  Secondly, according to the definition of a risk scenario, as highlighted in the previous theoretical chapters, it should be defined by a threat exploiting a vulnerability, that means that only one vulnerability and one threat should be connected to a risk scenario, and the risk scenario itself should define the connection between threats and vulnerabilities.

- Connection between vulnerability and solution deliverable types

  As shown in the figure 4.2, the decision of the vulnerabilities to be inserted into an assessment of a solution, is strictly correlated to the decision of the solution deliverable types during the creation of the assessment.

  This connection leads to different problems.

  First, not all the types cover all the different types of solutions since types to be implemented can change according to technology.

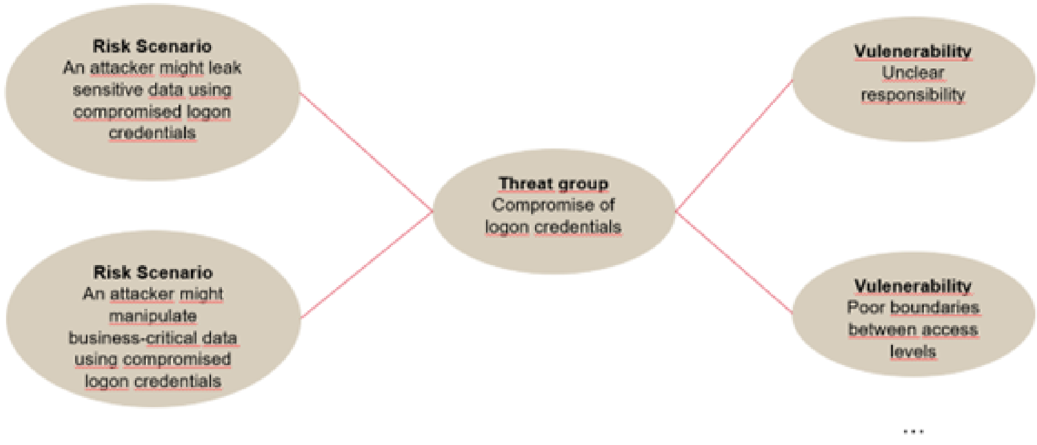

Risk Scenario
An attacker might leak sensitive data using compromised logon credentials
Risk Scenario
An attacker might manipulate business-critical data using compromised logon credentials
Threat group
Compromise of logon credentials
Vulenerability
Unclear responsibility
Vulenerability
Poor boundaries between access levels
...

Figure 4.1: Connection between risk-scenarios, threat groups, vulnerabilities.

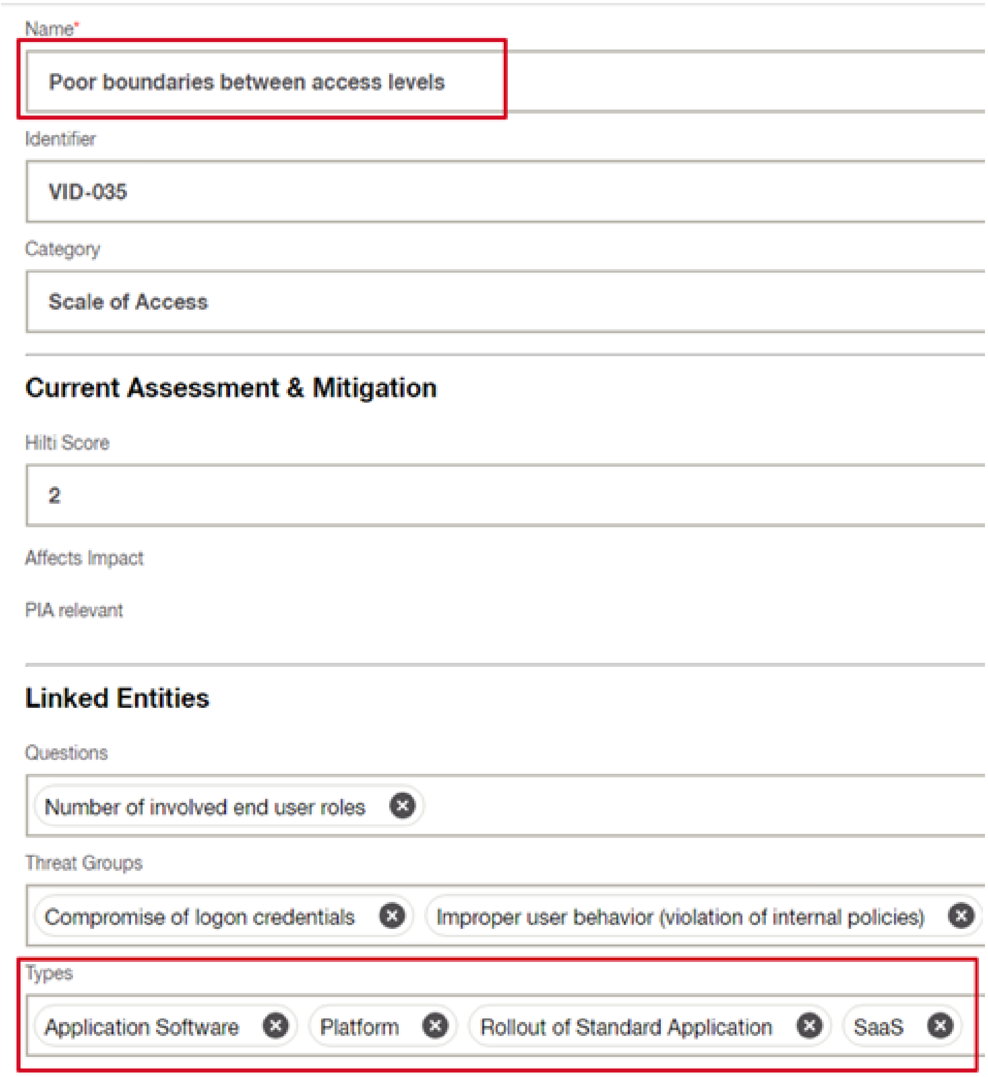
Second, the types classification cannot cover all the possible implementation details, leading to have a lot of vulnerabilities that don't make sense with the solution to be implemented (e.g., a Solution Deliverable Type can be connected to a vulnerability related to authentication, but if the solution to be implemented doesn't require authentication, or uses an already approved authentication mechanism, the vulnerability is still to be mitigated and managed).

Third, there is no connection between the questions in the Questionnaire - Step 2 and the choice of the vulnerabilities to be inserted, the only way in which the vulnerability is affected by the answers of the survey is by the weight that affects the likelihood of the risk scenario associated to that.

All these problems lead the user to have a huge number of vulnerabilities to mitigate and explain, without any connection to the solution to be implemented.

- Vulnerability list

  Due to the connection with Threat Groups, there are some inconsistencies in the form in which the vulnerabilities are written. As shown in the example 4.3, in the same vulnerabilities table, there are real vulnerabilities, but also threats. Going into details, in the current GRA there are 37 vulnerabilities written in the right form, and 18 threats, with 10 properties dealing with the GDPR.



Name*

Poor boundaries between access levels

Identifier

VID-035

Category

Scale of Access

Current Assessment & Mitigation

Hilti Score

2

Affects Impact

PIA relevant

Linked Entities

Questions

Number of involved end user roles

Threat Groups

Compromise of logon credentials; Improper user behavior (violation of internal policies)

Types

Application Software; Platform; Rollout of Standard Application; SaaS

Figure 4.2: Connection between vulnerability and solution deliverable types.

It leads not only to inconsistencies in concepts, but also in unclarity in what a vulnerability represents and how it should be mitigated.

- Risk-scenarios list

  Due to the wrong connections and definitions between the three components already mentioned above, the risk scenarios in the table are not written and defined appropriately.

  Since the definition requires a threat and an exploitable vulnerability to recognize a possible risk occurrence, each definition should highlight these

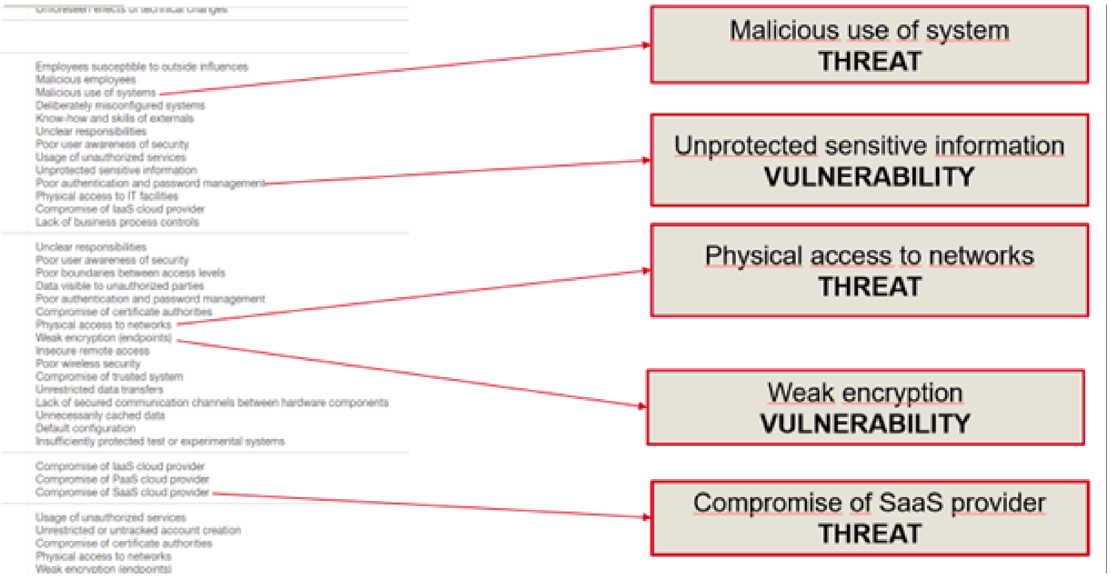

Unforeseen effects of technical changes
Employees susceptible to outside influences
Malicious employees
Malicious use of systems
Deliberately misconfigured systems
Know-how and skills of externals
Unclear responsibilities
Poor user awareness of security
Usage of unauthorized services
Unprotected sensitive information
Poor authentication and password management
Physical access to IT facilities
Compromise of IaaS cloud provider
Lack of business process controls
Unclear responsibilities
Poor user awareness of security
Poor boundaries between access levels
Data visible to unauthorized parties
Poor authentication and password management
Compromise of certificate authorities
Physical access to networks
Weak encryption (endpoints)
Insecure remote access
Poor wireless security
Compromise of trusted system
Unrestricted data transfers
Lack of secured communication channels between hardware components
Unnecessarily cached data
Default configuration
Insufficiently protected test or experimental systems
Compromise of IaaS cloud provider
Compromise of PaaS cloud provider
Compromise of SaaS cloud provider
Usage of unauthorized services
Unrestricted or untracked account creation
Compromise of certificate authorities
Physical access to networks
Weak encryption (endpoints)
Malicious use of system
THREAT
Unprotected sensitive information
VULNERABILITY
Physical access to networks
THREAT
Weak encryption
VULNERABILITY
Compromise of SaaS provider
THREAT

Figure 4.3: Vulnerability vs Threat.

two factors. In the current GRA, instead, a huge portion of the risk scenarios are written in a "threat form" or in a "vulnerability form", as is shown in the example 4.4, in which we can identify with red colour the threats, with blue colour the vulnerabilities.

