SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Laurea Magistrale in Cybersecurity

Thesis for the Master's Degree in Cybersecurity

Supporting the Italian National Cybersecurity Framework with Security Governance Data Flow Diagrams

Thesis Advisor
Prof. Angelini Marco

Candidate
Mignano Fulvio Pio
1810986

Academic Year MMXXI-MMXXII

Abstract

Information Security Governance includes a variety of processes to manage security aspects of the organization. Those processes are, in many cases, conducted at a very high level of abstraction, manually, by managers and security experts. However the lack of automation and integration with technical data collected from processes at a lower level of abstraction represent a limitation, as human errors, inexperience and bias can alter the results and potentially leave the organization vulnerable.

This thesis introduces a first specification for the *Governance Data Flow Diagram (GDFD)*, an extended version of the regular *Data Flow Diagram (DFD)*, designed to support Information Security Governance processes. In particular the goal behind the definition of *GDFD*s is to enable those processes to exist at a "middle" level of abstraction, not as technical as attack graphs or penetration testing, but neither as abstract as verifying compliance to standards with questionnaires.

The second part of this thesis shows how *GDFD*s can be leveraged to support the Italian National Framework for Cybersecurity and Data Protection. Firstly each subcategory of the Framework that allows for it, is assigned a Security Template, which is a pre-made *GDFD* describing a possible implementation to reach the security objectives defined by the subcategory.

This is followed by the presentation of the controlExtractor tool, which is capable of processing those Security Templates to extract automatically a set of controls, with the objective of supporting and complementing controls which are instead manually defined by security experts.

The output of the controlExtractor tool is evaluated and compared with the manually created controls found in the *General Data Protection Regulation (GDPR)* contextualization prototype provided by the Italian National Framework for Cybersecurity and Data Protection as well as the controls found in a real cybersecurity assessment performed by *Research Center of Cyber Intelligence and Information Security (CIS)* Sapienza members.

The comparison shows that the automatically extracted and manually defined controls are indeed complementary in nature under several aspects. It is thus feasible to combine the use of those two kinds of controls, having the automatically extracted ones support those manually crafted by humans, leading to more objective and comprehensive security assessments when using the Framework.

Keywords: Data-Driven, Security, Governance, Framework, Data Flow Diagrams

Contents

List of Figures

List of Tables

Acronyms

APOA Assessed Proportion of Affect

CINI Consorzio Interuniversitario Nazionale per l'Informatica

CIS Research Center of Cyber Intelligence and Information Security

CTI Cyber Threat Intelligence

DFD Data Flow Diagram

EDFD Extended Data Flow Diagram

EPC Error Producing Condition

GDFD Governance Data Flow Diagram

GDPR General Data Protection Regulation

GTT General Task Type

HEART Human Error Assessment and Reduction Technique

HEART-IS Human Error Assessment and Reduction Technique - Information Security

HRA Human Reliability Assessment

JSON JavaScript Object Notation

MTMT Microsoft Threat Modeling Tool

NIST National Institute of Standards and Technology

STIX Structured Threat Information eXpression

TAXII Trusted Automated eXchange of Intelligence Information

Chapter 1

Introduction

Information Security Governance is very important for organizations using information and communication technology systems, especially those with a large and complex structure, being a key component for a good security posture. However it is not easy to provide a formal definition for it. One possible way to define Information Security Governance is contained in the *National Institute of Standards and Technology (NIST)* Special Publication 800-100, which states "Information security governance can be defined as the process of establishing and maintaining a framework and supporting management structure and processes to provide assurance that information security strategies are aligned with and support business objectives, are consistent with applicable laws and regulations through adherence to policies and internal controls, and provide assignment of responsibility, all in an effort to manage risk" [9].

A simple way to model the processes involved in Information Security Governance was proposed by Von Solms [47]. This model is based on the direct-control cycle and divided into two parts: the core and the extended part(Figure 1.1). The core part is represented on the "face" of the model and shows the direct-control cycle across the three main organizational levels: strategic level, tactical level and operational level. Directives flow from the strategic level to the operational level, getting more technical along the way, until they are executed. The control process collects data from the execution and refines it in progressively more abstract reports, returning to the strategic level and completing the cycle informing the creation of new directives. The extended part is represented on the "depth" dimension of the model. It includes a series of transversal topics needed at all the levels of the organization which should to be properly addressed for a good Information Security Governance process.

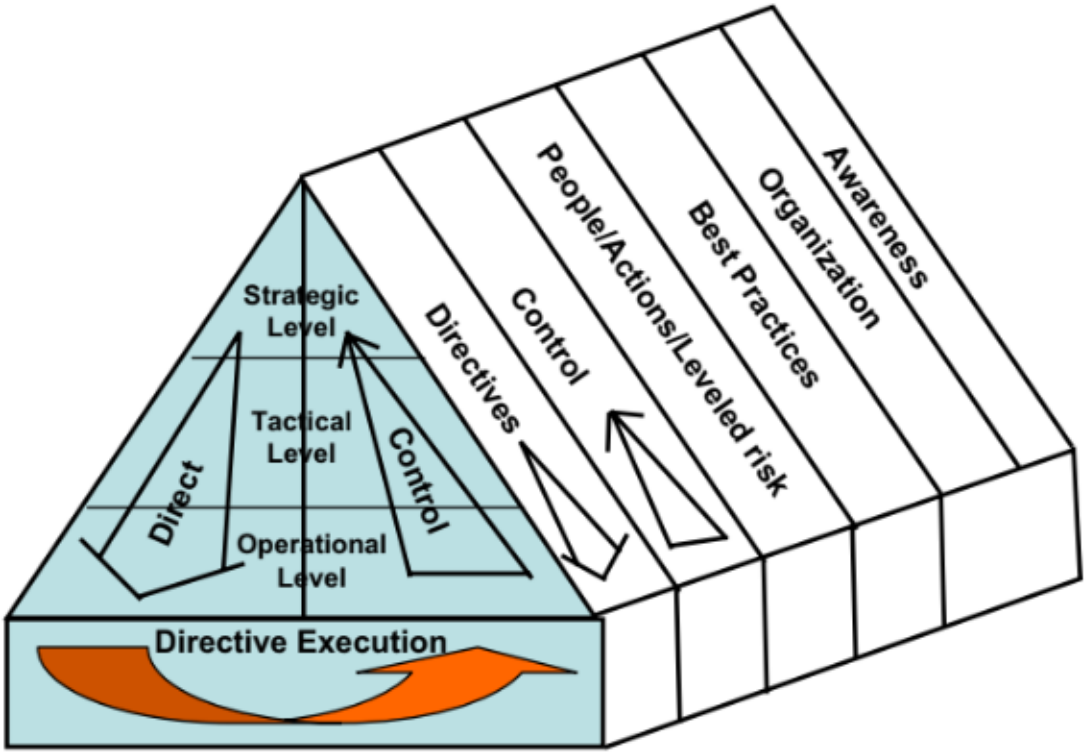


Figure 1.1: Von Solms's Information Security Governance Model

When it comes to actually implementing security governance processes in the context of an organization, there are many frameworks and standards that can help guide the implementation and then support and improve established processes. Usually standards are composed of several documents, each of which has a particular purpose. However, those documents can be widely different depending on the perspective which is taken in consideration by that standard: some documents provide only suggestions while others contain strict compliance rules required for getting a certification. Of particular importance are cybersecurity frameworks, documents designed to provide support to organizations wanting to manage and reduce cybersecurity risks.

In the United States. *NIST* is one of the leading parties in the field of security standards and publishes many documents called "Special Publications" which are identified by numerical codes. In 2014 *NIST* released the first version of their Cybersecurity Framework [1][39], which then received an update version 1.1 four years later[7]. Both versions are organized in categories, broad outcomes that should be met for achieving an effective cybersecurity posture. Categories are grouped according to five functions which correspond to basic cybersecurity activities: Identify, Protect, Detect, Respond and Recover. Moreover each category is divided into several subcategories which provide more specific outcomes and activities. Every subcategory also has a set of associated informative references, pointers to specific sections of other documents containing guidelines, standards and best practices which can assist in achieving what is stated in the subcategory.

In 2015 Sapienza University of Rome, in collaboration with the *Consorzio Interuniversitario*

[1] `https://www.nist.gov/cyberframework/framework`

Nazionale per l'Informatica (CINI) Cyber Security National Lab, developed an Italian Cybersecurity Framework called "Framework Nazionale per la Cybersecurity"[2][5]. The framework was revised in 2019 when it received its version 2.0 and renamed in "Italian National Framework for Cybersecurity and Data Protection"[4]. Most of the additions introduced in the new version are related to privacy and data protection and made the Framework compatible with the European *GDPR*. The Framework, in both its versions, is inspired by the *NIST* Cybersecurity Framework and builds on it to better fit the Italian environment. However the basic structure of the framework is the same as *NIST*, sharing the same organization in Functions, Categories and Subcategories.

Both the aforementioned frameworks operate at a very high level of abstraction and compliance is commonly verified resorting to interviews and questionnaires asking for each category and subcategory. This high-level approach is however not always reliable for a few evident reasons. Firstly by simply asking for compliance in an abstract way it is easy to overlook even simple technical problems which would be easily spotted by a deeper analysis. For example take in consideration the *NIST* subcategory PR.AC-3 "Remote access is managed". The statement says nothing about how to correctly manage access and, if the channel used for communication is not protected, any form of access control could be made useless by an attacker intercepting requests and responses to potentially gain access to remote resources without even having to break access control. Moreover, sometimes the interviewed person is not someone well informed on the technical details of the assessed system and may not fully understand the questions presented to them during the assessment process. Finally it is also important to keep in mind that humans are inherently unreliable: answering questions is affected by mistakes, forgetfulness and maliciousness, all of which can alter the results of an assessment.

Given the unreliable nature of interviews and questionnaires, it is legitimate to ask why they are not replaced by in-depth, highly technical assessments including penetration testing and technical analysis of systems and infrastructures. While such activities are far more accurate being based on facts and data, they also come with their own set of limitations. The main steep requirement of a technical assessment is the cost, both in terms of money and time. Not only is it costly to pay a team of cybersecurity experts to perform testing activities, but those same activities may also obstruct business processes, resulting in decreased productivity for all the duration of the analysis, which may last for a long time, especially in larger enterprises. Moreover, the results obtained by technical assessments are very fine-grained, which is good for specific interventions, but not suitable for large scale management without some form of synthesis process.

It is thus clear that a third, option, representing a middle point between reliability and cost, would be beneficial for security governance. Finding such compromise is one of the goals behind research in the field of data driven security governance. The way to achieve that is finding methods to inform abstract governance processes with technical data, leveraging automation as much as possible. The concept of using technical data for supporting security assessments was already touched upon in "Toward a Context-Aware Methodology for Information Security Governance Assessment Validation"[3]. That approach, however, relied on complex Attack Graphs, which require

[2] `https://www.cybersecurityframework.it/framework-v10`

highly granular technical data and thus are not suitable for an assessment at a "middle" level of abstraction.

In this thesis I will explore the field of Data Driven Security Governance in an effort to find and organize some of the possible solutions to take technical data and use it to support security assessment. Then, learning from the current state of the art, I will present *GDFD*s, an extension of *DFD*s designed to support the processes of security governance and, in particular, the extraction of security controls. I will also integrate the use of *GDFD*s with the current version of the Italian National Framework for Cybersecurity and Data Protection by providing each subcategory in the Framework with its own *GDFD*. Then I will develop an automatic tool which leverages *GDFD*s to automatically extract controls. Finally I will evaluate the controls automatically extracted from the *GDFD*s associated to the subcategories of the Italian National Framework for Cybersecurity and Data Protection, to understand if the controls generated by software can fulfill their goal of supporting and complementing those created by humans, leading to more objective and impartial results when using the Framework.

This thesis will be organized with the following structure. Firstly Chapter 2 offers background information on some of the topics touched by the thesis. Chapter 3 is dedicated to the collection and organization of several existing studies about leveraging technical data to support decision making at higher levels of abstraction. Chapter 4 contains the definition of *GDFD*s, alongside descriptions of different use cases. Chapter 5 contains information about the work done to support the Italian National Framework for Cybersecurity and Data Protection with the use of *GDFD*s. In Chapter 6 the results of the process of automatic control extraction based on *GDFD*s are presented and discussed. Finally Chapter 7 contains conclusions and possible directions for future work.

Chapter 2

Background

This chapter contains background information and definitions which will be taken for granted in the rest of the thesis. Each topic is assigned a section which will provide a brief introduction as well as references to more detailed sources. In particular four topic will be treated:

- Data Flow Diagrams and their basic extensions;
- Threat intelligence sharing with STIX and TAXII;
- The Italian National Framework for Cybersecurity and Data Protection;
- Cybersecurity assessments and risk assessments;
- Cybersecurity repositories;

2.1 Data Flow Diagrams

A *DFD* is a type of diagram, similar to a flowchart, commonly used to represent how data flows between the different components of a software or more in general a system.

The first definition of *DFD* by Tom DeMarco was provided all the way back in 1979[13]. This first proposal already presented all the four basic elements which still make up *DFD*s today namely:

- Data Flows;
- Functions, now more commonly called Processes;
- Files/Databases, now commonly referred to with the more abstract name of Data Stores;
- Input/Output, which over the years got generalized to Entities;

Data Flows are represented by directed arrows connecting the other elements and are usually accompanied by a name describing the kind of data which is getting transferred. Their meaning is extremely intuitive and they generally conform to the rule of having only one direction and two endpoints, one being the source and the other the destination.

Processes represent a component performing some kind of work, which receives data flows in input and produces data flows as output. In general they correspond to some piece of software, but

their scale can vary a lot being capable of representing both a simple atomic operation as well as a whole complex program. In the graphical representation of *DFD* Processes are generally ellipses with a name which indicates the functionality it provides.

Data Stores correspond to places where data is stored, like files, databases, hard drives etc... They are completely passive components and do not have any other function aside from storing data and providing read and write access to their space. They are generally drawn as a rectangle missing one or both the left and right sides. They also receive a name indicative of the kind of data they store.

Entities are the representation of everything else which interacts with data flows. Typically they correspond to components outside to the system that is being modeled and thus are also referred to as External Entities. Frequently they are the origin and/or the final destination of data. Additionally data flows connecting them directly are usually omitted from the *DFD* as they do not provide additional insight on the modeled system. The graphical representation for Entities is a rectangle, containing a name which describes them.

Currently *DFD*s are mostly used for the activity of security threat modeling alongside techniques like STRIDE[28] and LINDDUN[50]. However, even for that application, the suitability of basic *DFD*s is being questioned by researchers[43], while extended versions of this kind of diagrams are gaining popularity. In literature there are many examples of *DFD* extensions which add new elements and modify the rules for a well structured diagram, depending on the goal the *DFD* is designed for.

A very common addition in *DFD*s designed with Information Security in mind is the concept of **Boundary** sometimes alternatively rephrased as Area or Zone. They represent collections of elements sharing some kind of property, such as a common physical location or a relationship of trust between them. In the diagrams they take the form of squares or polygonal shapes with dotted lines as sides, which encircle all the contained elements.

Another common modification to standard *DFD*s is to use them with a **multilevel hierarchy**. This modification does not add new elements, but instead the ability of moving across different levels of abstraction by combining or dividing processes. To do so it is common to follow a top down approach, starting from a "level 0" diagram with only one process. Then it is possible to obtain diagrams with higher level numbers (i.e. lower level of abstraction) by partitioning the processes in the previous level into more detailed processes. However it is also possible to operate in the opposite direction and combine together processes tackling the problem with a bottom up perspective.

Those mentioned above are just a few of the most basic modifications made to *DFD*s, but there are a variety of more complex extensions such as Object-Oriented *DFD*s which include additional elements to model classes, attribute and methods[23]. Several more example of extended *DFD*s, will be discussed in Chapter 3 and then a new extended version, called *GDFD*, will be proposed in Chapter 4 discussing each element in detail.

2.2 Threat Intelligence Sharing

Structured Threat Information eXpression (STIX)[6] "is a language and serialization format used to exchange *Cyber Threat Intelligence (CTI)*"[1]. It is open source and managed by the community OASIS which includes members coming from governments, the academic world and also the private sector. The role of *STIX* is crucial to share complex indicators of compromise, especially those which are more abstract, since sharing them with natural languages would inevitably result in the risk of misinterpretation and translation errors.

The *STIX* format is graph-based, all the various elements involved in an attack are called Security Objects and are represented as nodes in a graph. Relationships between Security Objects are instead represented as the edges of the graph and are called Relationship Objects. By default *STIX* uses twelve Domain Objects ranging from "Attack Pattern" to "Vulnerability" as well as two main kinds of Relationship Objects, namely "Relationship", which can assume different meanings depending on the Security objects it is linking, and "Sighting" denoting the belief that a certain piece of intelligence was spotted. In addition a third kind of objects called "Markings" are used to store metadata, for example sharing permissions and requirements. Markings are applied to Security Objects to indicate they possess some particular properties.

From a more technical perspective, in the most recent version of *STIX* all objects are represented in *JavaScript Object Notation (JSON)*. Both Security Objects and Relationship Objects share a few common properties used for identification and versioning which are of key importance for storing them correctly in a database. On the other hand, Markings cannot be versioned, nor be part of a relationship, but rather are directly referenced by Security Objects with dedicated fields.

The fact that *STIX* objects are saved with standard *JSON* format is convenient for a variety of reasons including:

- Having *STIX* data be easily parsed with any programming language;
- Making *STIX* diagrams convenient to store in databases, in particular non relational one that use *JSON* by default;
- Allowing for easy sharing between multiple parties;

One of the most convenient ways to share *STIX* data is with *Trusted Automated eXchange of Intelligence Information (TAXII)*[11], an open source protocol built on top of HTTPS to share threat intelligence. However, despite being designed with *STIX* in mind, it is important to notice that *TAXII* can work with any format storing *CTI* and on the flip side *STIX* data can be transmitted with any suitably designed protocol.

2.3 Italian National Framework for Cybersecurity and Data Protection

The Italian Cybersecurity Framework [5] was created in 2015 as a collaboration between Sapienza University and the *CINI*. The Framework was inspired by the *NIST* Cybersecurity Framework which

[1] `https://oasis-open.github.io/cti-documentation/`

was taken as a starting point and adapted to better suit the Italian landscape, which is dominated by small and medium enterprises in contrast with the large corporations found in the United States.

The Italian Cybersecurity Framework inherits its structure from the *NIST* Cybersecurity Framework, thus the contents are hierarchically organized in functions, categories and subcategories. Functions represent the five main activities to consider in order to manage cybersecurity risks in an effective way. The five functions are: Identify, Protect, Detect, Respond and Recover. Each of these functions contains a series of broad security objectives called categories. Subcategories are more specialized objectives and each category contains a set of them. Moreover each category is accompanied by a set of informative references which contains references to other documents and standards.

The set of functions, categories and subcategories represents the Framework Core, but there are also two additional concepts inherited from the *NIST* Cybersecurity Framework: Profiles and Implementation Tiers.

Profiles are the results of the selection of a set of subcategories. Each Organization can select different sets of subcategories and create their current or target Profile.

Implementation Tiers describes the level of rigor and sophistication assigned to each security objective by the organization. They are meant to be a tool for supporting decision making regarding the security posture of the organization.

However the Italian Cybersecurity Framework also expands upon its American counterpart. The most notable additions which can be found in the version 1.0 of the Italian Cybersecurity Framework with respect to the *NIST* Cybersecurity Framework are: priority levels, maturity levels and ability to contextualize the Framework to the particular reality of a certain sector or organization.

Priority levels are assigned to each subcategory and support the organization in making more informed decisions on which cybersecurity objectives should be met first and which instead do not require urgent intervention.

Maturity levels allow to measure the maturity of security processes and resources assigned to meet the cybersecurity objective defined by each subcategory.

Contextualizing the Framework means selecting a set of relevant functions, categories and subcategories, then defining priority levels, implementation guidelines and maturity levels for all the selected subcategories. Once the framework has been contextualized for a certain sector, that contextualization can be shared among several organizations.

In 2019 the Framework received a major update and moved to version 2.0, changing its name into "Italian National Framework for Cybersecurity and Data Protection"[4]. This new version maintains the same exact structure as version 1.0, thus it is completely backwards compatible. However the update to version 2.0 brought many new additions which further differentiated the Italian National Framework for Cybersecurity and Data Protection from the *NIST* Cybersecurity Framework.

The main two additions found in the second version of the Framework are a set of new categories and subcategories and the introduction of contextualization prototypes.

The new categories and subcategories are for the most part focused on personal data protection and made the Framework compatible with the European *GDPR*.

On the other hand, contextualization prototypes are "pieces" of contextualization. They include a set of categories and subcategories with relative priority levels, implementation guidelines and maturity levels but instead of covering the full Framework they are focused one specific topics, such as compliance with the *GDPR*. They allow for a much easier customization of the Framework, since a full contextualization can be obtained by merging several prototypes together.

2.4 Cybersecurity Assessment and Risk Assessment

A cybersecurity assessment is the process of determining the cybersecurity posture of an organization. While the objective of cybersecurity assessments is clear, there are a wide range of methods which are employed to reach it. In their work "Review of cybersecurity assessment methods: Applicability perspective" Leszczyna[29] identifies six major assessment methods categories:

1. Checklist-based evaluation and compliance checking
2. Vulnerability identification and analysis
3. Penetration testing
4. Simulation or emulation-based testing
5. Model-based testing, formal analysis
6. Reviews

As it is possible to see, the list includes methods both at an extremely technical level and at a high level of abstraction. However the author states that most methods are based on tests, examinations or interviews. Many standards and frameworks define their own cybersecurity assessment methodologies, for example the methodology for the Italian National Framework for Cybersecurity and Data Protection can be found on its official website[2].

While the name is similar, the process of cybersecurity assessment is not to be confused with risk assessment, which is instead the activity of estimating risks by combining information about vulnerabilities, threats, threat actors and estimated impact of potential incidents. However cybersecurity and risk assessments are not completely separate activities as they can interact, each supporting the other with their own findings and results.

In both the case of cybersecurity assessments and risk assessments, the ever increasing complexity of systems and threats is creating new challenges. Zaydi and Nassereddine propose a synchronization between the various processes of risk management, governance and compliance[52]. On the other hand both in the industrial field and the IoT, which presents particular challenges for risk assessment[27] , several studies propose to introduce automated processes for risk assessment[32, 36].

[2] `https://www.cybersecurityframework.it/`

2.5 Cybersecurity Repositories

Online cybersecurity repositories are an essential source of data about cybersecurity and a crucial platform allowing for information sharing between public bodies, private companies, academic institutions and independent researchers. Currently some of the most important repositories are CAPEC[3], CWE [4] and CVE[5] all maintained by MITRE.

CAPEC is a catalog of attack patterns, which are descriptions of the methods used by attackers to exploit vulnerabilities. Each entry of CAPEC includes a description of an attack pattern and other useful information about it such as: typical severity, typical likelihood, prerequisites for the attack and sometimes even examples.

CWE is a list of common software security weaknesses. The entries of CWE contain information about a particular weakness, with each entry having fields like "Applicable platform" listing the affected programming languages or the self explanatory "Likelihood of exploit". Moreover CWE entries are referenced in CAPEC entries' "Related weaknesses" field.

CVE is even more technical, being a list of disclosed vulnerabilities. CVE entries have a detailed description field as well as references to further material to understand the vulnerability. Moreover CVE entries are referenced inside the CWE entries' "Observed examples" field. This makes it technically possible to reach the vulnerabilities from the attack patterns passing through weaknesses, however both CAPEC "Related weaknesses" and CWE "Observed examples" are frequently left empty, thus making a full "chain" of references uncommon.

[3] `https://capec.mitre.org/`
[4] `https://cwe.mitre.org/`
[5] `https://cve.mitre.org/`

Chapter 3

Related work

Even if not always geared towards governance, there exists a variety of studies examining how to gather, organize and leverage technical information from the operational level in order to inform decisions taken at higher levels of the organization. Moreover there are several papers exploring the possibility for automation in such processes.

This chapter provides a wide, but not complete, overview of such proposals, organizing them and highlighting their strengths and weaknesses. First a short section is dedicated to the methodology employed to find the scientific papers. Then papers are organized in three broad groups:

- High level frameworks, containing papers that propose using technical data and/or metrics from a top down perspective;
- Papers presenting tools assisting the conversion of *CTI* information in the *STIX* format and subsequently leverage the formatted data;
- Papers exploring the creation and use of *DFD* models and propose extended versions of such diagrams to better capture possible security issues;

Finally at the end of the chapter all related proposals are organized across two axes, each with values ranging from "None" to "Automatic". The first axis is "formatting" and represents the level of automation of converting data in a format that can be processed. The second axis called "analysis" instead represents the automation level achieved by the techniques to extract insight from correctly formatted data. The full systematization is reported in Table 3.2.

The process followed to search for articles related to data-driven security governance can be split into two major phases: the first characterized by the use of search engines and the second one leveraging cross references. However such phases are not mutually exclusive and the first one continued well after the second one started.

In the first phase, Google Scholar and IEEE Xplore were queried with a series of keywords related to the topics of security governance and security assessments. Moreover the papers found by the search were examined to inform the next selection of keywords for the next search queries.

In the second phases Google Scholar was used to search both for papers cited in previously found ones as well as to explore forward in the network of citations thanks to the "cited by" function provided by the platform.

In addition the survey "Information security governance challenges and critical success factors: Systematic review"[2] was scanned to find more related studies. The survey presents a list of success factors divided in several domains, three of which were selected as relevant and explored, namely: "Compliance", "Assessment/Auditing" and "Measurement".

After examining all success factors in each of the selected domains, the papers cited in the survey were filtered according to the following rules:

- the date of publishing is 2018 or later;
- the paper is listed as presenting a critical success factor in the "Measurement" domain or the "Evaluate procedures and policies" (EPP) success factor;

Applying the two rules listed above yielded a total of eight papers, which are listed in Table 3.1. Unfortunately, a more detailed examination of the results reveals that most of those papers, even those exhibiting critical success factors in the "Measurement" domain, do not consider supporting the processes of Security Governance with data from a lower level of abstraction.

	"Measurement" Domain	EPP Factor
Hina et al. (2019) [20]	yes	yes
Jackson and Rahman (2018) [25]	yes	yes
Maynard et al. (2018) [33]	yes	no
Nicho (2018) [35]	yes	yes
Mounia and Bouchaib (2019) [34]	no	yes
Alghamdi et al. (2019) [1]	yes	yes
Manjezi and Botha (2019) [31]	yes	no
Sönmez (2019) [45]	yes	no

Table 3.1: Survey filtering results

3.1 High level frameworks

In this section are grouped a series of proposals that, while keeping a high level of abstraction, try to attune governance processes with technical data and or metrics from the lower levels of the organization management.

Proposed by Evans et al. *Human Error Assessment and Reduction Technique - Information Security (HEART-IS)* [15] is *Human Reliability Assessment (HRA)* tool, obtained as an adaptation to information security of *Human Error Assessment and Reduction Technique (HEART)*[49] which was originally designed for the nuclear sector during the 1980s and later adopted in several industrial

sectors. The basic idea of *HRA* is to estimate the likelihood of incidents caused by human errors. The results obtained by applying *HEART-IS* can thus be used to inform information security governance processes with metrics regarding human behavior.

While *HEART-IS* is slightly extended, the process of computing the likelihood of incidents remains the same as *HEART*. First the task under examination is mapped in one of the nine *General Task Type (GTT)*s provided by the tool, each with its own value of "human reliability" ranging from zero to one. Then a small number of *Error Producing Condition (EPC)*s, usually three from a list of 38 provided with the tool, is selected for the task and each *EPC* receives an *Assessed Proportion of Affect (APOA)*, which is the weight of their impact on the task. Once all those values are assigned the likelihood of failure can be calculated using a simple formula.

As an addition *HEART-IS* expands basic *HEART* with two new components:

- the mapping element, which allows to match information security incidents with *HEART-IS* data;
- the analysis element, used for comparing the *HEART-IS* results with actual incident frequencies obtained by historical data;

While this work is ambitious in its intention to manage human unreliability with a metric of probability, the proposed *HEART-IS* tool is not effective in achieving its goals. Estimation errors could be related with the inherently imprecise manual assignment of *APOA* weights which is done manually, but they could also stem from the lack of a translation effort from the nuclear sector to information technology since both *GTT*s and *EPC*s, along with their empirical weights, were kept the same, without accounting for the differences of the work performed by humans in the two different environments.

The paper "A conceptual model for a metric-based framework for the monitoring of information security task efficiency"[45] by Sönmez contains the proposal for the Enterprise Security Productivity System, a tool to evaluate the impact of several factors on the productivity of people, tools and processes involved in enterprise security.

This study is not focused specifically on cybersecurity and the tool it proposes is still in a very early conceptual stage. However in the text are presented a variety of metrics which differ from those commonly found in literature[14, 37], yet still rely on factual information to inform processes of security governance.

Examples of such metrics include the following "Team Metrics":

- Days from last .... security audit/analysis
- Days to .... security audit/analysis
- Number of planned/unplanned night time work for security team members
- Number of planned/unplanned weekend work for security team members
- Most active team members

- Least active team members
- Number of days in a month where active security analysis results exist for the team member

There same paper also proposes several "Tool and Processes Metrics" such as:

- Number of file types supported by the security tool
- Number of file format types supported by the security tool
- Number of automated reports provided by the security tool
- Number of network alerts
- Number of restricted network requests
- Ratio of scanned and processed web pages
- Number of application vulnerabilities

In "Toward a New Integrated Approach of Information Security Based on Governance, Risk and Compliance"[52], Zaydi et al. propose a conceptual framework aligning a process of information security governance, inspired by the ISO 38500 standard[24], and risk management, based on 4D-ISS[51], with that of compliance. While this paper only shows a concept in its early stages of development it already shows the idea of informing abstract decisions taken in the information security governance processes with data coming from the risk management and compliance processes which deal more directly with technical issues.

3.2 Leveraging STIX for Cyber Threat Intelligence

STIX provides a lot of different objects to represent attackers, vulnerabilities and much more, making it a fitting format to communicate technical data to higher levels of management or feed it to automatic tools that assist decision making in governance processes. In the literature there are a variety of tools which try to convert raw data in *STIX* format and/or leverage data already in the form of *STIX* diagrams for security purposes.

STIXGEN [22] is a tool capable of constructing *STIX* formatted diagrams by receiving data from a database. A particular focus is put in creating diagrams about single APT campaigns. The most important component of STIXGEN is the *STIX* encoder which, receiving data from the database, performs the following operations:

1. Add header information;
2. Connect to the database;
3. For each campaign in the database:
   (a) Retrieve campaign ID and title;

(b) Fetch indicators, incidents, exploits and other intelligence adding them to the *STIX* report;

4. Exit after all campaigns are processed;

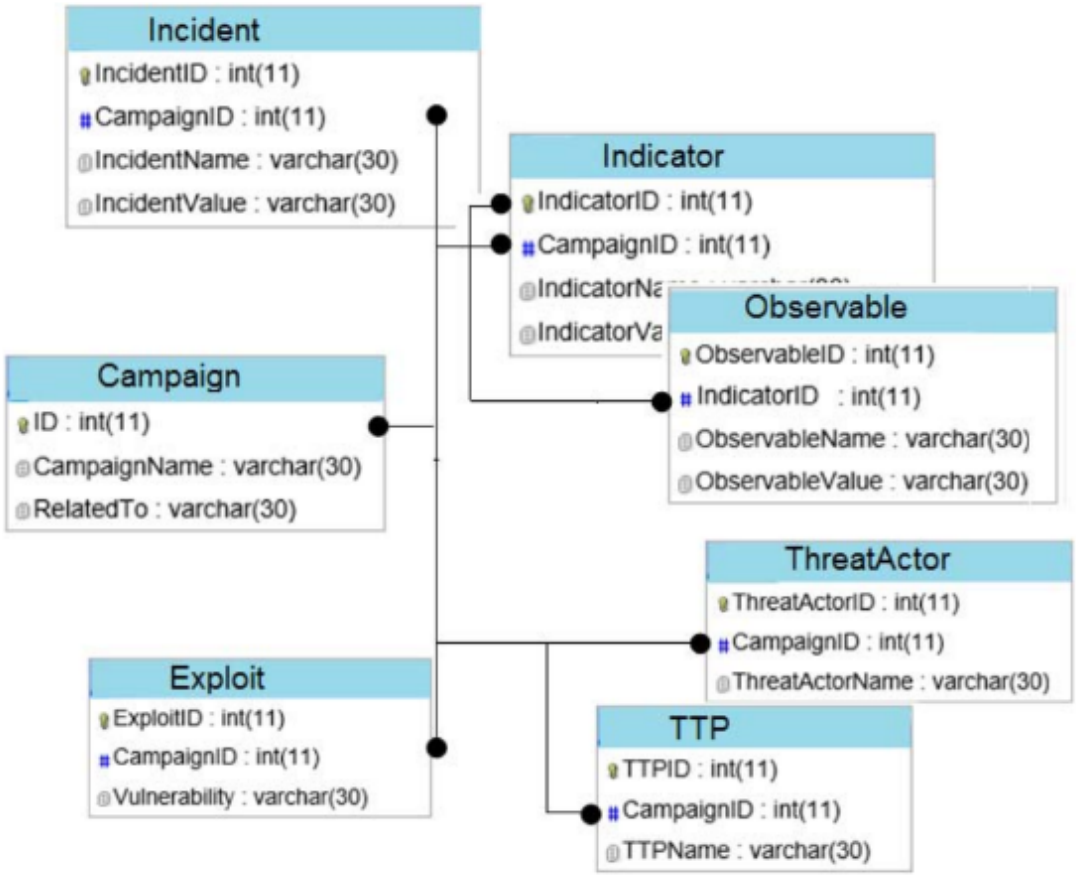


Figure 3.1: STIXGEN's Database Schema

While the *STIX* encoder is completely automatic it is important to notice two important limitations of STIXGEN:

- The tool relies on the existence of a database with a specific structure (shown in Figure 3.1) containing the information it needs and merely performs a conversion in the *STIX* format;
- It is assumed that the database is filled with data by a human operator and any mistake made by them when gathering or inserting will be propagated in the *STIX* output without any kind of checks by the tool;

Another automatic tool for conversion of threat intelligence data to the *STIX* format is proposed by Sadique et al.[38]. This study has the ambitious goals of designing a tool which gets as input raw data in heterogeneous formats, possibly distributed across many devices, while also adding an additional level of privacy preservation. The generation of *STIX* objects is organized in the three following steps:

1. Data collection > with appropriate plugins, data can be collected from several sources and in different formats (plain text, *JSON*, XML ...);
2. Data classification > receiving one event at a time, the classifier module labels the event depending on its source and data formatting, then passes the results to the converter;
3. *STIX* conversion > the converter module parses data obtained from the classifier and uses it to create *STIX* "observed data" objects, each containing one or more "cyber observable objects". Parsing uses custom python scripts while the conversion relies on the official *STIX* python library.

In addition each piece of data is assigned a privacy level ranging from level 0, meaning non-sensitive data that can be shared in clear text, to level 3, denoting private data that should be encrypted and only managed with private set intersection protocols.

The unnamed tool proposed in this paper is held back by several serious limitations. For starters the tool only produces "observed data" *STIX* objects which will need to be later integrated into a larger collection of *STIX* objects to reach their full usefulness. In addition it appears that the tool is not capable of discerning automatically the kind of data in input, nor its privacy level, meaning that each source has to be manually and statically configured. Finally, privacy levels do not seem to be enforced in any way nor have any impact on the process of conversion from raw data to *STIX* and are even ignored in the "demonstration" section of the paper.

A wider perspective is taken by Hague and Krishna[19] which envision a fully automated cyber-defence system. Such a system not only would gather data from multiple heterogeneous sources and convert it into the *STIX* format, but also provide functionalities to share intelligence and trigger the execution of defensive measures. The sharing of information is implemented via *TAXII*, while different components in the proposed system communicate with each other thanks to a variety of RESTful APIs.

In the conceptual system, the translation in the *STIX* format involves three main elements:

- Threat detection system, capable of detecting threat data used as input;
- *STIX* generation system, which given threat data translates it in the *STIX* format;
- Data handler REST project, mediating communications between the detection systems and the *STIX* generation system, as well as all communication from and to a threat database;

The cyber-defense system has been deployed in an experimental setup using a machine learning algorithm to detect the Slowloris DDoS attack as a threat detection system, an OpenStack server with appropriate GET and POST APIs as the data handler, and a python web service using the official *STIX* 2.0 library as the *STIX* generation system. However, looking at what is missing from such an experiment highlights some of the challenges to overcome in order to implement a complete cyber-defence system. Firstly the experimental setup only detects a specific kind of DDoS attack, extracting more kinds of threat intelligence from different sources and correctly categorizing such

data would require a more advanced solution. Secondly the experimental setup experiences drops in detection accuracy when receiving malicious data as input. Finally the latency introduced by HTTP might mean that RESTful APIs are not ideal for triggering defensive measures.

3.3 Security Applications for Data Flow Diagrams

*DFD*s are popular in the field of programming because they allow developers to quickly sketch the interaction between software components. For the same reason they are also used for threat modeling purposes, having each of their elements listed and examined to find security flaws and possible threats. Usually this process is performed manually by a security expert following a loose procedure, however there are several tools and methodologies to analyze *DFD*s in a more precise and faster manner, in some cases even bypassing a human operator completely in favor of automated software.

Microsoft developed the *Microsoft Threat Modeling Tool (MTMT)* [1] with that exact purpose in mind as a core part of the Microsoft Security Development Life-cycle. The tool provides a canvas and simple graphical tools to create a *DFD*, then it helps keeping track of threats by offering for each element a table which can be filled by a security expert. These tables have fields for basic information about the node and all STRIDE categories in order to facilitate threat elicitation. Once the process of filling the tables is finished the tool allows for easy sharing by automatically generating a human readable report.