Taking as example the risk-scenario "An attacker might manipulate business-critical data through sniffing", the "manipulate data" is the threat, the action that an attacker can perform, the "sniffing" is the used technique, but the vulnerability part is missing.

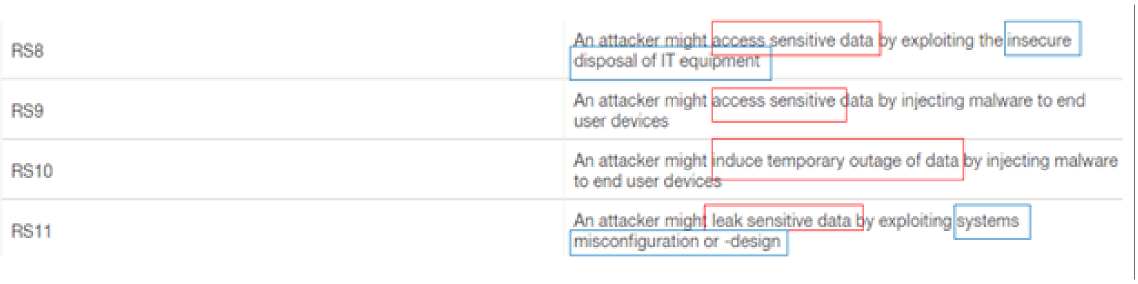
To better highlight the difference, we can take into consideration another risk scenario "An attacker might introduce temporary outage of data by exploiting system misconfiguration".

In this last example, the "system misconfiguration" is the vulnerability in the solution that, if exploited by the "introduce temporary outage of data" threat, can lead to the risk scenario with a certain likelihood and impact, and the relative "Short-term non-availability of the solution" risk effect.

- Risk calculation

  In conducting some tests with different assessments, types, answers to questions, some problems and inconsistencies with the risk calculation appeared.

  Having a look at the heatmaps 4.5, we can notice how the position of the risk scenario RS40 changes according to the mitigation of the related vulnerabilities.



RS8	An attacker might access sensitive data by exploiting the insecure disposal of IT equipment
RS9	An attacker might access sensitive data by injecting malware to end user devices
RS10	An attacker might induce temporary outage of data by injecting malware to end user devices
RS11	An attacker might leak sensitive data by exploiting systems misconfiguration or -design

Figure 4.4: Risk scenarios inconsistencies.

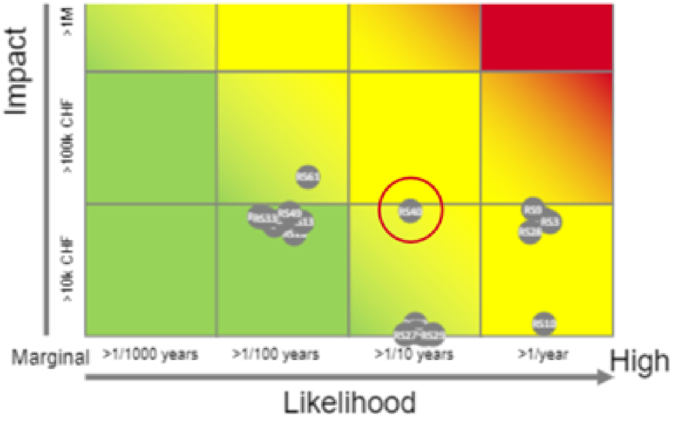


Figure 4.5: Initial position RS40.

Going into details, the first picture shows how the bubbles representing the risk scenarios appear according to changes of the Information Asset.

Trying then to partially mitigate one vulnerability, "Weak encryption", nothing changes.

Partially mitigating the "Know-how" vulnerability, the position of the bubble in the heatmap 4.6 changes, going to the right, to a more probable likelihood.

Then, fully mitigating the same vulnerability, the bubble changes position again 4.7, this time going to the left, through a better likelihood scenario.

It is clear how it represents an inconsistence, since a bubble should decrease the likelihood a bit with the partial mitigation and then more with the fully mitigation, but in this case, it increases with the partial and then decreases with the fully mitigation.

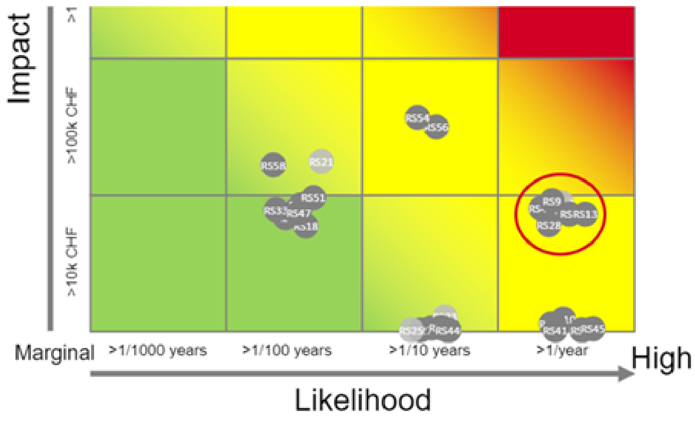


Figure 4.6: Position RS40 after a partial mitigation of one vulnerability.

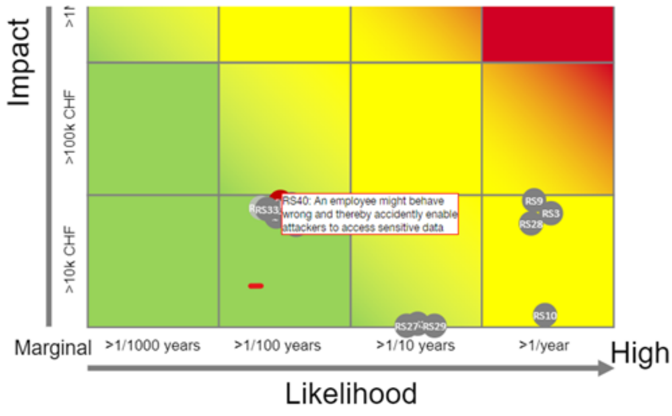


Figure 4.7: Position RS40 after a fully mitigation of same vulnerability.

It is due to a random factor inserted in the calculation of the likelihood.

4.2 Container Gaps

Talking about the container, an in-depth analysis and usage of the tool lead to some problems, inconsistencies and bugs that can cause a loss of usability and effectiveness.

- Users' representation

  In the assessments, some users are reported with an ID, some others with Name and Surname. It can lead to misunderstanding and unclarity for users who must insert editors, viewers etc. since they could not know how search/insert them.

- "No type" option

  During the creation of an assessment, the Solution Deliverable Types list also contains the option "No type", but since the vulnerabilities are directly inserted by the choice of the type, it leads to inconsistencies, having assessments without type, vulnerabilities, and consequent absence of risk scenarios and heatmap.

- "Internal" and "external" users cannot be inserted at the same time

  Inside the questionnaire, step 2, there is a question related to the number of the end users. It is a multiple-choice question, but the answers "internal" and "external" users are mutually exclusive, which means that a solution cannot have both internal and external users at the same time.

  It leads to unclarity filling the questionnaire and in the risk calculation.

- Bug deals with comments section

  During the Step 4 of the assessment, the number of comments is shown. Clicking in the "List view" button, comments can be readable and inserted. Since the save button is not disabled when the bar to write a comment is not empty, a user can save without inserting the comments, losing it.

  This can lead to problems of usability of the tool at user side.

The effort in understanding the functionalities of the Guided Risk Assessment tool, in the gaps identification, and the cooperation with the Hilti Cybersecurity team, lead me to identify and put in place the improvements, with the purpose of, on one side make the user experience easier and better as possible, and on the other side build solutions as secure as possible in Hilti perspective.

Chapter 5

GRA Improvements

The previous chapter described the process I followed in understanding the functionalities of the tool, the connections between the components, the structure and strategies behind the calculation of the risk, and the identification of the existing GRA's gaps.

In order to put into practice the knowledge and the awareness of the problems with which the tool has to face, to provide a complete and robust risk assessment of a solution, I started questioning the entire structure and strategy, from the basis to the top.

At this purpose, the next paragraphs will describe not only the effective improvements, but also the process that led me to implement them, to deeply understand the reasons behind every choice made.

5.1 Database structure improvement

As shown in the diagram 5.1, the database is structured into two sections, one related to the Vulnerabilities and Threats (and consequently to the components associated, as Types, Questions etc.), and one related to the Risk. Thanks to the effort spent in understanding the strategy followed by the tool during the analysis, as described in 4, I found several inconsistencies not only in the way in which the tables are connected, but also in the tables themselves.

If we focus on the vulnerability part (highlighted in the diagram 5.1 with the beige color), we can notice some of the inconsistencies described in the previous chapter4.