While convenient, the *MTMT* offers very little benefits in terms of automation and reliability. However there are several studies on the topic of extending the capabilities of *DFD*s and improving the procedures using them, all in an effort to make powerful *DFD*-based security tools.

Being actively developed as an OWASP project, pytm[2] is a threat modeling tool with several automated features. It takes as input a file containing the definition of an architectural design and then automatically generates a *DFD* and a sequence diagram. Moreover the tool is bundled with around one hundred threat definitions which are used to find those threats in the generated *DFD*. Threat definitions are expressed in the *JSON* format and describe patterns indicating the presence of some security flaw or possible threat. Additionally each threat definition also contains a "references" field with URLs pointing to relevant entries in CAPEC and CWE. For a few threats there are also pointers to CVE vulnerabilities.

SPARTA[41] is another automated tool which leverages *DFD*s. Given a *DFD* as input this tool can find threats and in addition also establish an order of priority between them based on risk. This is possible thanks to the combination of several factors.

Firstly SPARTA does not use a classic *DFD* model, but instead leverages the additional functionalities provided by Solution-aware Data Flow Diagrams[42], an extension of *DFD*s designed by the same researchers developing the tool. As the name suggests, among the several additions brought by Solution-aware *DFD* the most notable one is the introduction of reusable templates for

[1] `https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool`
[2] `https://github.com/izar/pytm`

security solutions, which can be instantiated and bound to *DFD* elements. Each solution is also linked to a set of countermeasures that help mitigate specific kinds of threat and these relationships are considered when estimating the risk. The meta model for Solution-aware *DFD*s is implemented in Encore from the Eclipse Modeling Framework and diagrams can be drawn on a virtual canvas thanks to the graphical interface provided by the software.

For finding threats, SPARTA uses rules specified in the Acceleo Query Language, each corresponding to a particular pattern, and then leverages VIATRA[12] to find matches in the *DFD*.

Finally for estimating the risk SPARTA considers the FAIR[17] risk components, which are: strength, threat capability, contact frequency, probability of action and loss magnitude. Each of such components is considered as a modified PERT distribution to account for uncertainty, and they are handled by performing Monte Carlo simulations sampling from the distributions.

SPARTA is still being worked on and its developers plan to make it easier to add security solutions, add support to decision making by evaluating the impact on risk of adding a new security solution, and centralizing knowledge bases for providing SPARTA with up to date information about threats and countermeasures.

In their paper "A data-driven security risk assessment scheme for personal data protection"[10] Cha and Yeh point out how many existing privacy regulations require to "adequately protect" personal data without expressing exact requisites, hence they try to solve the problem by developing a tool to compute what is a reasonable level of protection.

The proposed solution is based on the use of *DFD*s enriched with the addition of logical boundaries and physical boundaries represented with dotted lines around the elements and interfaces allowing data flows to cross those boundaries, as well as the distinction between physical and virtual data storage elements.

Once all data flows are represented with a *DFD*, a formal procedure (shown in Figure 3.2) can be followed to identify potential incidents iterating on each data flow, system, interface, physical region, physical media and entity.

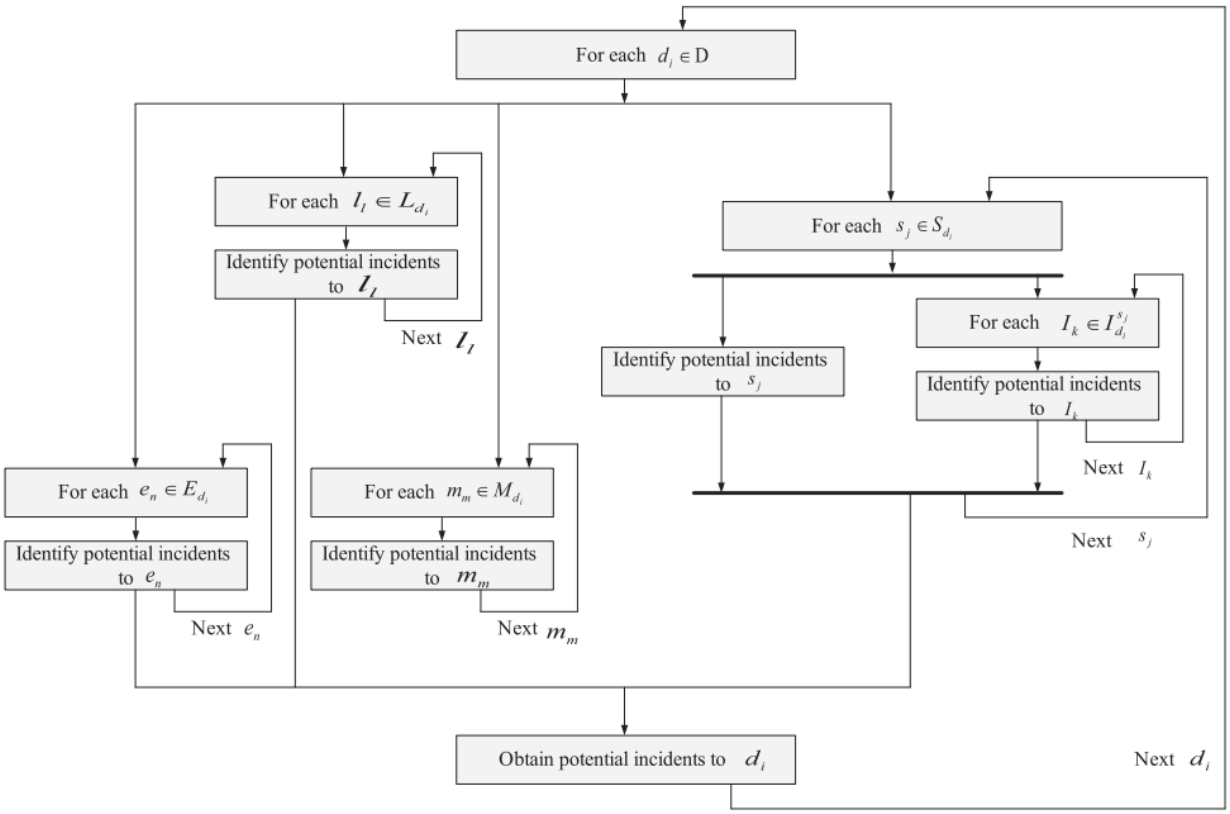


Figure 3.2: Steps to identify potential incidents

The authors themselves recognize some of the limitations of this tool, in particular:

- the lack of software supporting the creation and analysis of *DFD*s;
- the limited exploration of new types of element for *DFD*s aimed at increasing their expressiveness;
- the fact that all data is treated in the same way by their current solution;

In their paper "Automatically Extracting Threats from Extended Data Flow Diagrams" [8] Berger et al. present a modified version of the *DFD* model, simply called *Extended Data Flow Diagram (EDFD)*. Alongside that, they made a catalog of threats and a tool.

*EDFD*s are different from regular *DFD*s in several key aspects. The most evident modification involves data flows which are replaced by "channels" and "data" elements. Differently from data flows which are unidirectional and can connect only two elements, channels can have multiple endpoints and data can flow in them in any direction. Data elements are associated with channels and can be further specialized depending on their nature, allowing, for instance, to have User Data as a subtype of generic Data. Data stores, processes and interactors (i. e. entities) are all grouped together as subtypes "Elements" and can be further specialized as needed. Finally *EDFD*s also include "Trust Areas" an evolution of trust boundaries already seen in other *DFD* extensions, which can be specialized as well depending on their nature, for example in "Network boundaries" or "Machine boundaries".

To find threats in *EDFD*s the authors propose the definition of rules in the Cypher Query Language[16], which is similar to SQL, but designed to describe subgraphs to be matched in an arbitrary graph. In the paper are provided some examples of rules made to find patterns related to relevant CWE entries. However the reliance on humans to create rules is an important limitation of this approach, with another being the NP-complexity of the subgraphs isomorphism problem, which means that on sufficiently large *EDFD*s performance will quickly degrade.

Nevertheless, that second limitation can be overcome by deploying machine learning solutions to solve the problem of finding the subgraphs. This will inevitably slightly decrease the accuracy of the search for subgraphs, but is also much more scalable than a brute force approach. Many researchers gave their contribution to the topic of solving subgraph isomorphism with artificial intelligence as shown by the surveys by Jiang et al.[26] and Somkunwar and Vize[44] as well as the more recent study by Liu et al.[30].

Another kind of *EDFD* is described in "Detecting violations of access control and information flow policies in data flow diagrams"[40]. As the title suggests, the additional information stored with this kind of *DFD* is related to how data flows traverse the diagram. For this reason nodes, which correspond to data stores, processes and actors (i.e. entities), as well as data flows receive a label. Several behaviors are defined to specify what are the effects of source and destination on any given data flow. Examples of behavior include: "Forward" which copies the data in input as output without changing properties and "Declassify" which instead changes the value of classification of the data flow.

Similarly to the Solution-aware*DFD*s used in SPARTA, also this model has been implemented in the Eclipse Modeling Framework which also allows for an automatic analysis of the *EDFD* searching for security violations. However, while solutions like SPARTA search for patterns in the*DFD*, this one leverages labels and behaviors.

The procedure to to correctly perform an analysis is described in the paper and includes the following steps:

1. The security expert creates an analysis definition, a collection of properties for data and nodes and behaviors;
2. The security expert defines a comparison function in the prolog language specifying how the elements in the analysis definition should interact with each other;
3. The software designer models the structure of the system with an *EDFD* applying the elements defined in the analysis definition;
4. The comparison function is used to perform a fully automated analysis of the *EDFD* and detect violations;

The current implementation of this kind of *EDFD* works, but the authors recognize an important limitation in the fact that it is not possible to represent individual users or pieces of data, preventing the expression of constraints involving two or more different elements of the same class. Moreover the usability of the model is yet to be validated.

A similar yet different extension of *DFD* is the Security *DFD*, presented in "Flaws in Flows: Unveiling Design Flaws via Information Flow Analysis"[46]. This extension focuses on the analysis of confidentiality and integrity and in order to achieve its objectives leverages a security specification language for Sec-DFD. Such language is used to assign to each node a type with an associated security contract defining its behavior with respect to labels assigned to data flows. Examples of behaviors include "Join" which merges two or more flows keeping the more restrictive label, and the specular "Encrypt" and "Decrypt" which affect the level of confidentiality of the output flow.

In addition to labels and behaviors a particular element introduced by Security *DFD* is the concept of "Attacker zone". Similar to boundaries and trust areas they are represented by dotted lines around a certain set of elements they contain, however their meaning is completely different as they are used to mark groups of elements presenting vulnerabilities which can be exploited by attackers.

Security *DFD*s have been implemented by the authors with the Eclipse Modeling Framework and the VIATRA framework. Given a Security *DFD*, the automated analysis is carried out by visiting every node and applying label propagation functions to spot violations. To support this kind of approach certain nodes of the Security *DFD* need to be marked as data source and data sinks, which is considered a limitation since it can be challenging to do as the designer may not always be aware of all of them.

While there are many tools for performing different kinds of analysis on *DFD*s and their extension, creating a *DFD* is still mostly a manual operation and thus prone to human errors. An early attempt at supporting the creation of *DFD*s was made by Ibrahim and Yen in 2010[21]. They presented a tool which allowed a human operator to draw *DFD*s with a graphical interface, but in addition also capable of checking if a series of basic rules about *DFD* structure were observed. Moreover, the tool also supports the creation of multi-level *DFD* and is capable of checking for consistency between the various levels of abstraction. To this day it seems that no significant advances have been made about the topic of *DFD* generation, with most studies still assuming that *DFD*s are created manually by a human operator, without any particular assistance from software or other resources other than documentation and their knowledge about the system.

3.4 Systematizing Related Proposals

The previous sections organized related work based on the different ways data is handled. However automation is also a key aspect in Data-Driven Security Governance, as its benefits include a faster processing of information and generally more objective results with respect to those obtained with a manual process. That is the reason why I organized, based on their level of automation, all the main solutions proposed in the literature and explored in the previous sections of this chapter.

Depending on the degree of automation, the solutions proposed in related work are organized across two axes:

- Analysis, which indicates the ability of the tool of processing data and producing results to be used in Security Governance processes (e.g. a list of threats)

- Formatting, which indicates the ability of the tool of harvesting raw data and converting it in a format in which it can be processed (e.g. *DFD*)

For each axis, each solution is evaluated on a qualitative scale with the following possible values:

- None = solution does not exhibit the evaluated capability;
- Manual = the work is performed by a human;
- Assisted = the solution includes software which assist the work of a human;
- Automated = the solution include software which automatically performs the task;

The Table 3.2, containing the full systematization, is designed with an unusual structure to more intuitively symbolize the increasing level of automation from the bottom left corner to the top right corner. The row containing column headers was moved to the bottom of the table, in this way the structure of the table reflects that of a pair of Cartesian axes. The level of automation in the "Analysis" process can be seen as represented in the vertical "y" axis, while the horizontal "x" axis holds the values for the level of automation in the "Formatting" process. Thus the overall level of automation intuitively increases the further away the cell is from the "origin" of the axes in the bottom left corner.

Automated		[41][42][8][40][46]	pytm	
Assisted		MTMT		[19]
Manual	[15][45]	[10]		
None			[21]	[22][38]
A./F.	**None**	**Manual**	**Assisted**	**Automated**

Table 3.2: Table of related work ordered by level of automation in analysis (A.) and formatting(F.)

After examining the position of the various proposals in Table 3.2, it is possible to conclude that data can be easily converted in the *STIX* format automatically, however data in the *STIX* format is rarely processed automatically to extract knowledge and inform decision processes at a higher level of abstraction. On the other hand there is currently a lack of solutions for the automated creation of *DFD*s, but, once a *DFD* is created, there are a variety of different proposed methodologies to further process the data contained inside the diagram. Moreover, studies conducted at a higher level of abstraction typically lack any form of automation, even if dealing with more technical data and/or metrics.

Chapter 4

Governance Data Flow Diagrams

This chapter is dedicated to presenting *GDFD*s, a new extended version of *DFD*s which adds several functionalities geared towards supporting Information Security Governance.

The first four sections present all the components which can be found in a *GDFD*, namely: Data Flows, and their specialized versions, Processes, with their behaviors, Data Stores, Entities and Areas, and their specialized versions. A complete structure of the relationships between these components can be found in Figure 4.1. Then a summary of the graphical representations for the basic elements can be found in Figure 4.2, while those for specialized Data Flows and Areas can be found in Figure 4.3. On the other hand the graphical representation for the different Process Behaviors are contained in Figure 4.4.

The second half of this chapter is instead devoted to all the different applications and use cases supported by *GDFD*s, namely: extracting cybersecurity controls, defining Security Templates, computing security metrics and detecting security flaws.

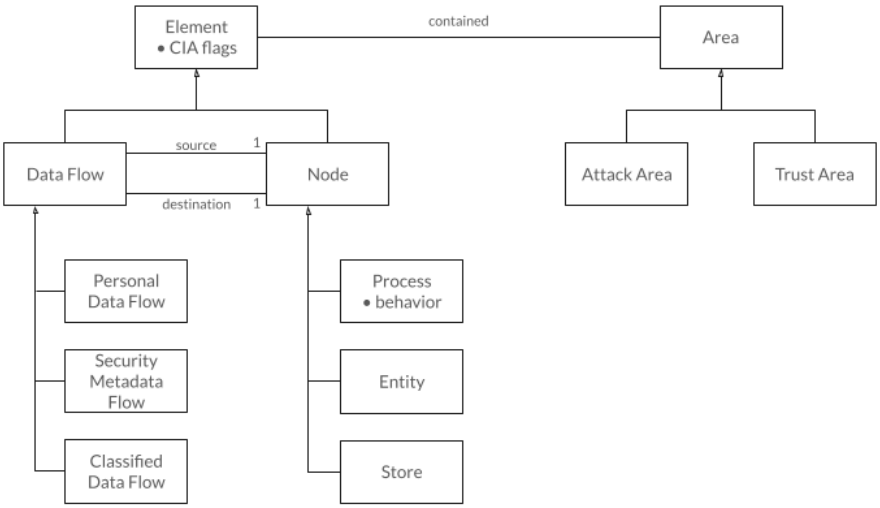


Figure 4.1: Relationships between *GDFD* components

4.1 Elements

The elements composing *GDFD*s are extremely similar to those or regular *DFD*s. A first distinction is made between Data Flows and Nodes. Nodes are then further divided into Processes, Data Stores and Entities.

Data Flows as usual represent data moving from one Node to another. Each Data Flow has one single source node, becoming an output for that node, and one destination node, becoming an input for that node. There are no restrictions on how many Data Flows a single node can have as input or as output. In addition each Data Flow is given a name, which should be representative of its meaning in the system, and a set of CIA flags, which represent the requirements for the data inside the flow in terms of Confidentiality, Integrity and Availability. In the graphical representation of a *GDFD*, Data Flows are directed arrows connecting the source node to the destination. The name and the CIA flags are written near the arrow, taking care to make clear to which one they refer to if there are multiple Data Flows close to each other. Where there are no particular requirements and CIA flags are all turned off, it is suggested to omit them from the graphical representation for the sake of a less cluttered diagram.

Nodes have different meanings and graphical representations depending on their type. However all nodes have a name, representative of their meaning in the system, and the three CIA flags representing the properties each node needs to preserve. This information is drawn inside the shape of the Node and, like with Data Flows, CIA flags are omitted when they are all turned off.

Processes represent any kind of procedure which manipulates, creates or consumes data, regardless if it is automatic or performed by a human. Each Process in addition to the properties shared with all other types of Nodes, also has a behavior which characterizes the relationship between data flows in input and those in output. Graphically Processes are represented as ellipses. Different behaviors are represented by small shapes placed outside the ellipse, this topic is treated in more detail within Section 4.4.

Data Stores, similarly to standard *DFD*s, represent logical or physical locations where information is stored. They are passive elements which do not possess any particular functionality if not those of storing and providing data to processes and entities. They are graphically represented as rectangles.

Entities are all other nodes which do not fit the definition of Process nor Data Store such as external organizations, people or external systems. In general Entities are mostly elements which cannot be controlled directly or are outside the scope of the diagram. Often they generate data flows or act as sinks for the information. In *GDFD*s the interaction between Entities are often, but not always, outside the scope of the diagram. As such, direct Data Flows between Entities, while allowed, are to be carefully considered. The representation for Entities is a rectangle with the top right corner cut off.

4.2 Areas

One additional feature that *GDFD* have over regular *DFD* is the support for Areas. At a very intuitive level Areas are collections of elements, and should have a name which reflects the properties

common to all those elements. Nodes can be inside or outside an Area, while data flows can traverse the boundaries of an Area connecting a Node inside with one outside and vice versa. Formally such Data Flows are considered to be inside the Area but will typically require special attention with regards to security.

In the graphical representation of a *GDFD* Areas are represented as a polygonal shape with dashed sides, containing all elements inside the Area.

It is possible to have generic Areas, but also define specialized Areas with specific meanings. Such Area types can be distinguished in the diagram with the use of different colors. Two types of specialized Areas are already defined and ready to use:

- Trust Area;
- Attack Area;

A **Trust Area** is a set of elements protected by proven and reliable security solutions or more in general they represent segments of the system in which security requirements are guaranteed. Such areas are colored in green.

Attack Areas instead denote dangerous zones in the system, presenting some kind of weakness or vulnerability which could allow an attacker to perform malicious activities. This type of Area is represented with the color red.

In a *GDFD* it is possible to have elements that are both in a Trust Area and in an Attack Area. That is a desirable situation as it means that, while there are some vulnerabilities affecting those elements, they are covered by the security guarantees provided by the Trust Area.

Name	Diagram element	Meaning
Data flow		Data moving between two nodes
Node : Process		Procedure working on data, usually with input and output data flows, like a piece of software but also the work of a human
Node : Entity		Services, organizations, people and all other entities which typically produce and consume data
Node : Store		Elements that store data, for example databases
Area		Collections of elements

Figure 4.2: Graphical representation for basic Elements and Areas

4.3 Specialized Data Flows

One of the first major differences between *GDFD*s and regular *DFD*s is the introduction of multiple types of data flows, which are subtypes of the generic data flows. This means that all specialized Data Flows still retain all properties of the generic Data Flow type while opening the possibility of adding new properties fit to their type. Graphically while generic Data Flows are black, each of the specialized Data Flow types is represented by a different color.

Specialized Data Flow types can be defined as needed by the users creating the *GDFD*. However three types are provided by default:

- Personal Data Flow;
- Classified Data Flow;
- Security Metadata Flow;

Personal Data Flows represent the movement of personal data, which in many countries around the world is protected by special regulations, such as the *GDPR* in the European Union. On Personal Data Flows the Confidentiality and Integrity flags are active automatically. In the graphical representation they have the color blue.

Classified Data Flows represent data which has a level of classification and need to be kept confidential in some way. This type of Data Flows has the Confidentiality flag active by default. Users that use more than one level of classification for data are encouraged to further specialize this Data Flow to reflect such levels. In the diagram Classified Data Flows are colored with magenta.

Security Metadata Flows are, as the name implies, the representation of metadata related to security measures flowing in the system. Examples of Security Metadata Flows include information about vulnerabilities, threat intelligence and security roles definitions. For this type of Data Flow all three of the CIA flags are active by default. Graphically they are distinguished by their color purple.

Name	Diagram element	Meaning
Personal data flow		Personal data, should be protected accordingly to regulations
Security metadata flow		Flows containing information about how security is managed in the system
Classified data flow		Classified data that is "confidential" or "secret", if there are multiple levels it is possible to add more specialized flows
Trust area		Areas protected by proven security solutions
Attack area		Areas containing elements which will likely be targeted by attackers because of known vulnerabilities and/or their value

Figure 4.3: Graphical representation for specialized Data Flows and Areas

4.4 Process Behaviors

As mentioned above each Process is characterized by a behavior which expresses what is the relationship between the Data Flows in input and those in output. By default there are five possible behaviors:

- Forward;
- Join;
- Cardinal Split;
- Semantic Split;
- Consume / Make;

Forward is the most simple type of behavior. A process with this behavior receives as input a single data flow and outputs a data flow of the same type. This behavior is used to represent all of those processes which do not modify the data in input or, if they do, do not change the nature of the data but only alter the contents. Graphically Processes with the Forward behavior are plain ellipses with no additional features.

The **Join** behavior is assigned to all those processes that take multiple inputs and merge them together in a single output. A process with this behavior has a generic Data Flow as output only if all the Data Flows in input are generic as well. On the other hand if one or more of the Data Flows in input is of a specialized type, then the output is a Data Flow of that type. However it is not allowed to have multiple different types of specialized data flow as inputs for a Process with a

Join behavior. Processes with such complex input combinations should have the Consume/Make behavior instead. In the diagram Join processes have a small square on the outside to which all input Data Flows are connected.

The **Cardinal Split** behavior indicates that a Process transforms a single Data Flow into more than one of the same type. This includes both processes that perform full and partial copies of the data in input, as well as processes that partition the data flow in input without changing the nature of the data flows in output. Processes with this behavior are not allowed to have multiple Data Flows in input, the type of all output Data Flows is the same as that of the input. When drawing a Process with this behavior, add a small circle adjacent to the ellipse representing the process. All output Data Flows originate from that circle.

The **Semantic Split** behavior is similar to the above mentioned Cardinal Split, but signals a partition of data which alters the type of the output Data Flows. The input Data Flow for a process with a Semantic Split behavior has to be of a specialized type. The process will divide generic data contained in that specialized Data Flow and produce as output at least two data flows. In particular at least one output Data Flow is of the same type as the one in input and at least another one is generic. The graphical representation for a Semantic Split is the same as the one for a Cardinal Split, but with the circle crossed.

Finally the **Consume/Make** behavior is the most flexible one denoting the absence of any special relationship between input Data Flows and output Data Flows. This is the behavior to assign to those processes that in the context of the diagram generate Data Flows or act as sinks for the information, as well as to all those processes taking a complex set of inputs and applying numerous operations to turn them into completely different outputs. In the diagram this kind of processes are simple ellipses, but their border is a double line.

Name	Diagram element	Meaning
Forward		Data flows in input are reported as output
Consume/Make		Data flows in input are consumed, those in output are created
Join		Data flows are combined in one. Specialized data flows can be joined only with generic, resulting in specialized
Cardinal split		Data flow is divided in two of the same type
Semantic split		Divides a specialized data flow in a generic and a specialized of the same type

Figure 4.4: Graphical representation for Processes with different behaviors

4.5 Control extraction

The analysis of *GDFD*s can be helpful to select security controls by considering its elements, (i.e. Data Flows, Processes, Data Stores and Entities), their type and interactions, as well as Areas, Trust Areas and Attack Areas. To help this process of analysis it is recommended to follow a well defined list of rules for control extraction. The rules should be defined in a way that allows for a software to automatically extract all controls in a *GDFD* or a series of *GDFD*, thus minimizing the possibility of human error. While users of the *GDFD* may define any rule they see fit to extract controls, this section provides a set of 13 basic and easy to automate rules which can generate a control each.

The following four rules can be applied for each Process and Data Store:

- If the Process/Data Store exists as a Node in the *GDFD*, generate the control "There is a <name> <node type>. It has the following Data Flows in Input <list of Data Flows in input>. It has the following data flows in output <list of Data Flows in output>";
- If the Process/Data Store has the "C" flag active, generate the control "There is a security measure protecting confidentiality for the <node type> <name>";
- If the Process/Data Store has the "I" flag active, generate the control "There is a security measure protecting integrity for the <node type> <name>";
- If the Process/Data Store has the "A" flag active, generate the control "There is a security measure protecting availability for the <node type> <name>";

The following five rules can be applied to each Data Flow:

- If the Data Flow exists in the *GDFD*, generate the control "There is a <label> Data Flow, with type <flow type> flowing from the Node <source> to the Node <destination>";
- If the Data Flow has the "C" flag active, generate the control "There is a security measure protecting confidentiality for the Data Flow with label <label>";
- If the Data Flow has the "I" flag active, generate the control "There is a security measure protecting integrity for the Data Flow with label <label>";
- If the Data Flow has the "A" flag active, generate the control "There is a security measure protecting availability for the Data Flow with label <label>";
- If the Data Flow is a Personal Data Flow, generate the control "There is a security measure preserving privacy requirements for the Personal Data Flow with label <label>";

Finally the following four rules should be applied by considering each Area in the *GDFD*:

- If the Area exists in the *GDFD*, generate the control "The <area type> <area name> is defined and well documented. It contains the following Nodes: <list of contained nodes>. It contains the the following Data Flows: <list of contained data flows>";

- If the Area is a Trust Area, generate the control "There are security measures protecting the boundaries and the elements contained inside the <area name> Trust Area";
- For each Data Flow traversing the boundaries and entering the Area, generate the control: "The <flow label> Data Flow is managed and well documented as it enters the <area name> <area type>";
- For each Data Flow traversing the boundaries and exiting the Area, generate the control: "The <flow label> Data Flow is managed and well documented as it exits the <area name> <area type>";

As an example of automated control extraction I developed a python tool capable of taking as input *GDFD*s Security Templates saved as multigraphs in a node-link format inside *JSON* files and generating a list of controls by applying the rules above mentioned. The tool works with Security Templates created for the Italian National Framework for Cybersecurity and Data Protection and is presented in more detail in Section 5.8

4.6 Security Templates

A Security Template is a pre-built *GDFD* which does not model any particular system, but rather an example that achieves certain security goals. Security Templates are a powerful tool which can both aid in the creation of custom *GDFD*s as well as supporting the assessment process.

Creating *GDFD*s from scratch can be a difficult and time consuming task. That process could be made easier and faster by leveraging Security Templates as a starting point and then iterating on each element and area to select those to keep and those to prune. Assessment questions automatically extracted from the template are a helpful tool to guide such decisions.

Security Templates can be also seen as objective to reach. In this perspective they can be used to perform gap analysis comparing them with the *GDFD* modeling an actual system. Since *GDFD*s are at their core directed graphs, this activity could achieve some level of automation by leveraging one of the several measures of distance between graphs. Such a measure could then be interpreted as a quantitative indicator of the gap between the current and the target situation. Moreover when transitioning towards the target, the introduction of new components could be prioritized in a way to add first those elements which most reduce the distance with the target *GDFD*.

Security templates can also work together with security frameworks, such as the *NIST* Cybersecurity Framework and the Italian National Framework for Cybersecurity and Data Protection. Those documents already express a multitude of security objectives as a series of Categories and Subcategories. Designing Security Templates for those objectives, aside from providing a different perspective on the subject, can also integrate and expand the text with information that could have not been expressed in a short sentence. As part of this work, I translated most of the subcategories of the Italian National Framework for Cybersecurity and Data Protection into *GDFD*s. More details can be found in chapter Chapter 5.

4.7 Security Metrics

Another way of using *GDFD*s is to leverage their modeling capabilities to make a detailed diagram of the system under assessment and then compute a variety of metrics to evaluate quantitatively the security measures in place. Two simple metrics can be easily computed in an automatic way:

- T/A-C;
- PDX;

The T/A-C metric allows to quickly gauge the level of coverage provided by security measures. This metric is based on the analysis of Attack Areas and Trust Areas. In order to compute T/A-C, it is sufficient to divide the number of elements falling inside both Trust Areas and Attack Areas by the total number of elements falling inside attack areas. The result can be expressed as a number from 0 to 1, or as a percentage. The closer the T/A-C is to 1 (i.e. total coverage) the better protection is in place in the system. However, depending on the context, it could be very difficult to reach the status of complete coverage or even close to it. Thus, before computing the T/A-C, it is suggested to define the "minimum coverage threshold", a number corresponding to the minimum value of T/A-C that is considered to be acceptable.

On the other hand the PDX metric, as the names suggested, focuses on personal data protection. There are two versions of PDX, one called Strict PDX, the other called Relaxed PDX. The strict version is computed as the ratio of Personal Data Flows which are not inside Trust Areas divided by the total number of Personal Data Flows. In the relaxed version instead only those Personal Data Flows which fall outside Trust Areas and Inside Attack Areas are counted and divided by the total number of Personal Data Flows. In both cases PDX can be expressed as a number between 0 (i.e. zero exposure) and 1 (i.e. total exposure), or as a percentage. The closer the PDX is to 0 the more personal data is protected. Similarly to T/A-C, it could be difficult for a real PDX to reach the ideal value of zero exposure. As such it is suggested to define "maximum exposure threshold", corresponding to the maximum value of PDX considered acceptable. Another crucial decision to make before computing PDX is whether to use the relaxed or the strict version. In most cases the relaxed version is better representative of the actual exposure, however the strict version could be useful in a context where personal data requires a very high level of security or highly motivated attackers could be present anywhere in the system.

4.8 Rule-based Flaw Detection

The fact that a *GDFD* is a particular variation of a directed graph can be also leveraged for detecting flaws. In particular it would be possible to find specific patterns in a *GDFD* which indicate a security flaw. Such patterns could be expressed in one of several patterning languages already available for use like the Acceleo Patterning Language or the Cypher Query Language and included in rules to automatically scan *GDFD*s and trigger alarms if a match is found.

Finding flaws by matching patterns has several advantages, but also limitations, both of which are shared with many other rule-based approaches. Such aspects related to the use or rules have already been studied and compared to statistical machine learning approaches in literature[48].

Between the main advantages there are:

- expressiveness and transparency of rules respect to "black box" solutions;
- easy maintenance;
- direct translation of domain knowledge in rules by experts;

Between the main limitation and shortcomings there are:

- the need of having experts learn the language to express rules;
- cost in terms of time and effort to manually craft rules;
- the fact that even minor changes in the data may require a different rule;

While not the focus of this thesis, the possibility of rule-based flaw detection is yet another opportunity opened by the use of *GDFD*s, which further proves the flexibility of this kind of diagrams.

Chapter 5

Designing Security Templates for the Italian National Framework for Cybersecurity and Data Protection

Security templates are particularly convenient when they are general in scope and highly re-usable. While any security objective can receive a corresponding Security Template, the subcategories of the Italian National Framework for Cybersecurity and Data Protection represent a fitting example of widely applicable security objectives.

Having Security Templates associated with the subcategories of a Cybersecurity Framework can provide additional insight on the security objective expressed by that subcategory in a way that a description provided by a string of text cannot express as effectively without being very long. Thus Security Templates can give users of the Framework a better understanding on how to reach the objective stated by the subcategory, while also highlighting hidden security pitfalls which may otherwise be ignored when only relying on text. Additionally Security Templates can assist the process of the assessment as they can be used to automatically generate controls via software. Hence instead of completely relying on the ability of the assessor to select appropriate controls for the set of selected Subcategories, a software can instead take as input all the Security Templates corresponding to the selected subcategories and, using a series of control extraction rules, present as output a list of controls from which to start the assessment, all in a fraction of the time a human assessor would take to write even a single control.

This chapter summarizes the work of designing Security Templates based on Subcategories of the Italian National Framework for Cybersecurity and Data Protection. Out of the total of 117 Subcategories provided in the most recent version of the Framework, 97 received a Security Template. Moreover between those 97, there are 91 Security Templates which provide an expansion of what the text of the corresponding subcategory states. On the other hand, the other subcategories received a Security Template which is more akin to a direct translation. Accounting all the previously discussed aspect together the net count is:

- 91 expanded subcategories (78%);
- 6 translated subcategories (5%);

- 20 subcategories not translatable in a *GDFD* Security Template (17%);

The overwhelming majority of Security Templates are unique, however there are ten pairs of templates that share a similar structure and meaning with each other. Those pairs are:

- ID.AM-1 and ID.AM-2, which are about keeping inventory of hardware and software;
- ID.AM-6 and DP-ID.AM-7, which are about roles for security and personal data treatment;
- ID.RA-4 and DP-ID.RA-7, which involve impact evaluation in the cases of business processes and personal data protection;
- ID.RM-1 and ID.SC-1, which are both about risk management processes but declined in the context of the organization and the supply chain respectively;
- PR.MA-1 and PR.MA-2, which model the processes of maintenance and remote maintenance;
- DE.DP-5 and PR.IP-7, describing the improvement loop for monitoring and protection processes;
- RS.MI-1 and RS.MI-2, which model the processes of containing the impact and mitigate effects after an incident occurs;
- RS.RP-1 and RC.RP-1, which are about the existence of a response and a recovery plan;
- RS.IM-1 and RC.IM-1, modeling the improvement cycle for incident response and recovery.

A complete overview of the number of Security Templates created for the subcategories of each Function of the Framework is presented by Table 5.1. All the graphical representations of the Security Templates can be found in Appendix A, while both the graphical and computer friendly versions are available on GitHub[1].

	Expanded	Translated	No Template
All Functions	91 (77.78%)	6 (5.13%)	20 (17.09%)
Identify	30 (81.08%)	3 (8.11%)	4 (10.81%)
Protect	27 (69.23%)	2 (5.13%)	10 (25.64%)
Detect	15 (83.33%)	0	3 (16.67%)
Respond	14 (82,35%)	1 (5.88%)	2 (11.76%)
Recover	5 (83,33%)	0	1 (16.67%)

Table 5.1: Number and percentage of subcategories with a Security Template

5.1 Identify Function

The categories grouped under the Identify Function revolve around establishing the context of the organization as well as finding critical assets, processes and risks associated with them. As

[1] `https://github.com/Fulvio-P/controlExtractor`

such the Security Templates defined for the subcategories in this function generally include many Security Metadata Flows and various Data Stores containing the inventories for all kinds of resources, vulnerabilities and threats. A notable exception is represented by the Security Templates related to the Data Management category, where it is possible to find many Personal Data Flows. That is because the subcategories grouped under that category mostly focus on the topic of personal data protection.

Of the 37 subcategories that fall under the Identify Function, 30 have a Security Template expanding the text and three are directly translated. Only four subcategories, accounting for around 11% of the total, do not have an associated Security Template, making this Function the one with proportionally the lowest percentage of subcategories missing Security Templates.

5.2 Protect Function

With a total of 39 subcategories the Protect Function is the most populated out of the functions present in the Italian National Framework for Cybersecurity and Data Protection. The security objectives specified by the members of this function are geared towards protection of all assets and processes, regardless of whether they are related to information technology or not. Because of this focus on protection, many of the Security Templates associated with subcategories in this function involve one or more Trust Areas. Also Security Metadata Flows are commonly used to inform processes enforcing access control or other security measures.

Between the 39 subcategories falling in this function, only 27 are expanded upon by the associated Security Template, two are translated directly, while ten did not receive any Security Template. With around 69% of Security Templates expanding on text and 25% of subcategories without Template, this function has the lowest percentage of expanded subcategories and the highest percentage of missing Security Templates. However it is easy to find the cause for this in the "Awareness and Training" category. Counting five subcategories, the "Awareness and Training" category focuses on people and their comprehension of their own roles, rather than on technological tools and procedures. As such none of the subcategories in this category has received a Security Template, noticeably impacting the above mentioned statistics. Removing those five subcategories makes the proportions of expanded subcategories and missing Security Templates very close to those of the other Functions being around 79% and 15% respectively.

5.3 Detect Function

The Detect Function contains 18 subcategories which are focused on the activities that help to spot information security incidents as soon as possible. Security Templates associated with subcategories in this function make ample use of Security Metadata Flows to inform Join Processes and Consume / Make processes that detect and analyze various types of events and incidents.

In this Function 15 of the 18 subcategories are expanded upon by the corresponding Security Template, while the remaining three subcategories are not associated with a Security Template. The absence of translated subcategories makes this function tied with the Recover Function for the

highest percentage of expanded subcategories at around 83% of the total. However it is important to note that the Recover Function only has six subcategories, making its high percentage arguably less relevant.