With the aim of solving them, the cooperation with different members of the Cybersecurity team, became crucial. Indeed, in particular for the gaps regarding the threat part, described here 4.1, the proposal in changing the tables and the connections between them, was discussed by the Product Owner of the IT & Cyber

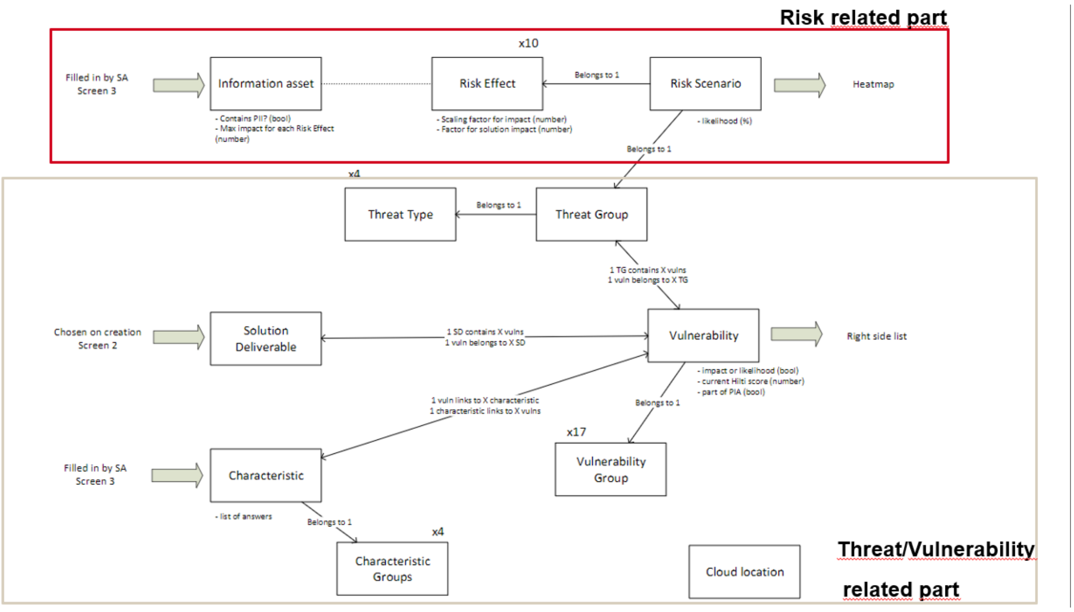


Figure 5.1: Current GRA: DB structure.

Risk Management, in order to align the needs of the tool with the risk management strategy followed at Hilti.

As shown in the Figure 5.2, firstly I changed the tables, in particular:

- The table *Threat Type*, containing the categories of the threats, in the new DB structure takes the name of *Threat Groups*, with the purpose of collecting the categories of the threats referencing to the IRAM2 framework described in 2.2.1.

- The table *Threat Group*, shifted to take the place of the table *Threat Type*, is substituted with a table *Threat*, in order to pre-define a list of threats, categorized as described in the first point.

- The table *Solution Deliverable*, is removed, since the researches demonstrated that the use of the Type(s) to categorize a system is not efficient to identify all the vulnerabilities that really fit for the solution, it is too vague and it does not cover all the possible implementations.

- The tables *Characteristic* and *Characteristic Group*, containing the questions to be answered by the user in the *Step 3 - Questionnaire*, are replaced with four tables *Question*, *Answer*, *Option*, *Weight*, in order to implement a conditional questions path, to filter the vulnerabilities since the beginning of the analysis, identifying only the relevant ones.

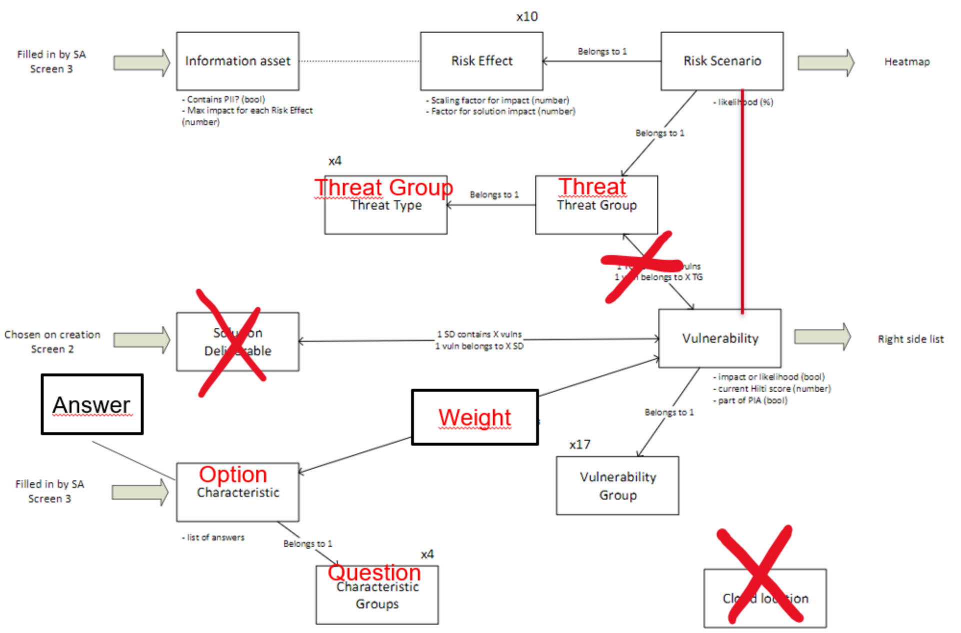


Figure 5.2: DataBase structure: current GRA versus new GRA.

- The table *Cloud Location* is removed, since it is not used.

In terms of connections between the tables:

- The connection between *Threat Group* and *Vulnerability* is removed, to solve the inconsistencies of a group of threats containing a group of vulnerabilities.
- The connection between *Vulnerability* and *Risk Scenario* is established, in order to put in practice the definition of the risk. Since the risk is created only when a threat can exploit a vulnerability in a specific asset, the risk scenario should be created by the union between the threat (always present), the vulnerability (exploitable by the threat) and the asset (which affect the criticality of the solution and, accordingly, the impact of the risk). At this purpose the new connection between the two tables is established.
- The connections between *Answer*, *Option* and *Question* are established, in order to implement a conditional questions path, in which every question has a set of possible answers, and the option represents the answer chosen by the user for that possible question.
- The connections between *Vulnerability*, *Weight* and *Option* are established, to recreate the same parameters used in the calculation. Since in the likelihood

calculation, in fact, the weight of the questions in the *Step 2 - Questionnaire* of the analysis, affects the likelihood of the risk scenarios through the connected vulnerabilities, this connection was needed.

It is important to highlight that the table *Assessment* is not inserted in this database structure, since it is connected to each of these tables. Every time a new assessment is created, in fact, a snapshot is created accordingly, in order to avoid inconsistencies if the content of the lists is changed.

For example, if we consider a case in which an assessment is created, approved, and attached together with the documentation of a solution, if a user with *Admin* role changes the vulnerabilities contained in the vulnerability list, all the vulnerabilities already mitigated and approved for that solution will disappear and need to be mitigated again, even if the solution is already implemented.

For this reason a snapshot is created for every assessment, in order to take trace and make the analysis available and consistent at every time and despite every change.

5.2 Lists content improvements

Due to the several gaps in the lists (e.g., Vulnerability list, Threat Group list etc.), I reviewed their organization and content, creating a consistent collection, using different formal sources (e.g., ISO/IEC 27005:2022), to make the lists accurate and suitable for the needs of the company.

5.2.1 Vulnerability list

As described in the paragraph 3.1.4, the vulnerability list in the current version of the GRA tool, contains 66 vulnerabilities grouped into 18 categories which were never updated or reviewed in the last few years. In the perspective of the creation of a conditional questions path, to identify the vulnerabilities accurate as possible for the solution under analysis, I reviewed all the categories, and restructured them. Studying the CIS v8 framework 2.2.1 the organization of the categories and the way in which are taken into consideration, I created a set of 15 categories and relative sub-categories, suitable for containing and filtering in the best possible way a list of vulnerabilities, covering every aspect of a solution development/deployment, in a security perspective. When needed, a sub-category *Always applicable* contains all the vulnerabilities applicable to a solution for which the category is relevant, without the necessity of other filters (e.g., vulnerabilities as *Application of 4-eyes principle*). The decision of the structure in categories and sub-categories became relevant and necessary in parallel with the implementation of a conditional questions path, following different steps of filtering, in order to avoid the user answers to questions and, accordingly, to mitigate vulnerabilities, that are not relevant for the solution under analysis.

Process and sources

Starting from the current GRA, the process followed to prepare a plan and a complete list of vulnerability was:

1. Take care of the 18 categories and 66 vulnerabilities contained in the current list in the GRA tool. At the end of this step, clarifying, splitting, re-writing each vulnerability, I identified a total number of 117 vulnerabilities grouped in the 15 categories during the first review, and 119 vulnerabilities after the second one.

2. As second step, the collection of the gaps, comments about vagueness, and suggestions coming from the Cybersecurity team, brought me to identify a total number of 121 vulnerabilities, grouped in 18 categories.

3. At this point, the IRAM2 [2.2.1] lists of vulnerabilities, helped me identifying a total number of 131 vulnerabilities.

4. Also the Application Security Standard, defined by Hilti Cybersecurity team, was taken into consideration, taking the number of the vulnerabilities to 141.

5. Another important framework, the ISO/IEC 27005:2022 [2.2.1], helped me identifying at this step a total number of 152 vulnerabilities.

6. As last step, as used sources, the CIS v8 controls list [2.2.1]. Extracting from the list of controls 48 vulnerabilities to add to the previous 152, the structure and the organization of the categories inspired me, in identifying also a set of 15 categories and relative sub-categories.

At this point, collecting vulnerabilities from different sources, and adapting them to the organization perspective, a total number of 152 vulnerabilities grouped into 15 categories and sub-categories was reached.

Since part of the improvements of the tool deals with the user experience, and not only with the building of secure solutions, the cooperation with the Cybersecurity team, experimenting the usage of the tool during the real working time, became crucial also in the definition of the vulnerability list.

It led the team, in fact, to have a clear idea of the user needs, being aware of the importance to find a trade off between the number of the vulnerabilities covering all the aspects of security, and the information that a user could provide and/or the vulnerabilities that the user can mitigate and explain.

A flow of reviews was therefore consequent, by different members of Cybersecurity teams, bringing suggestions about the structure, the vulnerabilities inserted, and helping me in creating a final version of the vulnerability list, with 115 vulnerabilities grouped into 15 categories and relative sub-categories.

The new list

Below the new list of categories and sub-categories:

- General: this category covers the vulnerabilities dealing with general concepts/principles (e.g., Application of *4-eyes* principle, Application of *Security by design* principle etc.), applied to every solution. For this reason, it contains only the *Always applicable* sub-category.
- Endpoint: this category is relevant for solutions that involve endpoints (e.g., Mobile devices, Laptops, Servers etc.), and it contains the subcategories:
  - Always applicable: containing vulnerabilities applicable to every solution inside this category, as "Lack of endpoint inventory".
  - Device connectivity: applicable for solutions that involve devices, it contains vulnerabilities as "Remote wipe capability on Portable end-users devices are not enforced".
  - Remote access: applicable for solutions that require remote access, it contains vulnerabilities as "Insecure remote access (to Hilti environment/systems)".
- Hardware: this category is relevant for solutions that involve hardware components, and it contains:
  - Always applicable: containing vulnerabilities as "Lack of protection for physical access to hardware components";
  - Hardware managed by Hilti staff: including vulnerabilities as "Lack of monitoring of physical tampering".
- Software: this category is relevant for solutions that involve software components, with the filtering:
  - Always applicable: containing vulnerabilities as "Poor isolation of tiers, unseparated production and non-production systems".
  - Control on the code: relevant for solutions in which the user has control on the code, containing for example "Lack of code analysis and well-known flaws in the software".

- Network & Communication: category relevant for solutions that involve network infrastructure, it contains:
  - Always applicable: with for example "Inadequate network management and configuration"
  - Integration with other systems: with for example "Use of insecure communication protocols"
  - Network devices: containing for example "Insecure network device security configuration"
- Personnel developing/maintaining the solution: it contains only the *Always applicable* sub-category, since includes vulnerabilities regarding people involved in the solution, as "Employees have more privileges than required"
- Data: category that includes all the solutions that manage data, with the differentiation:
  - Data stored: with vulnerabilities as "Data at rest is not encrypted"
  - Data processed: with for example "Data processed are not segmented based on sensitivity"
  - Data transferred: containing, for example, "Lack of data encryption in transit"
- End users and account management: category that involves solutions that require end users, it contains:
  - Always applicable: with vulnerabilities as "Poor account management for end users"
  - Hilti standard authentication: with "Poor user authentication and credential management for end users"
  - Privileged accounts: for solutions that involve for example administrator accounts, it contains for example "Poor privilege account management"
- Technical users: it only contains one subcategory *Always applicable*, for solutions that require technical users, containing for example "Lack of authorization concept for technical users".
- Logs: only with the *Always applicable* category, with for example "Logs are not stored and collected appropriately"
- File management: for solutions that support files, it only has the *Always applicable* category with for example "Lack of anti-malware software deployment and maintainance"

- Service provider: only with the *Always applicable* category, containing vulnerabilities as "TOMs is not state of the art and approved by Cybersecurity team"
- Incident response: only with the *Always applicable* category, containing vulnerabilities as "Lack of response and continuity plan to incidents"
- Key/Secrets management: only with the *Always applicable* category, containing vulnerabilities as "Lack of security in key/secrets generation"
- Changes: only with the *Always applicable* category, containing vulnerabilities as "Lack of change management process"

5.2.2 Mitigation list

As described in the paragraph 3.1, one of the functionalities provided by the tool, is to give suggestions to the user in the way with which a vulnerability should be mitigated. The *Guiding questions*, in fact, have the purpose to help the user identifying the best way with which mitigate a vulnerability, avoiding the waste of time to implement a mitigation that is not enough secure.

Since part of the problems detected by the Cybersecurity team, were also *Guiding questions* related, after the process of the creation of the vulnerability list, I also took care of the mitigations list.

For each vulnerability, I identified and filled a field containing:

- Description of the vulnerability: to be less vague as possible, but improve the usability with short titles, a better clarification of the vulnerability meaning is given to the user through this field.
- Guiding questions: to help the user in the reflection about how the vulnerability should be mitigated, and how the explanation should be provided, a pair of questions is provided for each vulnerability.
- Mitigation suggestion: to better identify the best and more secure way to mitigate a vulnerability, a practical suggestion is provided for each of them.

For example, for the vulnerability "Lack of protection against reverse engineering", the following content is provided to the user:

- Description: The reverse engineering is the process of analyse functions and information flow, so the functionality and the behaviour of the solution can be understood. With this technique, the compiled code is accessible and can be decompiled.

- Guiding questions: Which measures do you implement to prohibit reverse engineering of your solution?
- Suggestion: Obfuscation

In order to have a reference in covering all the controls needed for each vulnerability, and to be aligned with the need of the company following the ISO/IEC 27002 [24], I accurately mapped each vulnerability and each category, and the relative mitigation, to the ISO controls indicated in the formal documentation.

After defined, for each vulnerability, the content of the *Description, Guiding Question, Mitigation Suggestion*, I mapped one or more ISO controls sections for each of them, or for an entire category, in order to have a clear and consistent view of all the aspects covered by the list, being sure to help the user as much as possible, in implementing solutions as secure as possible, mitigating the vulnerabilities in the easiest and more secure way, since, as we have seen described before, even if a threat exists, it doesn't represent a risk if there is no exploitable vulnerability.

5.2.3 Threat Groups - Threats - Risk Effects lists

In order to create a consistent list of components, I took into consideration also the *threats*, *risk scenarios* and *risk effects*.

Since this part is strictly related to the risk calculation, the content of the lists is still to be defined, but I created a plan of the sources to be used:

- Threat Groups: from IRAM2 classification [30]
- Threats: from IRAM2 classification [30]
- Risk Effects: from STRIDE classification [30]

5.3 Conditional Questions Path

As described in the paragraph 4, one of the gaps identified by the users was the way in which the vulnerabilities were attached to the solution in analysis. Due to the connection between the *Solution Deliverable Types* and the vulnerabilities, in fact, the choice of which vulnerabilities were relevant for the solution was made only according to the type identified.

For example, if during the creation of the assessment the type *SaaS* was inserted as Solution Type, all the vulnerabilities connected to the type were added as relevant to the solution.

The *Step 2 - Questionnaire*, in fact, was only related to the affection of the likelihood of the risk scenario connected to the vulnerability in scope, through the different weight of the answers.

In order to improve the user experience, and better filter the vulnerabilities according to the solution needs, I established a conditional questions path.

The idea behind this implementation, is to fill the gap through a set of questions strictly correlated to the vulnerability categories and sub-categories, in order to make the association between the solution and the vulnerabilities consistent, avoiding cases in which for example the user has to provide a possible mitigation for an authentication vulnerability, even if the solution to be implement is not related to the authentication.

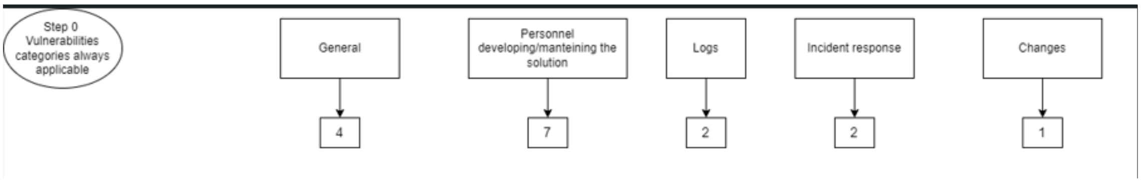


Figure 5.3: New GRA: Conditional Questions Path - Step 0.

The structure of the conditional questions path is structured in three different steps:

0. This step is related to all the categories containing *Always applicable* sub-categories only, which means that the vulnerabilities don't need additional questions to be added, but are relevant for every kind of solution.

   An example is the *General* category, that is applicable for every solution, without additional filtering needed, as shown in 5.3.

1. This first step is strictly related to the categories for which at least one question is needed. The purpose of this step is to categorize the solution with a first question, leading to all the questions needed to filter the vulnerabilities according to the solution.

   For example, the first question "Does your solution involve endpoints?" lead the user to avoid all the consequent questions and related vulnerabilities about the endpoint security, if the solution doesn't involve endpoints. In this case, therefore, the user will answer "No" to this question, and no more questions or vulnerabilities related to user endpoints will show up. If the user instead will answer "Yes", more questions dealing with this category will show up. The first step, together with the second one, is represented in 5.4, 5.5, 5.6.

2. The second step contains all the questions shown according to a positive answer to a question of the Step 1, and are strictly correlated to the concept of the vulnerabilities sub-categories. In this way, another level of filtering will

lead the user to have only vulnerabilities really related to the solution to be mitigated.

For example, if the user answered "Yes" to the question "Does your solution involve endpoints?" of the Step 1, two more questions will show up in this case: "Does your solution involve devices?" and "Does your solution require remote access?". If the user will answer "Yes" to the first one, and "No" to the second one, only the vulnerabilities contained in the sub-category "Device connectivity" plus the vulnerabilities in the "Always applicable" sub-category will show up, without vulnerabilities dealing with the remote access.

This means that if the solution involves endpoints, but it doesn't involve device connectivity and remote access, the user has to mitigate a total number of 4 vulnerabilities for this category, instead of a total number of 11 vulnerabilities (including the vulnerabilities related to device connectivity and remote access).

In some cases, the answer "No" of a question of a Step 2, doesn't mean having a total number 0 of vulnerabilities for that sub-category, but it means that the user will have more or less vulnerabilities, different from the ones that she/he would have answering "Yes", related to that answer.

For example, if we consider the category *Software*, as first question of Step 1 the user has to answer to "Does your solution involve software components?". If the answer is "Yes", the other question "Do you have control on the code?" will show up. If the user answers "Yes" a total number of 19 vulnerabilities are attached to the solution (11 deriving from the question, 8 deriving from the *Always applicable* sub-category). On the contrary, if the user answers "No", a total number of 11 vulnerabilities will be shown for the solution (3 deriving from the question, 8 from the *Always applicable* sub-category).

In order to keep the same risk calculation, the parameters have to be preserved, and for this reason, another layer of questions is needed.

The questions previously provided to the user during the *Step 2 - Questionnaire* of the analysis, in fact, are directly connected to the vulnerabilities, in order to affect the likelihood calculation of the risk scenarios related to the vulnerabilities, based on the different weight of the answers.