5.4 Respond Function

The subcategories in the Respond function express security objectives involving the course of action to take when an incident is detected in order to minimize the negative impact it could bring to the organization. Thus in the Security Templates associated subcategories in this Function it is common to find Security Metadata Flows carrying information about the incident which is then leveraged to perform critical decisions on the course of action, but also to improve the response plan itself and the overall security of the system.

There are 17 subcategories in the Respond Function, and 14 of them received a Security Template which expands the meaning of the text. Of the remaining subcategories only a single one is translated literally, while the last two do not have a corresponding Security Template.

5.5 Recover Function

The Recover function includes security objectives for all those activities geared towards resuming normal operations after an incident by restoring impacted processes, assets and services. As within the Security Templates related with subcategories in the Respond function, also in those developed for the subcategories in the Recover Function is common to spot Security Metadata Flows carrying information about the incidents that have just happened. This happens for similar reasons, as that information regarding the incident is needed to guide the recovery process and improve security.

Between all the Functions defined in the Framework, the Recover Function is the least populated, counting only six subcategories. However of those six only one is not translatable with a Security Template, while all the remaining five receive a Template which expands upon the text description

5.6 Data Protection Subcategories

One of the distinguishing features of the Italian National Framework for Cybersecurity and Data Protection is the inclusion of subcategories that have a particular focus on data protection. Such subcategories are distinguished by the acronym "DP" before the name of the subcategory, introducing an orthogonal classification which is independent from the division of categories and subcategories. In the version 2.0 of the Framework there are a total of nine subcategories focused on data protection, eight of which are part of the "Identify Function" while the remaining one is part of the "Respond Function".

None of the data protection focused subcategories was left without a corresponding security template. In particular seven out of nine subcategories were expanded, while the other two were translated directly into a *GDFD*. The percentage of expanded subcategories is around 78% and exactly the same as the percentage of expanded subcategories considering all functions at once.

Predictably, a common feature across the security template of subcategories listed with the "DP" prefix is the presence of Personal Data Flows reflecting the presence of personal data that need to be protected and kept private in accordance with strict regulations.

5.7 Security Templates in JSON Format

During the process of their creation, all Security Templates *GDFD*s for the Italian National Framework for Cybersecurity and Data Protection were manually drawn in their graphical representation. While the graphical format is much more readable from a human perspective, it becomes unwieldy to work with when leveraging programming languages to automate the work. For this reason all the Security Templates created for the Framework also have a translation in the *JSON* format.

From the point of view of a programmer it is possible to think about each Security Template as a directed multigraph where the Data Flows are edges, while Processes, Entities and Data Stores are vertices. As such, in the *JSON* format each Security Template is represented as a directed multigraph. In particular the representation employed for translating the diagrams into *JSON* files follows the node-link format which contains a list of "node" objects describing Processes, Entities and Data Stores, followed by a list of "link" objects representing Data Flows and, if present, also a list of "area" objects corresponding to the Areas in the *GDFD*. An example of Security Template in *JSON* format is reported below.

```
1  {
2      "nodes":[
3          {"id": "Service Providers", "type": "Entity", "flags": "", "area":
           ↪  "A1"},
4          {"id": "Monitor Activities", "type": "Process", "behavior": "F",
           ↪  "flags": "", "area": ""},
5          {"id": "Define Baselines", "type": "Process", "behavior": "F", "flags":
           ↪  "", "area": ""},
6          {"id": "Detect Cybersecurity Events", "type": "Process", "behavior":
           ↪  "M", "flags": "", "area": ""},
7          {"id": "Service Providers Baselines", "type": "Data Store", "flags":
           ↪  "CIA", "area": ""}
8      ],
9      "links":[
10         {"source": "Service Providers", "target": "Monitor Activities",
           ↪  "label": "Activities", "type": "Generic", "flags": "", "area":
           ↪  "A1"},
11         {"source": "Service Providers", "target": "Detect Cybersecurity
           ↪  Events", "label": "Activities", "type": "Generic", "flags": "",
           ↪  "area": "A1"},
12         {"source": "Monitor Activities", "target": "Define Baselines", "label":
           ↪  "Typical Activities", "type": "Metadata", "flags": "CIA", "area":
           ↪  ""},
```

```
13          {"source": "Define Baselines", "target": "Service Providers Baselines",
            ↪  "label": "Baselines", "type": "Metadata", "flags": "CIA", "area":
            ↪  ""},
14          {"source": "Service Providers Baselines", "target": "Detect
            ↪  Cybersecurity Events", "label": "Baselines", "type": "Metadata",
            ↪  "flags": "CIA", "area": ""},
15          {"source": "Detect Cybersecurity Events", "target": "Service
            ↪  Providers", "label": "Detection Alert", "type": "Metadata",
            ↪  "flags": "CIA", "area": "A1"}
16      ],
17      "areas":[
18          {"id": "A1", "name": "Trusted Path", "type":"Trust Area"}
19      ]
20  }
```

The formatting of the data for the *JSON* representation of Security Templates was inspired by the "Force Directed Graph" example provided on the D3.js website [2]. This kind of formatting was chosen in an effort to align the *JSON* representation of *GDFD*s to a widely adopted format which would be familiar and easy to understand to most developers. One of the effects of representing the Security Templates in this way, is making them compatible with the Network Analysis Python package NetworkX[18] which can be used to perform complex operations on graphs. In particular the node_link_graph method provided by the package can be used to automatically import the structure of the templates as a directed multigraph without the need of writing custom code. However it is important to keep in mind that the additional data such as the types of the Elements or the information about Areas is not automatically imported in that way.

5.8 Supporting Automated Control Extraction

The *JSON* format is easily readable by software, with most modern programming languages providing libraries for easily opening and parsing data saved in *JSON* files. As such, having all the Security Templates available in this format unlocks many opportunities for automation. In particular one of the aspects which can be automated is the process of extracting controls from Security Templates by using the rules presented in Section 4.5

This section is dedicated to the presentation of controlExtractor, a tool written in Python which is capable of performing automated control extraction applying all or a subset of the aforementioned rules for control extraction. The main objectives behind controlExtractor are:

- providing a proof of concept software with *GDFD*s in *JSON* format;
- showing how the control extraction rules can be used to extract controls form Security Templates;

[2] https://bl.ocks.org/mbostock/4062045

- presenting a possible use case for Security Templates associated to the subcategories of the Italian National Framework for Cybersecurity and Data Protection;

The controlExtractor tool requires two strings in input in order to run. The first is the path of a directory containing all the Security Templates which the tool needs to process. However the *JSON* files defining the Security Templates should not be placed directly in that directory. The tool expects to find inside the path in input directories named after the functions of the Framework, each containing directories with names corresponding to categories which are part of that function. The *JSON* files encoding the Security Templates should be placed inside the directory corresponding to the correct category and be named after the corresponding subcategory. Any missing Framework function, category or subcategory will be automatically skipped. The second required input string is the name of the Excel file, in xlsx format, which will be produced as the output.

In addition to the two strings in input the controlExtractor tool expects to find in its location a file called "config.JSON". This file should contain a single *JSON* objects with several fields, each corresponding to a control extraction rule. By editing the values for each field it is possible to change the set of rules that will be used to extract controls. All rules which have 'true' as a value will be applied, while those having value 'false' will be skipped.

The output produced by the controlExtractor tool is an Excel file, containing a single spreadsheet. The first column contains the names of Framework functions, the second one contains the names of categories, the third one contains the name of subcategories and finally the fourth column contains the text of the controls. The cells in the spreadsheet are ordered and merged so that the cell containing the name of the function corresponds to all the cells of the subcategories in that function. In the same way the cell containing the name of each category corresponds to the cells of all subcategories in that category and the cell of each subcategory corresponds to the cells of all controls extracted from the associated Security Template.

By default each control in the output receives a unique identifier which is placed at the start of its text, however that behavior can be deactivated by running the controlExtractor tool with the '-n' flag as an additional input. This feature is useful to make it easier to identify duplicate controls in the output.

If a Security Template does not generate any controls, the cell that would be occupied by the control in the output will be filled with the string "N/A". This behavior also allows to exclude a set of subcategories and still make them appear in the output by giving them empty Security Templates (i.e. Security Templates where the list of nodes and links are both empty).

The code for controlExtractor is organized in two Python files: "controlExtractor.py" and "rules.py". In particular the latter contains the core of the logic of the application, since it contains the definition for all the functions corresponding to the control extraction rules. On the other hand the code contained in "controlExtractor.py" concerns mostly the interpretation of user inputs and the creation of the output.

The entire source code for controlExtractor, alongside all the other material produced for this thesis, is made publicly available on GitHub[3]. Two external Python packages are required for it to run: openpyxl[4], which is required to edit xlsx files, and natsort[5], which is used to sort the names of categories and subcategories.

[3] https://github.com/Fulvio-P/controlExtractor
[4] https://pypi.org/project/openpyxl/
[5] https://pypi.org/project/natsort/

Chapter 6

Evaluation

This chapters reports and evaluates the results obtained from the usage of *GDFD*s, with particular reference to the process of automated control extraction supported by the controlExtractor tool taking as input Security Templates defined on the Italian National Framework for Cybersecurity and Data Protection, presented in Chapter 5, and the basic rules for control extraction presented in Section 4.5.

Initially it is considered the output of the controlExtractor tool when applied to all subcategories and with all rules active at the same time. This setting provides the maximum amount of controls that the tool is capable of generating given the current versions of the Security Templates and control extraction rules. The analysis of this "maximal" output is split in two phases. In a first quantitative phase, the number of controls generated is presented and evaluated. In the second phase, the analysis of controls proceeds with a qualitative approach, in search for common traits that set apart automatically generated controls from those created manually.

Later sections in this chapter compare the controls obtained by controlExtractor on two subsets of subcategories with manually crafted controls for those same subcategories. The sources for the aforementioned sets of manually created controls are:

- the *GDPR* contextualization prototype provided by the Italian National Framework for Cybersecurity and Data Protection;
- a real assessment executed in 2017 from *CIS* Sapienza members;

Since the assessment is from 2017, it uses the previous version of the Framework (i.e. version 1.0). Hence the assessment lacks the subcategories added in the current version 2.0. From a compatibility perspective that is not an issue, since the Italian National Framework for Cybersecurity and Data Protection is perfectly backwards compatible with its previous versions. Thus it is sufficient to just change the subset of Security Templates in input to reflect the missing subcategories and controlExtractor will work without any issues. On the other hand, controls related to those missing subcategories should be evaluated and compared to manually crafted ones. Fortunately most of the subcategories introduced in the second version of the Framework received manually crafted controls in the *GDPR* contextualization prototype. For this reason the two cases are complementary nature, which allows automatically generated controls to be compared with manually created ones across a wide variety of subcategories.

6.1 Quantitative Analysis

Counting the number of generated controls is one of the most straightforward tests for gauging the effectiveness of the automatic control extraction made possible by the Security Templates for the Italian National Framework for Cybersecurity and Data Protection and the controlExtractor tool. A "maximal output" can be obtained by letting controlExtractor apply all the rules for control extraction listed in Section 4.5 to all the 97 Security Templates corresponding to the subcategories of the Framework.

The "maximal output" is a large spreadsheet with controls organized by the corresponding function category and subcategory. In total the number of unique controls generated by applying all the rules is 1292. However in the output several controls are repeated multiple times, as the controlExtractor tool is programmed to output the most complete, but at the same time verbose, set of controls. Most duplicates are created when a Node or a Data Flow is encountered several times by the tool, which can happen easily especially with those *GDFD* elements which are contained in several different Security Templates. For example the Data Store named "Vulnerability Inventory" is encountered multiple times across Security Templates modeling objectives related to the topics of risk assessment, incident prevention and incident response. Thus the control extractor will create the control requiring the existence of a "Vulnerability Inventory" Data Store, as well as those corresponding to its CIA requirements, each time that this Data Store appears in a Security Template.

The presence of duplicates, however, can be easily bypassed when selecting controls. A simple software could be used to present to the human assessor only unique controls and then report the input provided by said assessor also to all other copies of the control automatically. Then, after its first appearance, any subsequent occurrence of the same control will be skipped as it has already been examined.

Yet, even excluding duplicates by counting only the same occurrence of each control, the number of controls generated by the controlExtractor tool is still an whole order of magnitude greater than the number of controls which would be typically created manually by a human expert. However not all controls are generated equal, as the relationships between the various elements present in Security Templates can be leveraged to create a hierarchy between controls generated with different rules. Then this hierarchy can be leveraged to cut the number of controls presented to human assessors.

As a general rule in *GDFD*s, and thus in all Security Templates, there are not any "disconnected" Nodes (i.e. Processes, Data Stores and Entities which do not receive in input, nor send as output any Data Flow). Even if there were, "disconnected" Nodes would be of any interest in a diagram focused on representing the flow of information between interconnected elements. The absence of Nodes which do not interact with any data flow, means that each Node is at least once the source or the destination of at least one Data Flow. The opposite also is true, as all Data Flows must have one source and one destination in order to exist, hence all Data Flows appear as either input or output of a Node.

By taking in consideration of the above mentioned relationships, it is clear that, if a Node does not exist, neither do all the Data Flows which have that Node as either their source or their destination. Thus if the control requiring the existence of a Node is not selected, it is useless to present to the assessor those controls requiring the existence of Data Flows which are received as input or sent as output by that Node. A similar logic can be applied to controls derived from the Confidentiality, Availability, Integrity and Privacy requirements of both Nodes and Data Flows. If either type of Element does not exist then the controls related to its security requirements should be skipped.

On the other hand when considering controls generated by Areas, requiring protection for the borders of a Trust Area only makes sense if that Trust Area exists. Similarly, each control generated by a Data Flow crossing the boundaries of an Area, either entering or exiting from it, requires the existence of both that Area and the Data Flow in order to be meaningful. Moreover the existence of an Area makes sense only if there is at least one Data Flow or Node inside it.

By combining all the above mentioned requirements together, it is possible to obtain a hierarchy with the controls requiring the existence of Node at the top. Such hierarchy can be then leveraged to present to the assessor a smaller number of controls by presenting first the controls requiring the existence of nodes. Then it is possible to present the controls derived from CIA requirements of Nodes, skipping those Nodes which do not exist. After the controls requiring the existence of Nodes it is also possible to present the controls requiring the existence of Data Flows, but skipping those having as source or destination Nodes that do not exist. Knowing which Data Flows exist, it is then possible to check all those controls regarding Confidentiality, Availability, Integrity and Privacy of Data Flows, avoiding those corresponding to non-existent Data Flows. At the same time it is possible to consider those controls requiring the existence of Areas for all those Areas containing at least one existing Node or Data Flow. Finally the controls requiring protection of Trust Areas can be presented only for Areas that exist, while controls generated by Data Flows crossing the boundaries of Areas are presented only if both the Area and the Data Flow exist.

The hierarchy between the controls generated by all the different rule is visualized in Figure 6.1. Each block in the figure corresponds to a set of rules for control extraction. A control extracted with a rule inside any of the blocks requires the implementation of a corresponding control extracted with the rules inside all the blocks above. The controls extracted with the rule "the Area exists" require at least the existence of one Element inside that Area. This special relationship is depicted with the splitting arrow in the top right corner of the figure.

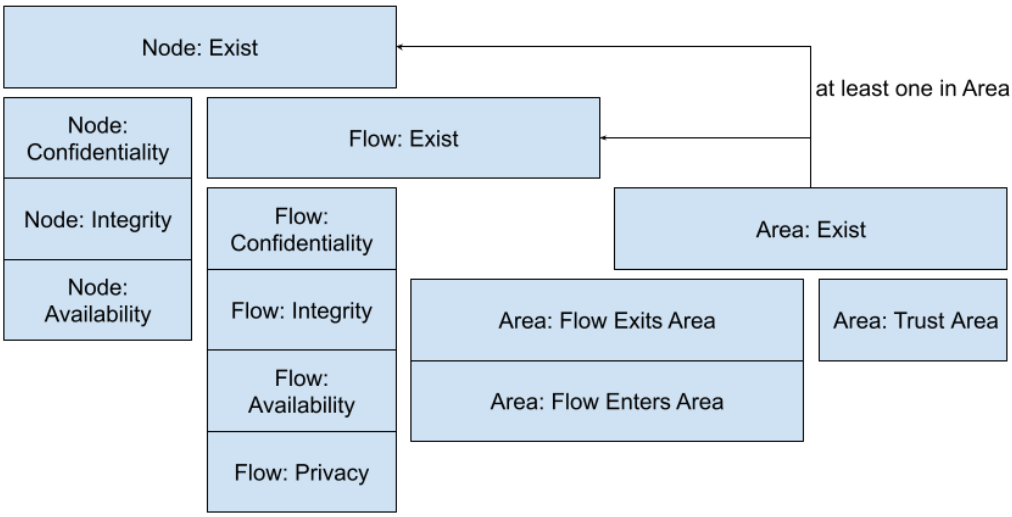


Figure 6.1: Hierarchy between controls, based on generation rule

As an example consider the following controls extracted from the Security Template corresponding to the first subcategory in the framework ID.AM-1:

- **CTR-ID.AM-1-1**: There is a "Install Device" Process. It has the following Data Flows in input: 'Accountability Info', 'Device'. It has the following Data Flows in output: 'Installation Info'.
- **CTR-ID.AM-1-9**: There is a "Accountability Info" Data Flow, with type "Metadata" flowing from the Node "User" to the Node "Install Device".
- **CTR-ID.AM-1-10**: There is a "Device" Data Flow, with type "Generic" flowing from the Node "Check Authorization" to the Node "Install Device".
- **CTR-ID.AM-1-11**: There is a "Installation Info" Data Flow, with type "Metadata" flowing from the Node "Install Device" to the Node "Update Inventory".

If the control CTR-ID.AM-1-1 is not implemented, there is no Process for the installation of devices. Thus also all the Data Flows which should be received as input or produced as output by that Process do not exist, as it is not possible to have a Data Flow without a source or without a destination. Hence the controls CTR-ID.AM-1-9, CTR-ID.AM-1-10 and CTR-ID.AM-1-11 can be automatically marked as not implemented.

This process gets then repeated towards lower levels of the hierarchy. For example as the data flow required by CTR-ID.AM-1-11 does not exist, also the following controls can be safely skipped:

- **CTR-ID.AM-1-15**: There is a security measure preserving Confidentiality for the Data Flow with label "Installation Info".
- **CTR-ID.AM-1-21**: There is a security measure preserving Integrity for the Data Flow with label "Installation Info".
- **CTR-ID.AM-1-25**: There is a security measure preserving Availability for the Data Flow with label "Installation Info".

By following the hierarchy described above, the number of controls that will always be presented to the assessor corresponds only to the number of unique controls derived by the existence of Processes and Data Stores. When instructing the controlExtractor tool to only use that rule for extracting controls from all the 97 Security Templates defined for the Italian National Framework for Cybersecurity and Data Protection produces 306 unique controls. That number, however, does not correspond to the minimum number of controls from which to start, as usually only some of the subcategories are selected depending on the scope of the assessment.