In order to keep this likelihood calculation, another the layer of the questions affecting the likelihood, and not the choice of the vulnerabilities to be mitigated for that solution, is added.

Example of this questions and possible answers are:

- Number of end users

    1. No end users (Weight: 0)
    2. Few internal users (Weight: 1)

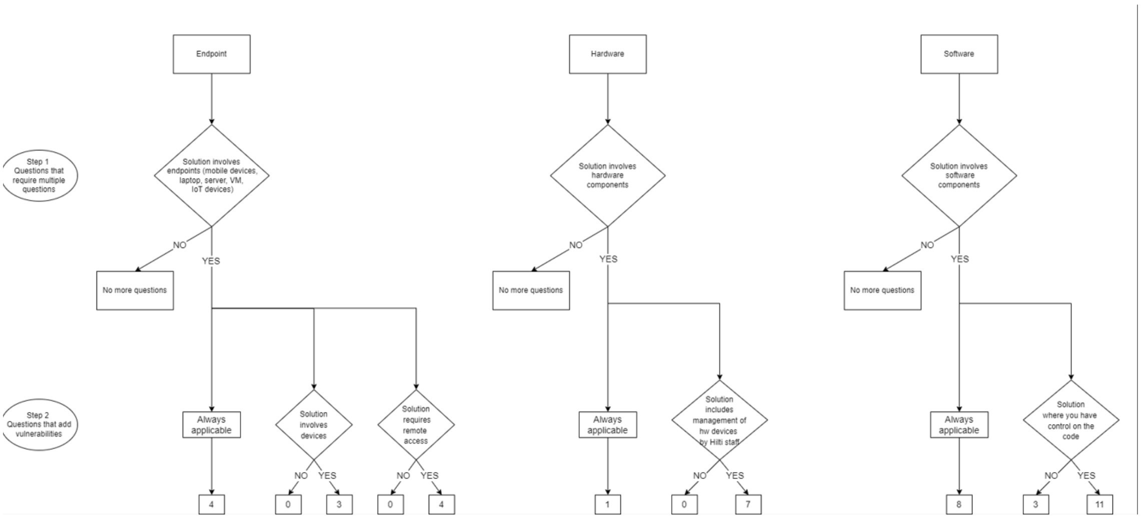

Endpoint
Hardware
Software
Step 1 Questions that require multiple questions
Solution involves endpoints (mobile devices, laptop, server, VM, IoT devices)
Solution involves hardware components
Solution involves software components
NO
YES
No more questions
Step 2 Questions that add vulnerabilities
Always applicable
Solution involves devices
Solution requires remote access
Solution includes management of hw devices by Hilti staff
Solution where you have control on the code
NO
YES
4
0
3
0
4
1
0
7
8
3
11

Figure 5.4: New GRA: Conditional Questions Path - Steps 1 and 2 - Part 1.

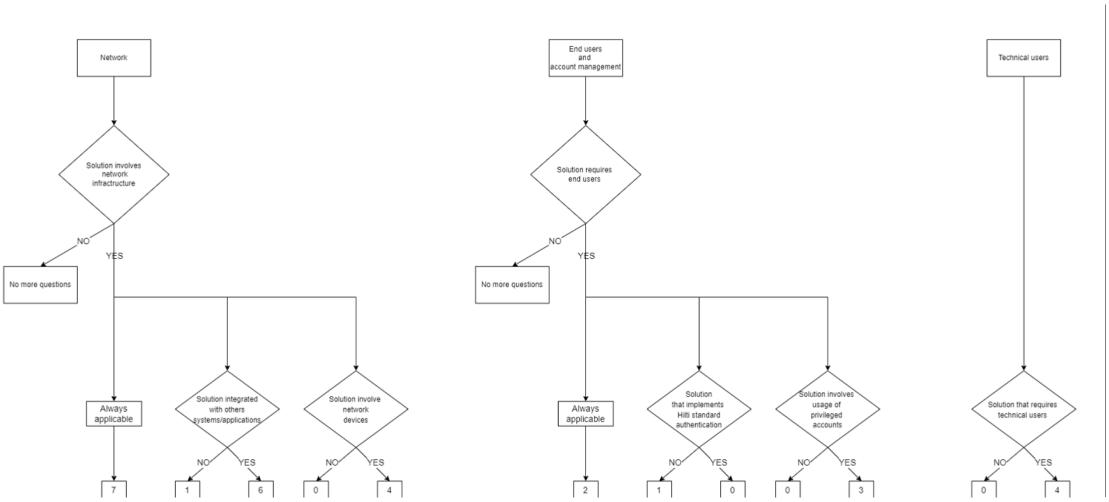

Network
End users and account management
Technical users
Solution involves network infractructure
Solution requires end users
NO
YES
No more questions
Always applicable
Solution integrated with others systems/applications
Solution involve network devices
Solution that implements Hilti standard authentication
Solution involves usage of privileged accounts
Solution that requires technical users
NO
YES
7
1
6
0
4
2
1
0
0
3
0
4

Figure 5.5: New GRA: Conditional Questions Path - Steps 1 and 2 - Part 2.

3. Several internal users (Weight: 2)
4. Many internal users (Weight: 3)
5. External users (Weight: 4)

At the end of this description is important to highlight that, despite the effort spent in analyse the tool, the functionalities, the steps of the analysis, and the questions provided to the user, no documentation was found to clarify the choice

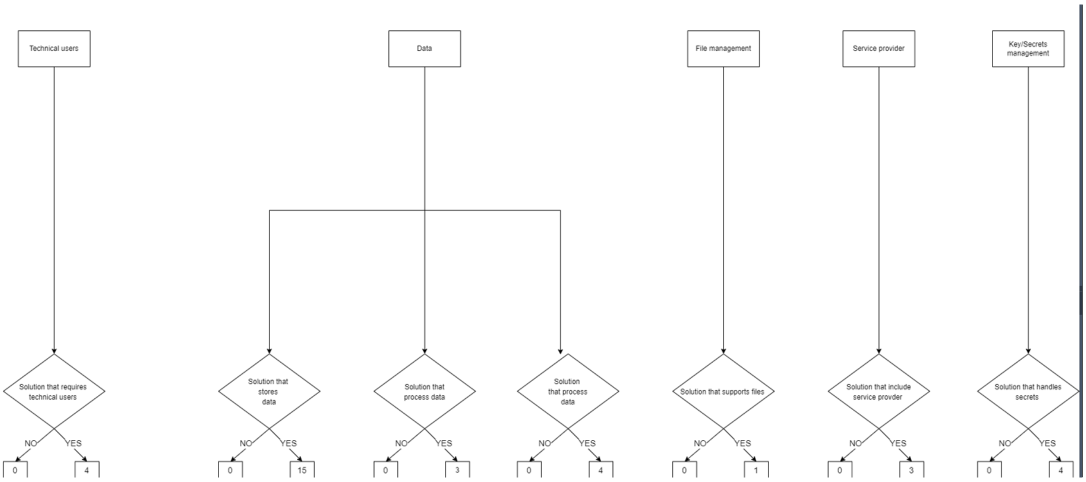

Technical users
Data
File management
Service provider
Key/Secrets management
Solution that requires technical users
Solution that stores data
Solution that process data
Solution that process data
Solution that supports files
Solution that include service provder
Solution that handles secrets
NO
YES
0
4
0
15
0
3
0
4
0
1
0
3
0
4

Figure 5.6: New GRA: Conditional Questions Path - Steps 1 and 2 - Part 3.

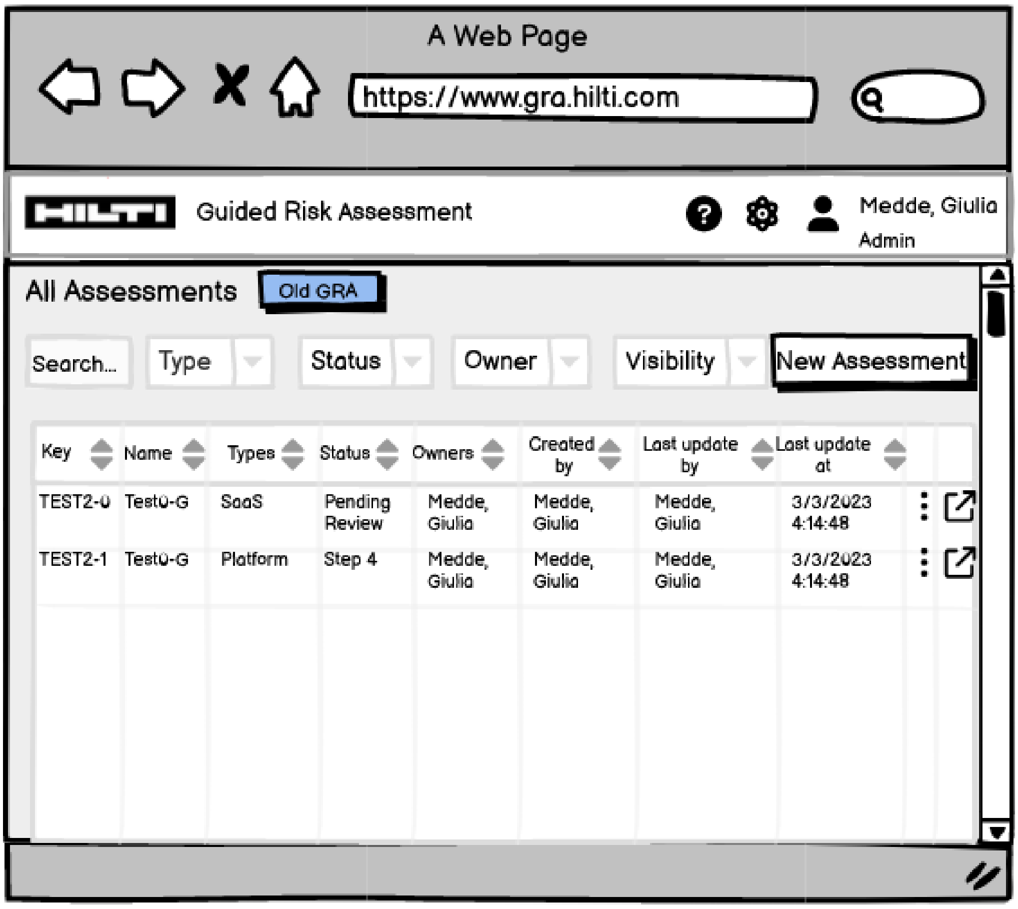
behind the weights for the questions in the *Questionnaire*, and since the risk calculation will be reviewed later on, I was careful in implementing the improvements to make them adaptable to the current risk calculation (to make possible continuing using the tool), and the future new computation.

5.4 Wireframes

Since all the improvements described in this chapter require changes and implementation in coding the application, part of this thesis work should have done in code perspective. Due to delays in gain softwares installation and privileges, which made me aware also about the negative aspects of working inside a big organization, I created wireframes to clarify how the application should appear after the real implementation of the improvements. These subsections below will show the changes in the User Interface, but the other changes in the code will come accordingly (e.g., Create the new DB, Retrieve the lists from the new DB etc.).

5.4.1 Homepage

The homepage contains the list of the assessments created over time. Since the way in which the assessments will show will be different, in order to be consistent, the solution identified is a button that lead the user to switch between the assessments created in the old version of the tool, and the assessments created with the new version. As you can see in 5.7, the *Old GRA* button, already selected, has been placed on the top of the filters bar, for space reasons, but also to be suddenly visible to the user.



Key	Name	Types	Status	Owners	Created by	Last update by	Last update at
TEST2-0	Test0-G	SaaS	Pending Review	Medde, Giulia	Medde, Giulia	Medde, Giulia	3/3/2023 4:14:48
TEST2-1	Test0-G	Platform	Step 4	Medde, Giulia	Medde, Giulia	Medde, Giulia	3/3/2023 4:14:48

Figure 5.7: Homepage view of the new version of the tool.

5.4.2 New Assessment page

The pop-up for the creation of a new assessment, in the current version, contains the list of the Solution Types that the user *Admin* can choose for the solution in analysis. Since the concept of the types is removed in the new version of the tool, the pop-up has to change accordingly, as you can see in 5.8.

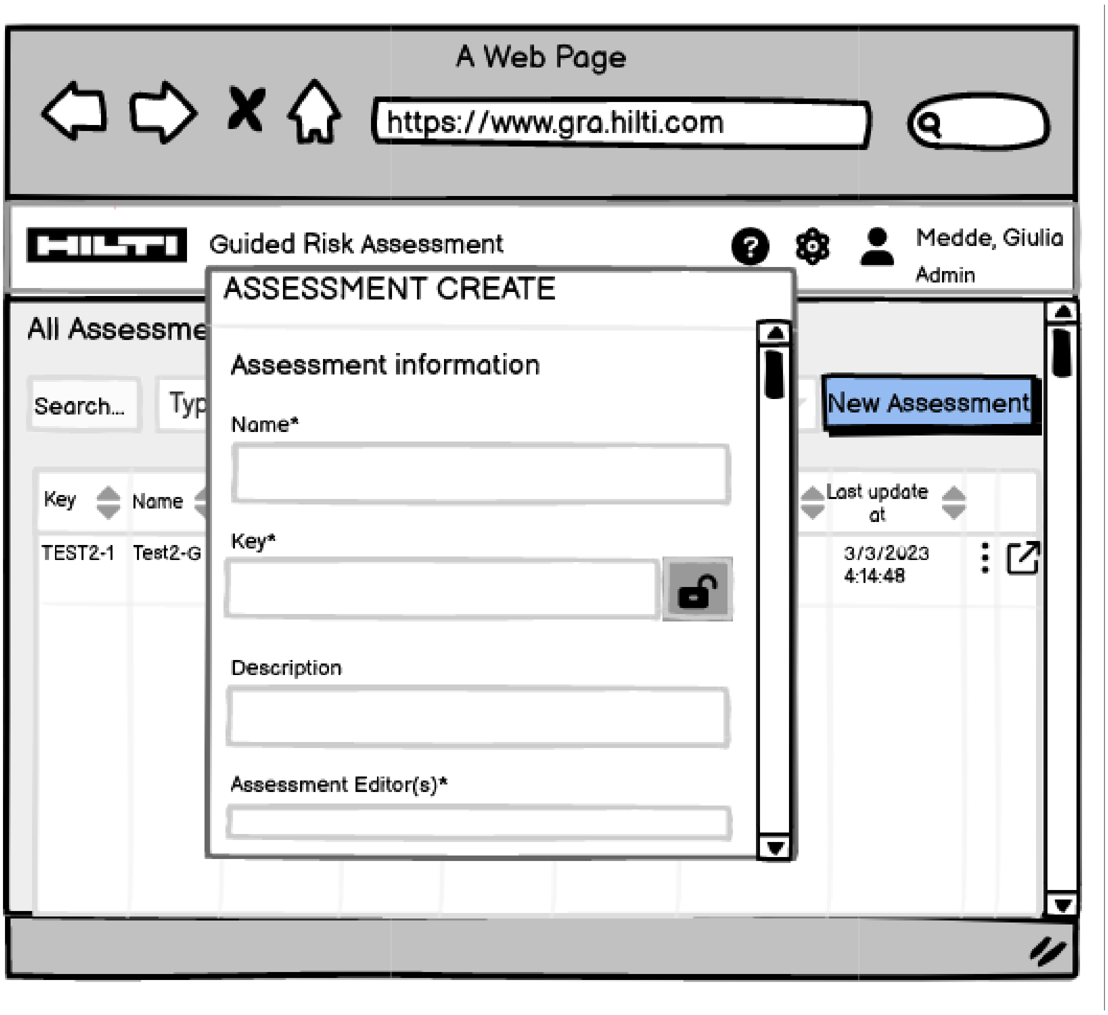


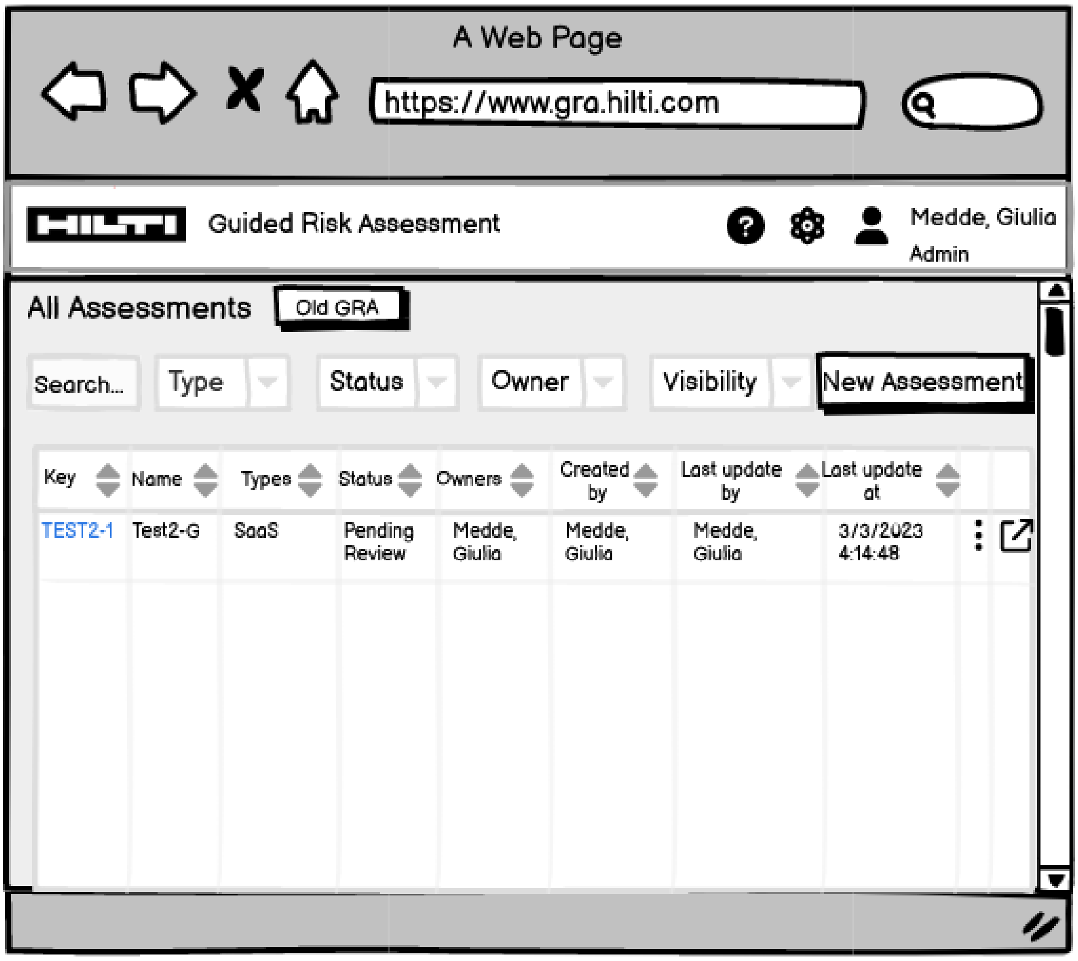
Figure 5.8: New assessment pop-up in the new version of the tool.

5.4.3 Assessment page

The change in the strategy of the analysis, requires the entire process to be changed, and the visualization of the steps accordingly.

As you can see in 5.9, a selection of an assessment from the homepage is required, to be able to open the correct assessment and provide the information needed. At this point, the different steps will be shown to the user, and they will require the following changes, as you can notice in 5.10:

- In the *Step 1 - General Information*, the field listing the types chosen for the solution has to be removed.