In case the number of controls is still deemed too high, it is possible to decrease the number of controls generated by the tool even further by sacrificing the level of granularity. For example by combining controls generated by examining the security requirements of Processes, Data Stores and Data Flows into a single one per element, it is possible to decrease the number of those kinds of controls by almost 300, since many elements requiring protection from security measures have more than one security requirement active. However it is not recommended to compromise too much on granularity, as achieving a finer level of granularity is one of the objectives of automated control extraction.

6.2 Qualitative Analysis

This section is dedicated to the evaluation of the text of the controls generated by the controlExtractor tool. The objective is to find traits common to all generated controls, discerning what they share with manually created controls, and, most importantly, what sets them apart. Finding strong differences between manually created controls and automatically generated controls is both expected and desired. That is the case because automatically generated controls are not meant to replace manually created one, but rather complement and enrich them, with the goal of making the assessment more objective.

One of the most noticeable features of automatically generated controls is their simplicity, given by their restricted scope. As it was possible to infer by looking at the rules for control extraction used by the controlExtractor tool, each generated control is focused on a single, or at most two elements of a Security Template. This feature is favorable as it represents a difference which well complements human generated controls, which often have, instead, a larger in scope and can include complex and abstract concepts, such as the compliance with laws and regulations. On the other

hand the much simpler automatically generated controls are specific and could be seen as "atomic" when compared to controls defined by humans. In this perspective a collection of controls generated by controlExtractor may correspond to intermediate steps or prerequisites in order to implement human-made controls.

Another feature, which is especially evident in all those controls generated by rules checking for security requirements of the elements in the Security Template, is a very high level of granularity. Each Data Flow, Process and Data Store will receive an appropriate control if it requires a security measure protecting Confidentiality, Availability or Integrity. In addition each Personal Data Flow will generate controls requiring the presence of a security measure protecting the privacy of the personal information it carries. This level of granularity may capture security aspects which could be otherwise overlooked or taken for granted by a human expert. Moreover, even if this kind of requirement is included in manually created controls, it would be most likely defined for an entire class of elements. On the other hand the controlExtractor tool will generate controls specific to each element, decreasing the risk of overlooking the security requirements of less obvious members of a certain class.

6.3 Comparison with the GDPR prototype

In this section controls generated automatically by controlExtractor are compared with those provided alongside the *GDPR* contextualization prototype. In order to perform a fair comparison, the tool receives as input only those subcategories receiving one or more controls in the contextualization prototype. Those categories are the ones with class "Mandatory" (i.e. "Obbligatoria") and a high level of priority.

In total the number of manually crafted controls provided with the contextualization is 41. On the other hand the controlExtractor tool generates 150 unique controls. However when leveraging the hierarchy and starting by considering only those controls related to the existence of Nodes the number is decreased to 28, which is even less than the number of human-made controls.

As for the case of the assessment, also between the controls for the *GDPR* contextualization prototype it is possible to find controls generated by software which are "atomic" with respect to those created by humans. For example consider the subcategory ID.DM-2: "Sono definiti, implementati e documentati i processi riguardanti l'informazione dell'interessato in merito al trattamento dei dati." which receives only the following control.

IT: "Il titolare del trattamento adotta processi per fornire all'interessato tutte le informazioni inerenti il trattamento dei dati personali comunque raccolti (sia presso lo stesso interessato che presso terzi) nonché l'esistenza e le modalità di esercizio dei diritti previsti dal Regolamento."

EN: "The data controller shall adopt processes to provide the data subject with all the information inherent in the processing of personal data however collected (whether from the data subject himself or from third parties) as well as the existence and methods of exercising the rights provided for in the Regulations."

The controlExtractor tool generates controls which correspond to a partition of the manually created controls in more technical and granular steps, as it can be seen by examining the text of the controls:

- **CTR-ID.DM-2-DP-0**: There is a "Policies Storage" Data Store. It has the following Data Flows in output: 'Data Treatment Policy'.
- **CTR-ID.DM-2-DP-1**: There is a "Inform About Data Treatment" Process. It has the following Data Flows in input: 'Data Treatment Policy'. It has the following Data Flows in output: 'Data Treatment Policy'.
- **CTR-ID.DM-2-DP-2**: There is a "Data Treatment Policy" Data Flow, with type "Generic" flowing from the Node "Policies Storage" to the Node "Inform About Data Treatment".
- **CTR-ID.DM-2-DP-3**: There is a "Data Treatment Policy" Data Flow, with type "Generic" flowing from the Node "Inform About Data Treatment" to the Node "Data Subject".

In addition, controls for the subcategories selected in the *GDPR* contextualization prototype provide several examples in which the different level of granularity of software generated controls and human made controls makes them complement each other. While human generated controls often operate in a higher level of abstraction and thus can deal with laws and regulations, controls generated from Security Templates are related to technical details. This is evident in the controls assigned to the subcategory ID.DM-5: "Sono definiti, implementati e documentati i processi di trasferimento dei dati in ambito internazionale.".

While the three manually created controls provided in the contextualization prototype are focused on compliance with the articles of the *GDPR*, those generated by the controlExtractor tool deal specifically with dataflows leaving the Area defined by the borders of the country, making sure that there are security measures in place to protect the security requirements associated with the data traveling internationally.

6.4 Comparison with a Manual Assessment

In this section the controls generated by the controlExtractor tool are compared with those present in a real assessment from 2017. The assessment is older than the current version of the Framework, however the version 2.0 is completely backwards compatible with the version 1.0 used for the assessment. For sake of comparing controls, the Security Templates fed as input to the controlExtractor tool are limited to those corresponding to the subcategories present in the assessment.

Running controlExtractor with all the rules active on the above-mentioned subset of Security Templates produces 1079 unique controls. The number of automatically generated controls is very high when compared to the number of controls present in the assessment which are only 244. However by leveraging the hierarchy of controls presented in Section 6.1 it is possible to start by only considering the controls generated by the rule requiring the existence of Processes and Data Stores. Hence the number of unique controls from which the assessment starts is reduced to 277,

indicating that the actual number of controls to manage could be not much higher than the number of manually crafted controls.

Comparing the text of the controls generated by software and those manually created by humans, it is possible to notice the different scope and level of granularity. Take for example the controls relative to the subcategory RS.MI-1: "In caso di incidente vengono messe in atto procedure atte a contenerne l'impatto". In the assessment this subcategory receives only one and very broad control.

IT: "In caso di incidente viene tempestivamente applicato il piano di risposta associato per contenerne l'impatto e/o mitigarne gli effetti."
EN: "In the event of an incident, the associated response plan is promptly applied to contain its impact and/or mitigate its effects."

In contrast to the broad and vague human-created control present in the assessment, the controlExtractor tool generates a larger set of 32 unique controls. Among those, five are on top of the hierarchy being extracted by following the rule that each Process and Data Store in the Security Template exists.

A subset of the software-generated controls can be considered "atomic" with respect to the man-made control, defining basic elements which can be implemented to reach the security objective described by the text of the subcategory. These controls are:

- **CTR-RS.MI-1-0**: There is a "Monitoring and Detection" Process. It has the following Data Flows in output: 'Incident Damage', 'Incident Alert', 'Incident'.
- **CTR-RS.MI-1-1**: There is a "Respond" Process. It has the following Data Flows in input: 'Incident Alert', 'Response Plan'. It has the following Data Flows in output: 'Impact Reduction'.
- **CTR-RS.MI-1-4**: There is a "Response Plans Inventory" Data Store. It has the following Data Flows in output: 'Response Plan'.
- **CTR-RS.MI-1-12**: There is a "Incident Alert" Data Flow, with type "Metadata" flowing from the Node "Monitoring and Detection" to the Node "Respond".
- **CTR-RS.MI-1-13**: There is a "Response Plan" Data Flow, with type "Metadata" flowing from the Node "Response Plans Inventory" to the Node "Respond".
- **CTR-RS.MI-1-15**: There is a "Impact Reduction" Data Flow, with type "Metadata" flowing from the Node "Respond" to the Node "Identify Impact".

In addition many of the other controls guarantee that several fine grained security objectives are met. For example the "Response Plan" Data Flow contains critical information and there could be serious consequences if it was to be intercepted, modified or otherwise distupted by an adversary. As such several controls focused on protecting that specific Data Flow are extracted from the Security Template, namely:

- **CTR-RS.MI-1-19**: There is a security measure preserving Confidentiality for the Data Flow with label "Response Plan".
- **CTR-RS.MI-1-25**: There is a security measure preserving Integrity for the Data Flow with label "Response Plan".
- **CTR-RS.MI-1-31**: There is a security measure preserving Availability for the Data Flow with label "Response Plan".

6.5 Discussion and Limitations

The previous sections provided tangible evidence on how the controlExtractor tool and the set of Security Templates defined for the Italian National Framework for Cybersecurity and Data Protection perform when compared with manually created controls on the same subcategories. Hence it is now possible to discuss the results and summarize the positive and negative aspects of using automated control extraction software.

One positive aspect of automatically generated controls is how they complement manually created controls. In most cases the controls created by controlExtractor position themselves either as "atomic" with respect to human made controls or cover specific elements that might otherwise be missed. In the former case, automatically created controls assists those created manually by specifying intermediate steps to reach the full extent of the manually defined control. In the latter, the increased granularity of the control generation allows each element appearing in the Security Templates to get its own set of controls, made to exactly fit its security requirements.

The requirements for automated control extraction are low both in terms of time and resources. Selecting a subset of subcategories is as easy as deleting the unwanted ones from the directories containing the various Security Templates, while the rules for control extraction can be activated and deactivated by editing a simple configuration file. The running controlGenerator only takes a fraction of a second, even when all rules are active and all Security Templates are selected. On the other hand customizing the security templates and the rules for control extraction could require a moderate amount of time, but they are both designed to be easily expandable and customizable.

Being based on rules, the automated control extraction proposed in this thesis also inherits all the advantages and disadvantages of rule-based approaches, which are already documented in literature [48]. In particular, the reliance on rules intrinsically links the quality of the controls generated by this approach to the quality of the rules employed. However, being aware of the limitation brought by the use of rules, guided the design of *GDFD*s towards a simple and intuitive structure, in an effort to overcome the limitation given by the difficulty of designing and reusing rules.

Aside from the intrinsic limitations given by the rule-based design of the automatic control extraction process, there are also some easier to overcome limitations which are related to the current state of the work presented in this thesis. In particular, the rules currently employed for control extraction are simple and mostly focused on existence and security requirements. This

makes them easy to understand and shows good results, however they might be integrated with more complex rules which consider several elements of the Security Templates at the same time. Moreover the methodology currently used to lower the number of controls from which to start the assessment is effective, but also coarse-grained, since it prioritizes controls only based on the rule that generated them. It could be possible to develop different approaches which would present a more curated selection of controls to the human assessor.

Finally the focus on automated control extraction can be seen as a limitation of this work, as there are other aspects of the use of *GDFD*s which could receive additional attention and further studies, namely: the effectiveness of the security metrics defined on *GDFD* elements, the ability of Security Templates to support the creation of customized *GDFD*s and the possibility of detecting automatically security flaws in those customized *GDFD*s.

Chapter 7

Conclusion

In the field of Information Security Governance, cybersecurity frameworks operate at a very high level of abstraction. As such, security assessment can be easily impacted by human errors and bias. In this context, given the lack of automated and data-driven processes, the experience of those in charge of defining security controls is often a determining factor for their quality and completeness. On the other side of the spectrum, in-depth security analysis processes, such as penetration testing or the creation of attack graphs, are expensive to perform and often produce results which are too technical and detailed to be used for security management at a large scale. The field of Data-Driven Security Governance strives to support security decision making with technical data at a "middle" level of abstraction, leveraging automation and data-driven processes whenever possible, with the goal of reaching more reliable and objective results.

In this thesis first I collected a number of studies proposing the use of metrics and technical data in various ways to support decision making related to Information Security Governance topics. Then I have organized the various proposals contained in the examined literature according to their level of automation.

By learning from the collected literature, I later presented a first specification for a new extended version of the *DFD*, named *GDFD*. This new kind of diagram was designed with the goal of supporting Data-Driven Security Governance processes. Several use cases were identified, namely: automatically extracting security controls with a set of simple rules, the definition of Security Templates describing a possible way to achieve a given security objective, the ability to compute of two kinds of security metrics and finally the ability to automatically detect patterns indicating security flaws in a diagram.

To support the Italian National Framework for Cybersecurity and Data Protection, a total of 97 Security Templates were created, corresponding to all the subcategories of the Framework which allowed this kind of treatment. In addition I developed controlExtractor, a tool capable of scanning those Security Templates and extracting automatically a number of controls from each of them.

The output of controlExtractor was later evaluated from both a quantitative and qualitative perspective, examining both the number and the text of the extracted controls. In addition controls extracted in this way were compared to manually created ones taken from the *GDPR* contextualization prototype for the Italian National Framework for Cybersecurity and Data Protection, as well as to those found in a security assessment performed by *CIS* Sapienza members.

The comparison showed that the automatically extracted controls achieve their goal of being,

under several aspects, complementary to those manually created by humans. Hence their combined use would allow for more objective and comprehensive assessments when using the Framework.

This thesis provided two key contributions to the field of Data-Driven Security Governance. The first being the definition of *GDFD*s and the presentation of different ways they can be leveraged as a "middle layer" between abstract assessments and technical data. The second is the creation of Security Templates for the Italian National Framework for Cybersecurity and Data Protection, both in graphical and *JSON* format, along with the controlExtractor tool which allows to automatically apply control extraction rules to those Security Templates.

All the material produced for this thesis, which includes all Security Templates, both in graphical and *JSON* format, the controlExtactor tool and all its source code, is available on GitHub[1]. This makes it easy to download, customize, use and add contributions to the tool, its rules for control extraction and Security Templates.

Despite the work done, there are still aspects of *GDFD*s which will need to be properly tested and evaluated such as the effectiveness of security metrics defined on *GDFD*s, the accuracy of rule-based flaw detection and the ability of using Security Templates to support the creation of highly customized *GDFD*s to model specific systems and/or security requirements.

As future work I envision, on one hand, the development of more tools leveraging *GDFD*s, and, on the other hand, the expansion of *GDFD*s themselves with additional functionalities.

The definition of formal tools to perform a quantitative comparison between different *GDFD*s could expand even further their functionalities. Having the ability to compute quantitatively the differences and similarities between two *GDFD*s, for example, could support a process of gap analysis between a current state and a target state, both modeled as *GDFD*s. Moreover, being able to measure the effects on security provoked by the modification, introduction or elimination of elements in a *GDFD* could be used to prioritize actions to perform on the current state of a system depending on their impact on security.

Another possible direction for future work is represented by the adaptation of *GDFD*s to make them compatible with different security frameworks and standards. In their current state, given the similarities between the Italian National Framework for Cybersecurity and Data Protection and the *NIST* Cybersecurity Framework, Security Templates and the controlExtractor tool should also work with the *NIST* Cybersecurity Framework with little to no modifications. On the other hand there are several other widely recognized standards and frameworks, such as the ISO 27000 family of standards and the ENISA Cybersecurity Framework, which are structured in different ways and are not compatible with the use of *GDFD*s yet.

Additional ideas for future work include the development and evaluation of software made to assists or even automate the creation of *GDFD*s, as well as software for automatic flaws detection, able to recognize dangerous patterns in *GDFD*s.

[1] `https://github.com/Fulvio-P/controlExtractor`

Bibliography

[1] F. Alghamdi, N. Hamza, and M. Tamimi. Factors that influence the adoption of information security on requirement phase for custom-made software at smes. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–6. IEEE, 2019.

[2] S. AlGhamdi, K. T. Win, and E. Vlahu-Gjorgievska. Information security governance challenges and critical success factors: Systematic review. *Computers & Security*, 99:102030, 2020.

[3] M. Angelini, S. Bonomi, C. Ciccotelli, and A. Palma. Toward a context-aware methodology for information security governance assessment validation. In *International Workshop on Cyber-Physical Security for Critical Infrastructures Protection*, pages 171–187. Springer, 2020.

[4] M. Angelini, C. Ciccotelli, L. Franchina, A. Marchetti-Spaccamela, and L. Querzoni. Italian national framework for cybersecurity and data protection. In *Annual Privacy Forum*, pages 127–142. Springer, 2020.

[5] R. Baldoni and L. Montanari. Italian national cyber security framework. In *Proceedings of the International Conference on Security and Management (SAM)*, page 168. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2016.

[6] S. Barnum. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation*, 11:1–22, 2012.

[7] M. Barrett. Framework for improving critical infrastructure cybersecurity version 1.1, 2018-04-16 2018.

[8] B. J. Berger, K. Sohr, and R. Koschke. Automatically extracting threats from extended data flow diagrams. In *International Symposium on Engineering Secure Software and Systems*, pages 56–71. Springer, 2016.

[9] P. Bowen, J. Hash, and M. Wilson. Information security handbook: a guide for managers. In *NIST SPECIAL PUBLICATION 800-100, NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY*, 2007.

[10] S.-C. Cha and K.-H. Yeh. A data-driven security risk assessment scheme for personal data protection. *IEEE Access*, 6:50510–50517, 2018.

[11] J. Connolly, M. Davidson, and C. Schmidt. The trusted automated exchange of indicator information (taxii). *The MITRE Corporation*, pages 1–20, 2014.

[12] G. Csertán, G. Huszerl, I. Majzik, Z. Pap, A. Pataricza, and D. Varró. Viatra-visual automated transformations for formal verification and validation of uml models. In *Proceedings 17th IEEE International Conference on Automated Software Engineering,*, pages 267–270. IEEE, 2002.

[13] T. DeMarco. Structure analysis and system specification. In *Pioneers and Their Contributions to Software Engineering*, pages 255–288. Springer, 1979.

[14] E. Doynikova, A. Fedorchenko, and I. Kotenko. Ontology of metrics for cyber security assessment. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–8, 2019.

[15] M. Evans, Y. He, L. Maglaras, and H. Janicke. Heart-is: A novel technique for evaluating human error-related information security incidents. *Computers & Security*, 80:74–89, 2019.

[16] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, and A. Taylor. Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1433–1445, 2018.

[17] J. Freund and J. Jones. *Measuring and managing information risk: a FAIR approach.* Butterworth-Heinemann, 2014.

[18] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[19] M. Haque, R. Krishnan, et al. Toward automated cyber defense with secure sharing of structured cyber threat intelligence. *Information Systems Frontiers*, 23(4):883–896, 2021.

[20] S. Hina, D. D. D. P. Selvam, and P. B. Lowry. Institutional governance and protection motivation: Theoretical insights into shaping employees' security compliance behavior in higher education institutions in the developing world. *Computers & Security*, 87:101594, 2019.