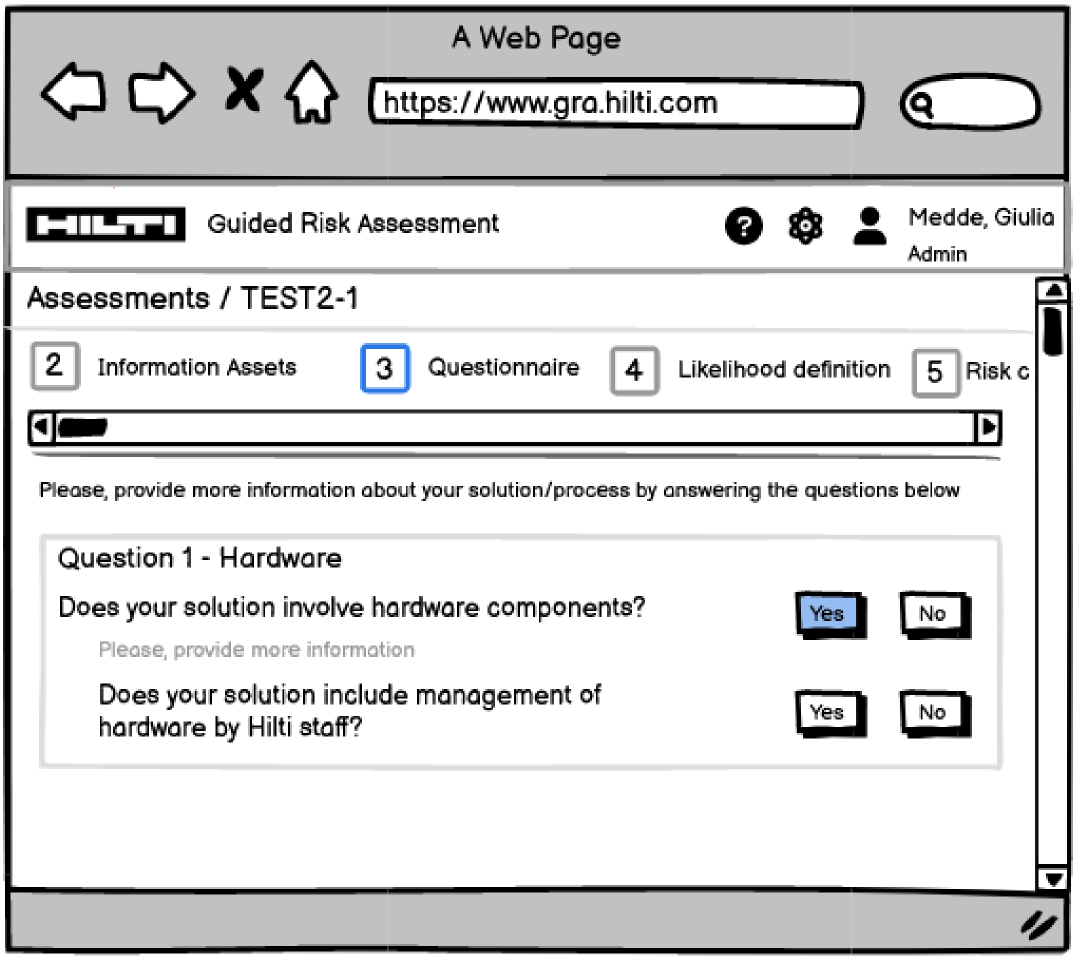


Key	Name	Types	Status	Owners	Created by	Last update by	Last update at
TEST2-1	Test2-G	SaaS	Pending Review	Medde, Giulia	Medde, Giulia	Medde, Giulia	3/3/2023 4:14:48

Figure 5.9: Selection of an assessment before provide the information.

- Between the steps 2 and 3, a new step has to be added, in order to implement the *Conditional Questions Path* concept. For this reason, the *Step 3 - Questionnaire* in the current GRA, becomes the *Step 4*.
- The new step related to the *Conditional Question Path* will keep the same name of the current *Questionnaire*, but it will contains the questions related to the association of vulnerabilities. The current step *Questionnaire*, will become the *Likelihood definition*, since it will contain the same questions as now, that affect the likelihood.
- The Step 4 - Vulnerabilities of the current GRA, need to become the Step 5, due to the insertion of the *Conditional Questions Path* step. It needs to

be adapted also to the removal of the types, deleting the field shown this information.



Please, provide more information about your solution/process by answering the questions below

Question 1 - Hardware

Does your solution involve hardware components? Yes No

Please, provide more information

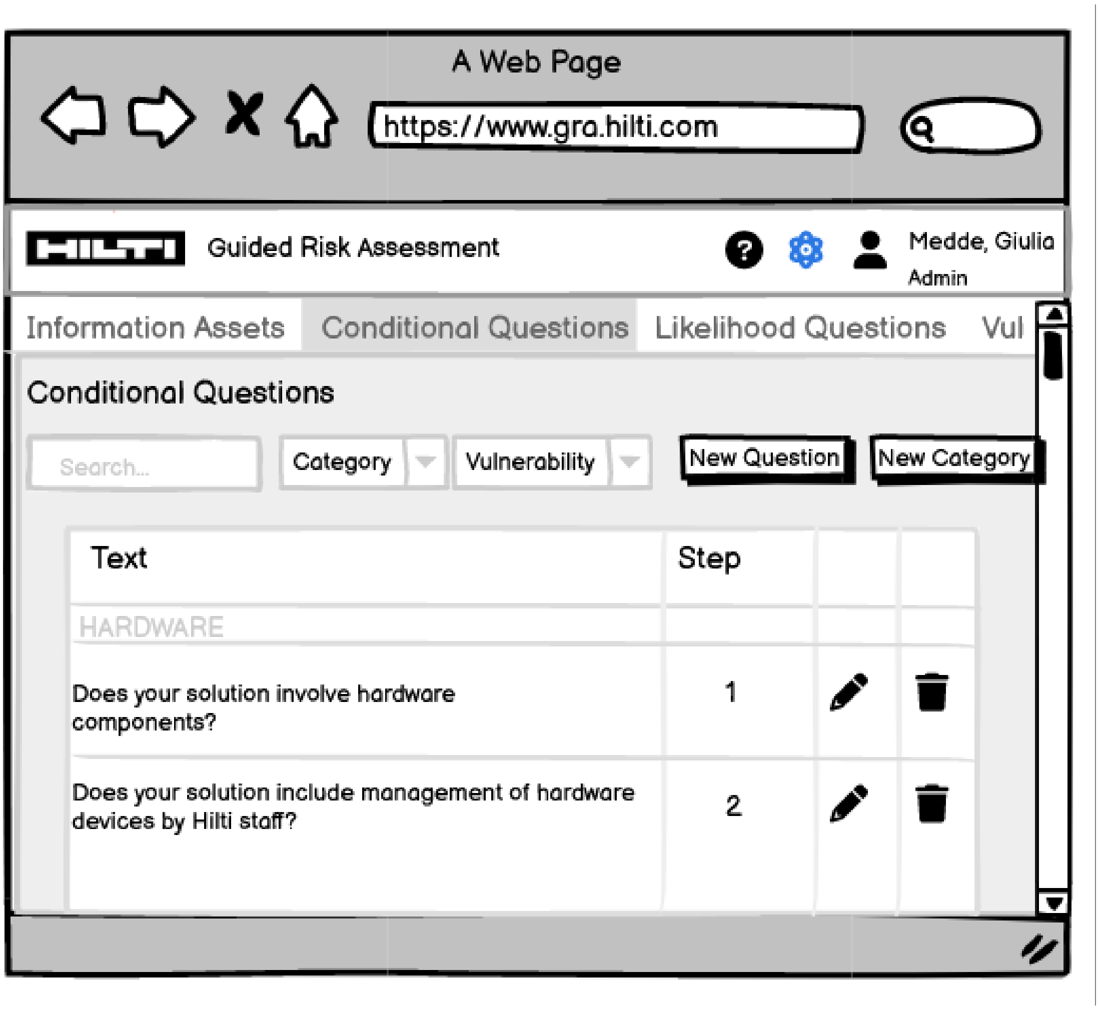
Does your solution include management of hardware by Hilti staff? Yes No

Figure 5.10: New step in providing the information.

5.4.4 Admin panel

According to the changes previously described, also the *Admin* panel need to be changed accordingly.

In particular, a new section to manage the *Conditional Questions Path* concept is needed, as shown in 5.11, and the section managing the types need to be removed. If in the visualization of the questions previously part of the *Step 3 - Questionnaire*, also the answers are shown, in the new section this information is omitted, since the answers are only "Yes" or "No" for each question.



Conditional Questions

Text	Step		
HARDWARE			
Does your solution involve hardware components?	1	✎	🗑
Does your solution include management of hardware devices by Hilti staff?	2	✎	🗑

Figure 5.11: Admin panel in the new version of the tool.

5.5 Documentation

The lack of documentation of the current implementation of the GRA tool, brought me to follow a process of in-depth analysis, including reverse engineering and code analysis, to understand not only the gaps and inconsistencies, but also the functionalities provided by the tool.

In order to make the tool understandable to the Cybersecurity team, and the users in general, as part of this thesis work I also wrote the documentation, explaining the results of the researches, how the risk is computed, which are the connections between the components, the db description etc.

To let the users really use the tool, and experiment the improvements described in this chapter, a consequent implementation is needed and planned, leading to the future improvements planned, described in the next chapter.

Chapter 6

Future improvements of the GRA tool

The previous chapters give the necessary information to learn how to use the tool, which are the gaps and inconsistencies in the process of the risk assessment of a solution, and which techniques and frameworks I studied and applied in order to fill the gaps and improve the tool, both in the perspective of user experience and especially to allow and support solution architects and the other colleagues in building solutions as secure as possible within the organization.

6.1 New Threats and Risk Scenarios list

According to the new DB structure, described in 5.1, I already collected the new lists of vulnerabilities and mitigations, but the new setup also requires to have, in parallel, a new list of threats and risk scenarios pre-defined, to make effective the new connections between the components.

A consistent collection of the possible threats that can affect a solution, in fact, will give the tool the opportunity to be as accurate as possible in the evaluation of the risk, and for this reason the plan to use the IRAM2 framework [30] with relative tables, can lead to create a list of threats applicable in the real world, with a categorization that highlights the main areas of interest.

According to the threats and vulnerabilities lists, and to the definition of the risk, a crucial importance will have the pre-defined lists of the risk-scenarios. Representing the combination between a threat and an exploitable vulnerability, the simple join between the two tables can lead to have a combinatorially high set of risk scenarios, not only difficult to handle and solve, but also not enough accurate and suitable for the organization needs. The risk, in fact, identifying a combination between threat and vulnerabilities in general, is to create scenarios in which threats can be combined to a vulnerability not exploitable using that threat (e.g., an environmental threat combined with a vulnerability as "Lack of logs monitoring" leads to an unlikely scenario, increasing the risk that is not really effective, and requiring a lot of effort that can be avoided, while the same threat exploiting a vulnerability as "Lack of redundancy of hardware" increases the lack of the availability principle).

For this reason, I considered necessary not only a work in listing all the possible realistic threats, but also a review of each possible combination of the two components, reducing the list to be more realistic, applicable, and suitable within the organization set of risk scenarios, for which decreasing the likelihood is crucial for the security of the company, optimizing the effort spent in mitigating the related vulnerabilities.

6.2 New Risk Effect list

The categorization of the risk scenarios in the risk effects, becomes also critical, in identifying the minimum set of consequences which the company can meet, not only to improve the calculation of the impact, but also to help the users (supposed to be not security experts) to really understand and imagine a realistic consequence of the lack of an accurate risk assessment process, leading them to spend as much effort as possible in providing correct information about the solution and in identifying the best mitigation existent.

The participation of aware users in this process, is indeed the primary goal to achieve, helping them to not be forced to follow a procedure only because they have to do it, but providing them all the elements to understand the consequences for the company, for the solution, for the users themselves, if this process was not set up in the most efficient way.

The presence of an heatmap, described in 3.1, is a powerful way to give a user an overview of the condition of the solution to be implemented, being aware to recognize how dangerous it can be in a Cybersecurity perspective, and for this reason the representation of the risk effects (categories of risk scenarios, simpler, described with less technicalities) can improve significantly this experience. Assuming that part of the users actively using the tool do not have knowledge in this field, the visualization of consequences that they can imagine, for which they can deeply understand the damages, will give them the awareness and the motivation to follow this process proactively, being interested in make the solution secure not because they are forced by the company to do it, but because they can visualize the consequences and they want to avoid the solution to be responsible for that.

6.3 Other Improvements

If, on one side, the attention to the people using the tool to assess their solution is crucial, on the other side is also fundamental the experience of the users who use the tool with the *Admin* role, in this case the Cybersecurity team at Hilti.

Facing with the usage of this tool every day, providing support to the other departments, the best implementation as possible can help them to significantly improve the quality of their work, avoiding to spend a lot of effort in filling the gaps and solving the problems in the tool representation and/or calculation.

For this reason, I put in place several improvements, but a lot of effort should be spent implementing even more, for example:

- Improve the process of duplication of an assessment, providing a visualization in which the admin can choose and decide which vulnerabilities can maintain the mitigation status, and which ones need to be reviewed;

- Improve the process of integration of the assessments, providing the possibility to choose, by the admin, which assessments need to be merged, and which information can be considered already filled, instead of duplicate only one already existent and manually merge the information deriving from the other assessments to be integrated;

- Improve the comments in the vulnerabilities, providing an efficient comments mechanism through which the admins can take trace of all the problems in the mitigation descriptions.

Since the meaning of the Guided Risk Assessment tool is based on the calculation of the risk, the complicated process described in 3.2 makes it difficult to the Cybersecurity team in really understanding it. The improvement of the calculation is therefore necessary, in order to make it as simple and clear as possible, to help the Cybersecurity team gaining the knowledge of it and to improve consequently their job, providing robust and clear support to the other users in understanding the situation in risk perspective of the solution, and the best changes to be applied to minimize the risk.

It is challenging, in fact, supporting the users and helping them to minimize the risk of the solutions to be implemented, without the knowledge of which information affects the calculation, since not only the capacity in putting in place controls is needed, but also the understanding of which parameters really affect the calculation, in order to decrease the risk.

If an agile but consistent plan need to be followed, to make all the improvements as much efficient as possible, providing first all the theoretically elements to be suitable to the needs of the users and the company, a real implementation phase should follow, to make the improvements effective.

As shown in the previous chapter 5.4, a set of wireframes has been created, in order to visualize how the improvements in the strategy and structure, and in vulnerability perspective, can be effective, but a consequent implementation with coding is necessary.

The analysis of the future improvements, in comparison with the start point of this thesis work, make clear the importance of this work not only in deeply understanding the tool, identifying the gaps and inconsistencies and putting in place improvements, but also in planning which the next steps can be, to help the organization in speeding up the process of the risk assessment, building secure solutions in the easier and faster way.

Chapter 7

Conclusion

At the end of this thesis work, a better understanding of the functionalities of the Guided Risk Assessment tool brought me to be able to identify not only the gaps and inconsistencies in the surface, but also the root cause of them, leading me to implement a consistent and robust strategy and structure, applying the formal definitions of the components (e.g., the risk equation).

The consciousness that I gained about the real usage of the tool within the organization, together with the several touchpoints with the Cybersecurity team and the stakeholders involved, helped me in the challenge of finding trade-off and compromises between the security needs and the user experience, being able to implement a consistent conditional questions path that covers the security aspects, but at the same time in the clearest and easiest way for people who don't have a knowledge in this field.

The review of the vulnerabilities already present, helped me being aware of the limit and the problems during an assessment of a solution, inspiring me in studying the different formal sources (e.g., ISO documentation, IRAM2) and in applying them for the purpose required by the GRA tool.

The creation of a new vulnerability list with a new structure and content, compliant with the Cybersecurity strategy at Hilti, made me able to face the challenge in identifying a reasonable and consistent set through all the several sets of vulnerabilities collected, in parallel with the conditional questions path, constructing an analysis process capable of assess the risk of the solutions within the organization in a conscious way.

The possibility to cooperate with the members of the Cybersecurity team in a big company as Hilti gave me the opportunity to reflect on the importance of the GRA's purpose, and to face with different needs and points of view, inspiring me in making the functionalities clear, the research of the gaps accurate and the improvements secure and usable as possible.

Combining my knowledge with an agile but clear plan, and comparing the initial assumptions with the goal achieved, this thesis work made me build a conscious perspective of the way in which the Cybersecurity is applied inside of a global manufacturing company, and how the importance of this field is growing and improving over time.

Bibliography

[1] ISO. *ISO/IEC 27001*. URL: https://www.iso.org/isoiec-27001-information-security.html (cit. on pp. 3, 4).

[2] NIST. *NIST SP 800-16 - Information Technology Security Training Requirements: a Role- and Performance-Based Model* (cit. on pp. 3, 27).

[3] ISO. *ISO/IEC 27000:2014*. URL: https://www.iso.org/obp/ui/#iso:std:iso-iec:27000:ed-3:v1:en (cit. on p. 4).

[4] OWASP Foundation. *Threat Modeling*. URL: https://owasp.org/www-community/Threat_Modeling (cit. on pp. 4, 5, 9).

[5] Nataliya Shevchenko - Carnegie Mellon University. *Threat Modeling: 12 Available Methods* (cit. on pp. 5, 9).

[6] Andrew Sodergerg Rivkin - University of Gothenburg Sathya Prakash Kadhirvelan. *Threat Modelling and Risk Assessment Within Vehicular Systems* (cit. on pp. 5, 20).

[7] Christopher J. Alberts, Audrey J. Dorofee, James F Stevens, and Carol Woody - Carnegie Mellon University. *Introduction to the OCTAVE Approach*. 2003 (cit. on p. 5).

[8] CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST. *Introduction to the OCTAVE Approach*. URL: https://apps.dtic.mil/sti/citations/ADA634134 (cit. on pp. 5, 6).

[9] Christopher J. Alberts, Sandra Behrens, Richard D. Pethia, and William R. Wilson - Carnegie Mellon University. *Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) Framework* (cit. on p. 7).

[10] VerSprite Tony UcedaVelez - OWASP. *"Real World Threat Modeling Using the PASTA Methodology"*. OWASP, VerSprite, 2012 (cit. on p. 7).

[11] GitLab. *Threat Modeling - Threat Modeling Within GitLab*. URL: https://about.gitlab.com/handbook/security/threat_modeling/#threat-modeling-within-gitlab (cit. on p. 7).

[12] VerSprite. *PASTA methodology* (cit. on p. 8).

[13] OWASP Foundation. *STRIDE Reference Sheets* (cit. on p. 9).

[14] EC-Council. *What is Cyber Threat Intelligence | Become a Threat Intelligence Analyst* (cit. on pp. 10–12).

[15] Paul Saitta, Brenda Larcom, and Michael - OctoTrike Eddington. *Trike v.1 Methodology Document.* OctoTrike, 2005 (cit. on p. 11).

[16] Threat Modeler. *Threat Modeling Methodologies: What is VAST?* URL: `www.threatmodeler.com` (cit. on p. 12).

[17] OWASP Foundation. *OWASP Threat Dragon* (cit. on p. 13).

[18] Security Boulevard. *Threat Modeling Application Released By OWASP: Threat Dragon 1.0* (cit. on p. 15).

[19] Microsoft. *Getting started with the Threat Modeling Tool.* 2022 (cit. on pp. 14, 16).

[20] MITRE. *MITRE ATT&CK* (cit. on p. 16).

[21] Exabeam. *What is MITRE ATT&CK: An Explainer* (cit. on p. 18).

[22] Agile Stationery. *Elevation of Privilege (EoP) Threat Modeling Game.* URL: `www.agilestationerty.com` (cit. on p. 19).

[23] NIST. *NIST Risk Management Framework* (cit. on pp. 20, 21).

[24] Muhamad Al Fikri, Fandi Aditya Putra, Yohan Suryanto, and Kalamullah Ramli. *Risk Assessment Using NIST SP 800-30 Revision 1 and ISO 27005 Combination Technique in Profit-Based Organization* (cit. on pp. 20, 80).

[25] Telos Corporation. *NIST RMF Automation | Risk Management Framework | Xacta* (cit. on p. 22).

[26] ISO. *ISO/IEC 27005:2022* (cit. on pp. 21, 23, 46).

[27] Institute FAIR. *The Importance and Effectiveness of Cyber Risk Quantification* (cit. on pp. 22, 23).

[28] CIS. *CIS Controls Version 8* (cit. on p. 24).

[29] CIS. *CIS Critical Security Controls Implementation Groups* (cit. on p. 25).

[30] Information Security Forum. *Information Risk Assessment Methodology 2 (IRAM2).* 2014 (cit. on pp. 24, 26, 80, 91).

[31] Carnegie Mellon University CMU. *Nessus : A security vulnerability scanning tool* (cit. on p. 27).

[32] HILTI AG. *IT Risk Management Practitioner Guide.* Hilti AG, 2016 (cit. on p. 47).