[21] R. Ibrahim and S. Y. Yen. An automatic tool for checking consistency between data flow diagrams (dfds). *World Academy of Science, Engineering and Technology*, 69:2010, 2010.

[22] Z. Iqbal, Z. Anwar, and R. Mumtaz. Stixgen-a novel framework for automatic generation of structured cyber threat information. In *2018 International Conference on Frontiers of Information Technology (FIT)*, pages 241–246. IEEE, 2018.

[23] F. Irhamn and D. Siahaan. Object-oriented data flow diagram similarity measurement using greedy algorithm. In *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, volume 1, pages 274–278. IEEE, 2019.

[24] M. ISO. Iso/iec 38500: 2015 information technology–governance of it for the organization. *International Organization for Standardization*, 2015.

[25] G. W. Jackson and S. S. Rahman. Security governance, management and strategic alignment via capabilities. In *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 44–49. IEEE, 2017.

[26] C. Jiang, F. Coenen, and M. Zito. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(1):75–105, 2013.

[27] K. Kandasamy, S. Srinivas, K. Achuthan, and V. P. Rangan. Iot cyber risk: A holistic analysis of cyber risk assessment frameworks, risk vectors, and risk ranking process. *EURASIP Journal on Information Security*, 2020(1):1–18, 2020.

[28] L. Kohnfelder and P. Garg. The threats to our products. *Microsoft Interface, Microsoft Corporation*, 33, 1999.

[29] R. Leszczyna. Review of cybersecurity assessment methods: Applicability perspective. *Computers & Security*, 108:102376, 2021.

[30] X. Liu, H. Pan, M. He, Y. Song, X. Jiang, and L. Shang. Neural subgraph isomorphism counting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1959–1969, 2020.

[31] Z. Manjezi and R. A. Botha. From concept to practice: untangling the direct-control cycle. In *Proceedings of the 9th International Conference on Information Communication and Management*, pages 101–105, 2019.

[32] S. N. Matheu-García, J. L. Hernández-Ramos, A. F. Skarmeta, and G. Baldini. Risk-based automated assessment and testing for the cybersecurity certification and labelling of iot devices. *Computer Standards & Interfaces*, 62:64–83, 2019.

[33] S. Maynard, T. Tan, A. Ahmad, and T. Ruighaver. Towards a framework for strategic security context in information security governance. *Pacific Asia Journal of the Association for Information Systems*, 10(4):4, 2018.

[34] Z. Mounia and N. Bouchaib. A new comprehensive solution to handle information security governance in organizations. In *Proceedings of the 2nd International Conference on Networking, Information Systems & Security*, pages 1–5, 2019.

[35] M. Nicho. A process model for implementing information systems security governance. *Information & Computer Security*, 2018.

[36] T. Niesen, C. Houy, P. Fettke, and P. Loos. Towards an integrative big data analysis framework for data-driven risk management in industry 4.0. In *2016 49th Hawaii international conference on system sciences (HICSS)*, pages 5065–5074. IEEE, 2016.

[37] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu. A survey on systems security metrics. *ACM Computing Surveys (CSUR)*, 49(4):1–35, 2016.

[38] F. Sadique, S. Cheung, I. Vakilinia, S. Badsha, and S. Sengupta. Automated structured threat information expression (stix) document generation with privacy preservation. In *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 847–853. IEEE, 2018.

[39] A. Sedgewick. Framework for improving critical infrastructure cybersecurity, version 1.0, 2014-02-12 2014.

[40] S. Seifermann, R. Heinrich, D. Werle, and R. Reussner. Detecting violations of access control and information flow policies in data flow diagrams. *Journal of Systems and Software*, 184:111138, 2022.

[41] L. Sion, D. Van Landuyt, K. Yskout, and W. Joosen. Sparta: Security & privacy architecture through risk-driven threat assessment. In *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*, pages 89–92. IEEE, 2018.

[42] L. Sion, K. Yskout, D. Van Landuyt, and W. Joosen. Solution-aware data flow diagrams for security threat modeling. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 1425–1432, 2018.

[43] L. Sion, K. Yskout, D. Van Landuyt, A. van Den Berghe, and W. Joosen. Security threat modeling: Are data flow diagrams enough? In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, pages 254–257, 2020.

[44] M. R. Somkunwar and V. M. Vaze. Subgraph isomorphism algorithms for matching graphs: A survey. *IJETT*, 1(1), 2017.

[45] F. Ö. Sönmez. A conceptual model for a metric based framework for the monitoring of information security tasks' efficiency. *Procedia Computer Science*, 160:181–188, 2019.

[46] K. Tuma, R. Scandariato, and M. Balliu. Flaws in flows: Unveiling design flaws via information flow analysis. In *2019 IEEE International Conference on Software Architecture (ICSA)*, pages 191–200. IEEE, 2019.

[47] R. Von Solms and S. B. von Solms. Information security governance: a model based on the direct–control cycle. *Computers & Security*, 25(6):408–412, 2006.

[48] B. Waltl, G. Bonczek, and F. Matthes. Rule-based information extraction: Advantages, limitations, and perspectives. *Jusletter IT (02 2018)*, 2018.

[49] J. Williams. A data-based method for assessing and reducing human error to improve operational performance. In *Conference Record for 1988 IEEE Fourth Conference on Human Factors and Power Plants,*, pages 436–450. IEEE, 1988.

[50] K. Wuyts and W. Joosen. Linddun privacy threat modeling: a tutorial. *CW Reports*, 2015.

[51] M. Zaydi and B. Nassereddine. A new approach of information system security governance: A proposition of the continuous improvement process model of information system security risk management: 4d-iss. In *2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 112–118. IEEE, 2018.

[52] M. Zaydi and B. Nassereddine. Toward a new integrated approach of information security based on governance, risk and compliance. In *International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning*, pages 337–341. Springer, 2018.

Appendix A

Security Templates for the Italian National Framework for Cybersecurity and Data Protection

This Appendix contains all the Security Templates I designed for the Italian National Framework for Cybersecurity and Data Protection, in their graphical format. The Security Templates are organized in five different sections corresponding to the functions found in the framework, with each section containing the Security Templates corresponding to the subcategories contained inside that function. Within each section, the order in which Security Templates are organized reflects their order of appearance in the official documentation of the Framework.

All the png image files corresponding to the Security Templates, along with their corresponding computer friendly versions, are available on the GitHub repository[1] with all the material created for this thesis.

[1] `https://github.com/Fulvio-P/controlExtractor`

A.1 Identify Function

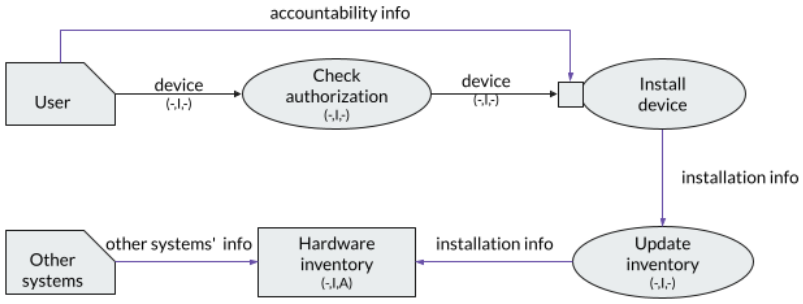


Figure A.1: ID.AM-1: Sono censiti i sistemi e gli apparati fisici in uso nell'organizzazione.

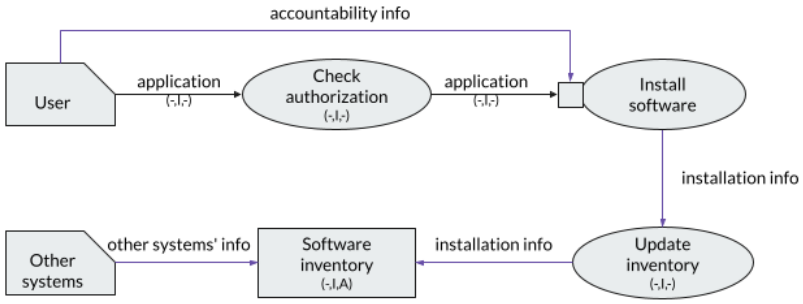


Figure A.2: ID.AM-2: Sono censite le piattaforme e le applicazioni in uso nell'organizzazione.

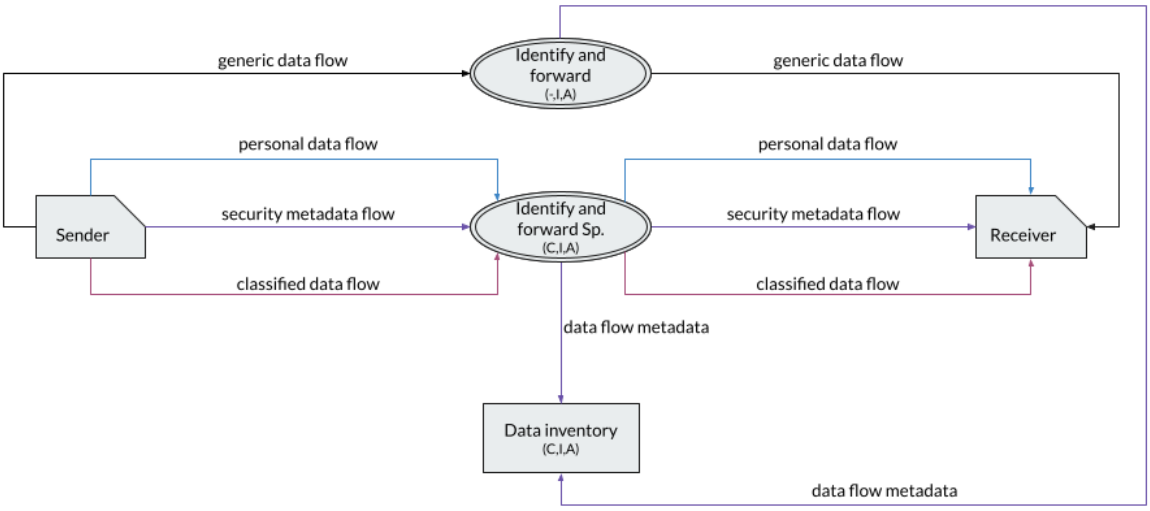


Figure A.3: ID.AM-3: I flussi di dati e comunicazioni inerenti l'organizzazione sono identificati.

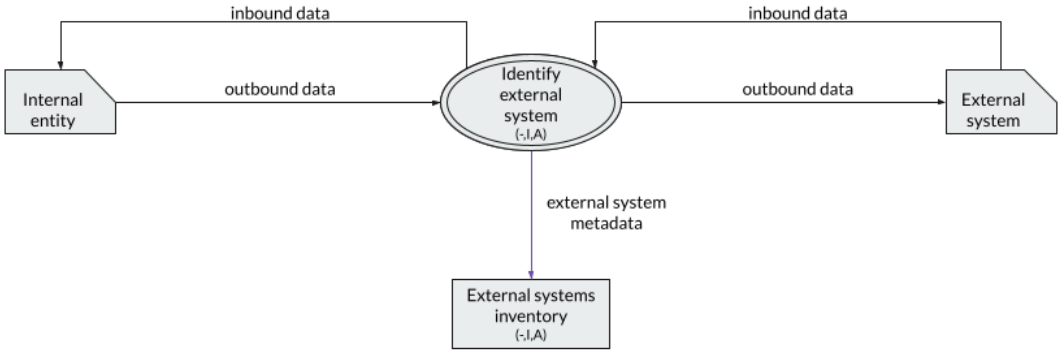


Figure A.4: ID.AM-4: I sistemi informativi esterni all'organizzazione sono catalogati.

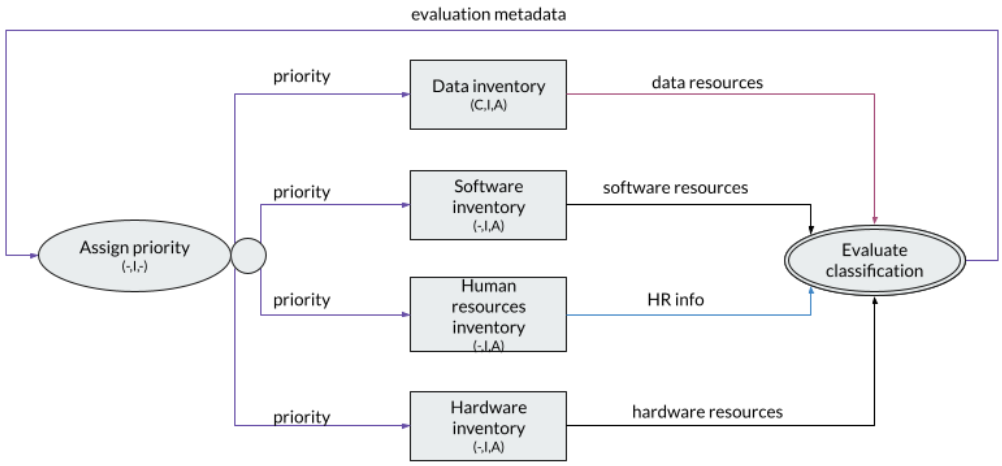


Figure A.5: ID.AM-5: Le risorse (es: hardware, dispositivi, dati, allocazione temporale, personale e software) sono prioritizzate in base alla loro classificazione (e.g. confidenzialità, integrità, disponibilità), criticità e valore per il business dell'organizzazione.

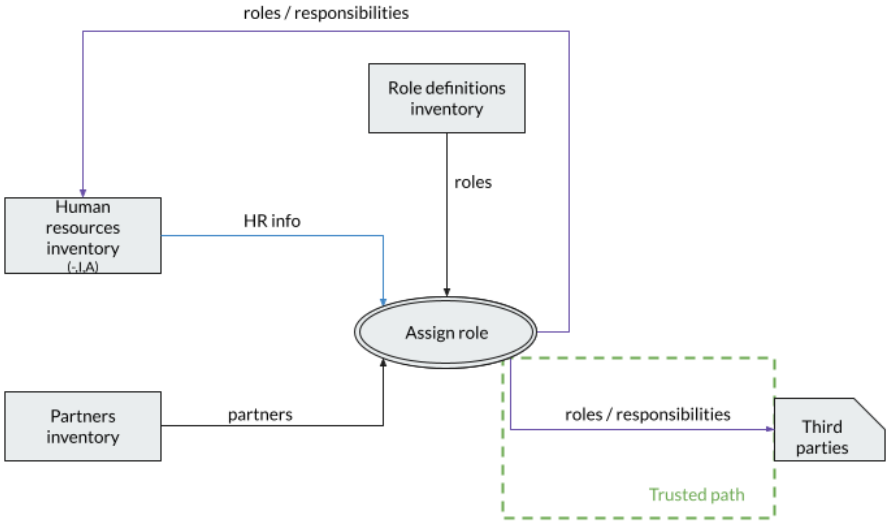


Figure A.6: ID.AM-6: Sono definiti e resi noti ruoli e responsabilità inerenti la cybersecurity per tutto il personale e per eventuali terze parti rilevanti (es. fornitori, clienti, partner)

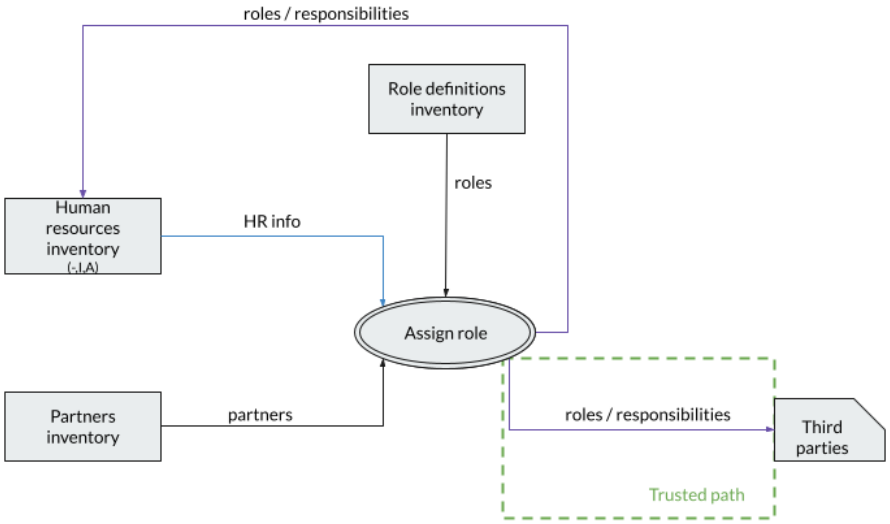


Figure A.7: DP-ID.AM-7: Sono definiti e resi noti ruoli e responsabilità inerenti al trattamento e la protezione dei dati personali per tutto il personale e per eventuali terze parti rilevanti (es. fornitori, clienti, partner)

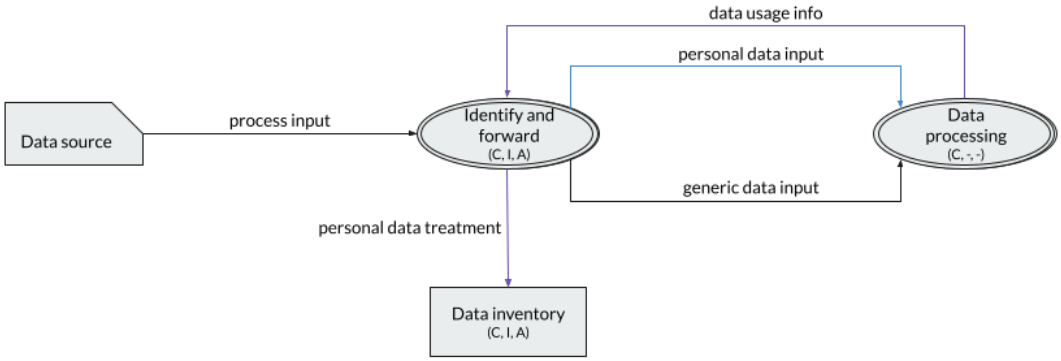


Figure A.8: DP-ID.AM-8: I trattamenti di dati personali sono identificati e catalogati

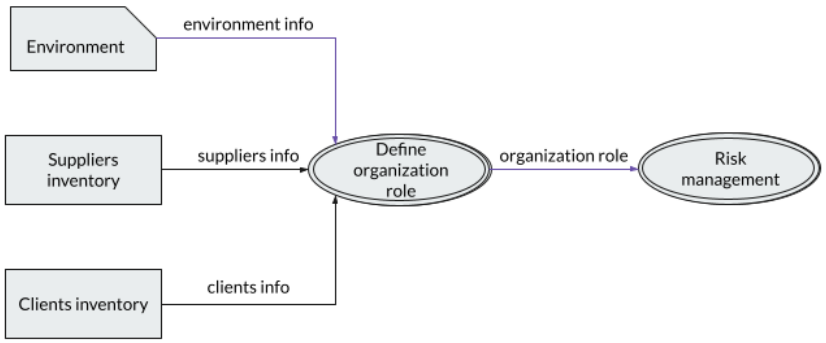


Figure A.9: ID.BE-1: Il ruolo dell'organizzazione all'interno della filiera produttiva è identificato e reso noto

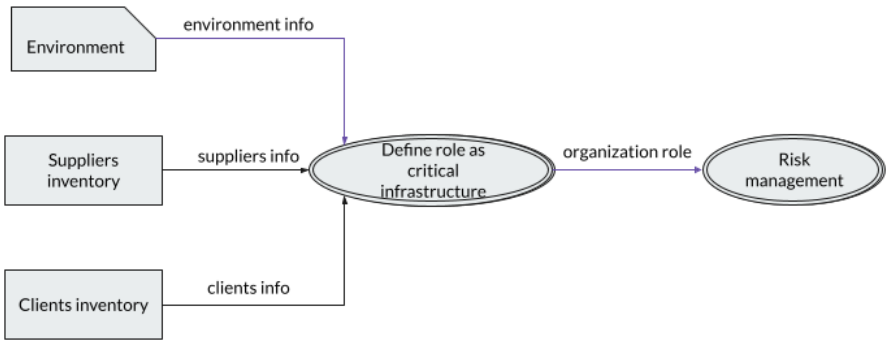


Figure A.10: ID.BE-2: Il ruolo dell'organizzazione come infrastruttura critica e nel settore industriale di riferimento è identificato e reso noto

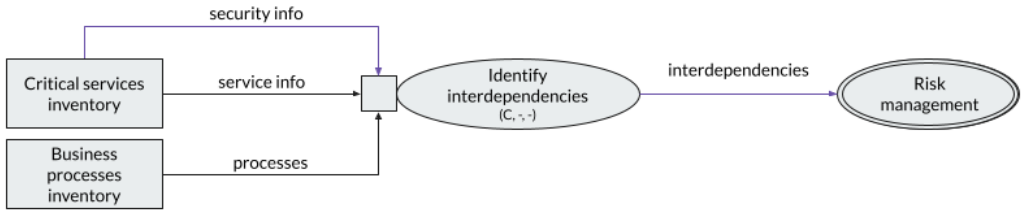


Figure A.11: ID.BE-4: Sono identificate e rese note interdipendenze e funzioni fondamentali per la fornitura di servizi critici

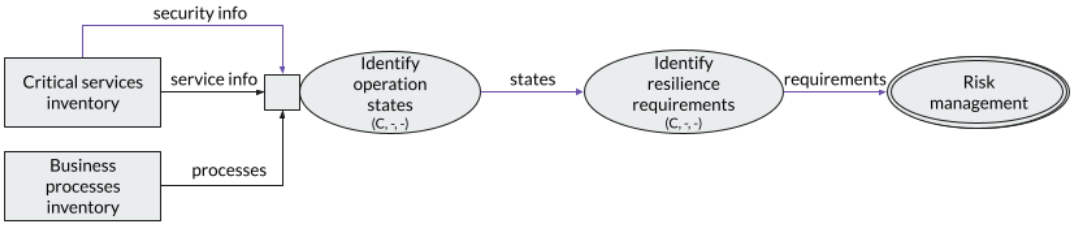


Figure A.12: ID.BE-5: Sono identificati e resi noti i requisiti di resilienza a supporto della fornitura di servizi critici per tutti gli stati di esercizio (es. sotto stress/attacco, in fase di recovery, normale esercizio)

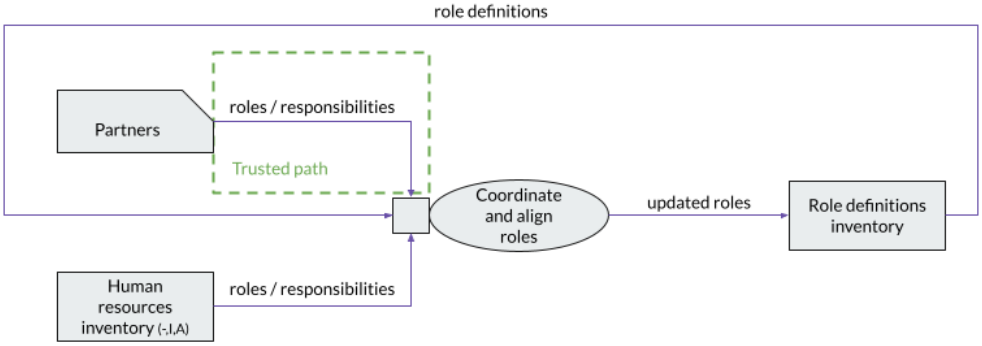


Figure A.13: ID.GV-2: Ruoli e responsabilità inerenti la cybersecurity sono coordinati ed allineati con i ruoli interni ed i partner esterni

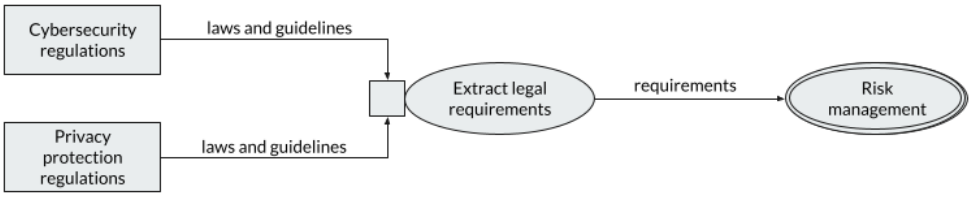


Figure A.14: ID.GV-3: I requisiti legali in materia di cybersecurity, con l'inclusione degli obblighi riguardanti la privacy e le libertà civili, sono compresi e gestiti

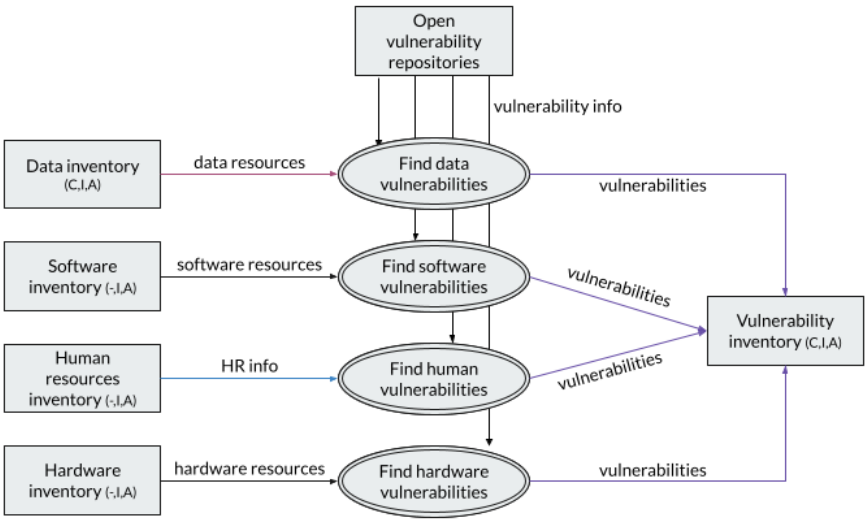


Figure A.15: ID.RA-1: Le vulnerabilità delle risorse (es. sistemi, locali, dispositivi) dell'organizzazione sono identificate e documentate

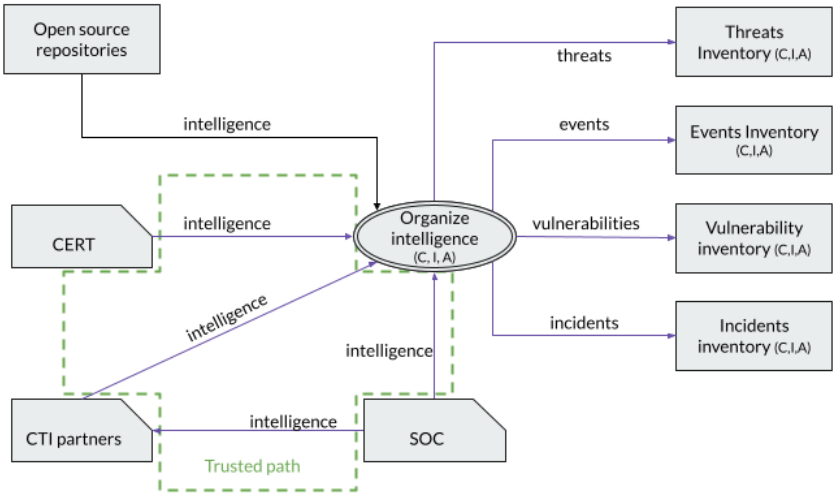


Figure A.16: ID.RA-2: L'organizzazione riceve informazioni su minacce, vulnerabilità ed altri dati configurabili come Cyber Threat Intelligence da fonti esterne (e.g. CERT, fonti aperte, forum di information sharing)

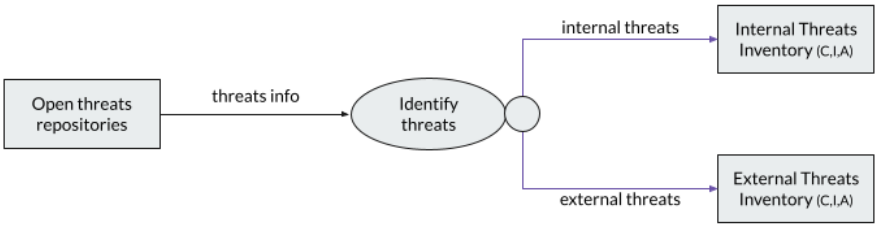


Figure A.17: ID.RA-3: Le minacce, sia interne che esterne, sono identificate e documentate

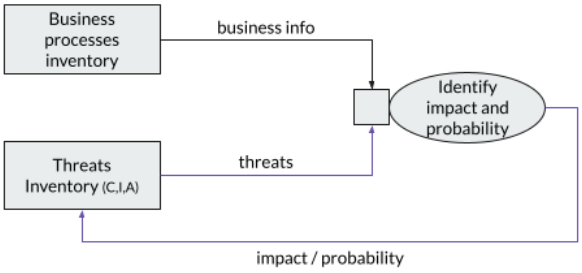


Figure A.18: ID.RA-4: Sono identificati i potenziali impatti sul business e le relative probabilità di accadimento

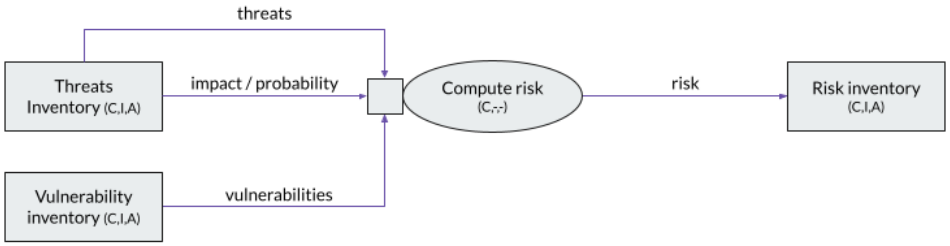


Figure A.19: ID.RA-5: Le minacce, le vulnerabilità, le relative probabilità di accadimento e conseguenti impatti sono utilizzati per determinare il rischio

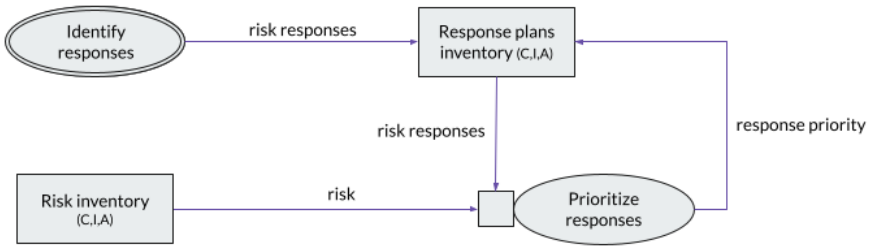


Figure A.20: ID.RA-6: Sono identificate e prioritizzate le risposte al rischio

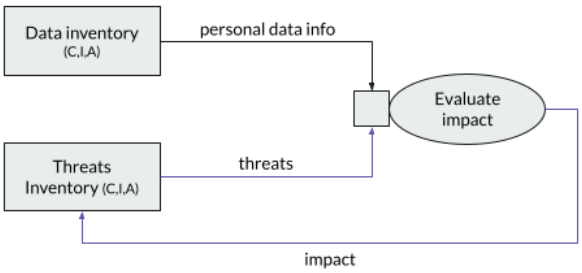


Figure A.21: DP-ID.RA-7: Viene effettuata una valutazione di impatto sulla protezione dei dati personali

Figure A.22: ID.RM-1: I processi di risk management sono stabiliti, gestiti e concordati tra i responsabili dell'organizzazione (c.d. stakeholder)

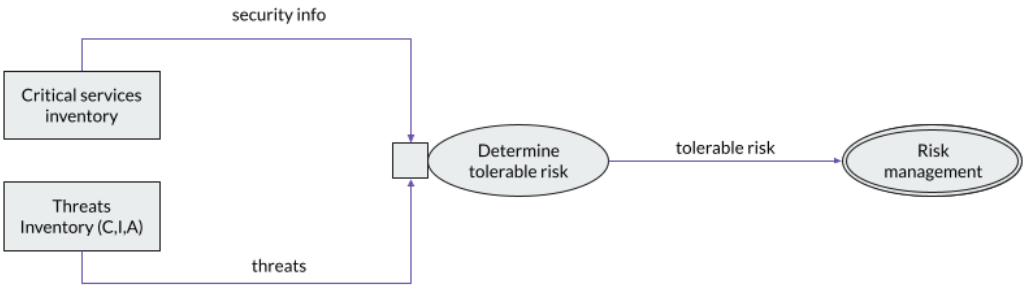


Figure A.23: ID.RM-3: Il rischio tollerato è determinato tenendo conto del ruolo dell'organizzazione come infrastruttura critica e dei rischi specifici presenti nel settore industriale di appartenenza

Figure A.24: ID.SC-1: I processi di gestione del rischio inerenti la catena di approvvigionamento cyber sono identificati, ben definiti, validati, gestiti e approvati da attori interni all'organizzazione

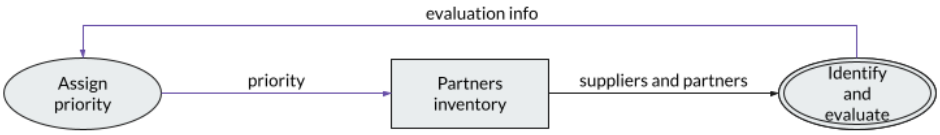


Figure A.25: ID.SC-2: I fornitori e i partner terzi di sistemi informatici, componenti e servizi sono identificati, prioritizzati e valutati utilizzando un processo di valutazione del rischio inerente la catena di approvvigionamento cyber

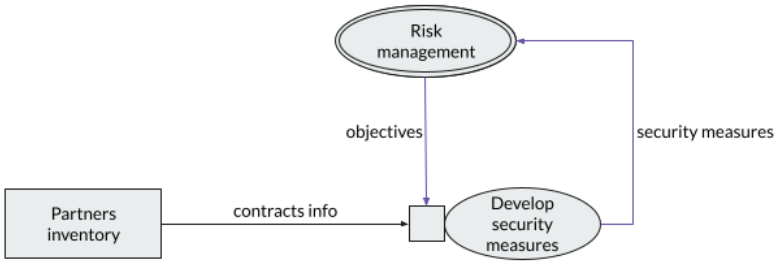


Figure A.26: ID.SC-3: I contratti con i fornitori e i partner terzi sono utilizzati per realizzare appropriate misure progettate per rispettare gli obiettivi del programma di cybersecurity dell'organizzazione e del Piano di Gestione del Rischio della catena di approvvigionamento cyber

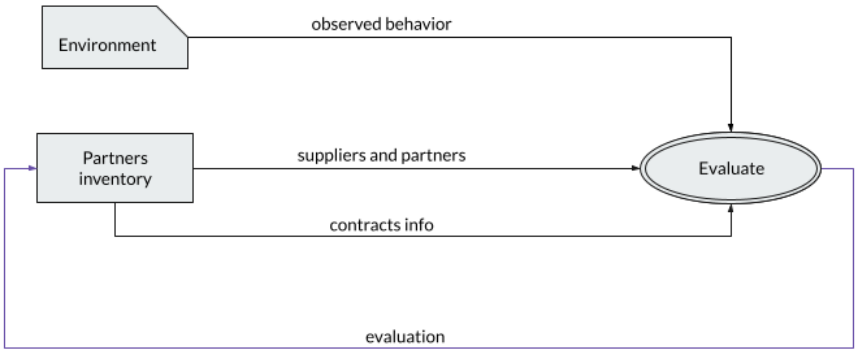


Figure A.27: ID.SC-4: Fornitori e partner terzi sono regolarmente valutati utilizzando audit, verifiche, o altre forme di valutazione per confermare il rispetto degli obblighi contrattuali

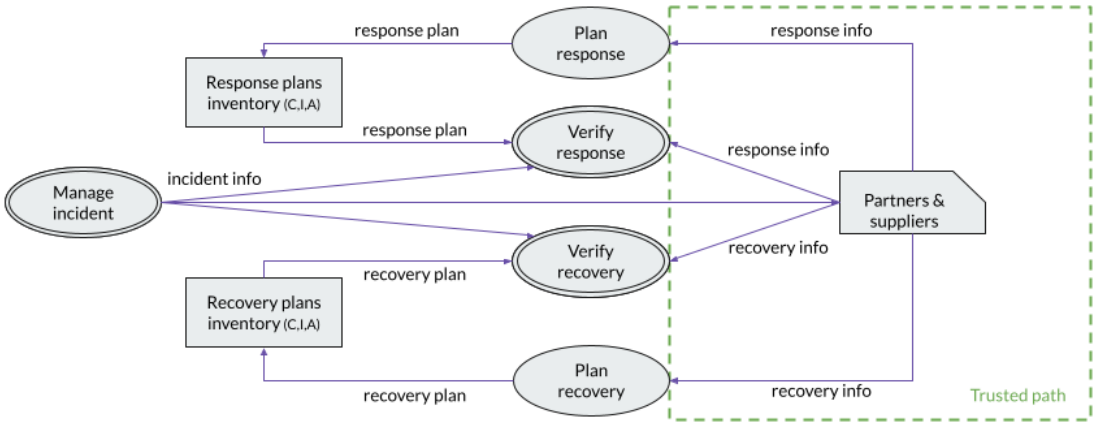


Figure A.28: ID.SC-5: La pianificazione e la verifica della risposta e del ripristino sono condotti con i fornitori e i partner terzi

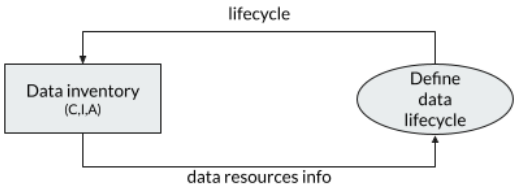


Figure A.29: DP-ID.DM-1: Il ciclo di vita dei dati è definito e documentato

Figure A.30: DP-ID.DM-2: Sono definiti, implementati e documentati i processi riguardanti l'informazione dell'interessato in merito al trattamento dei dati

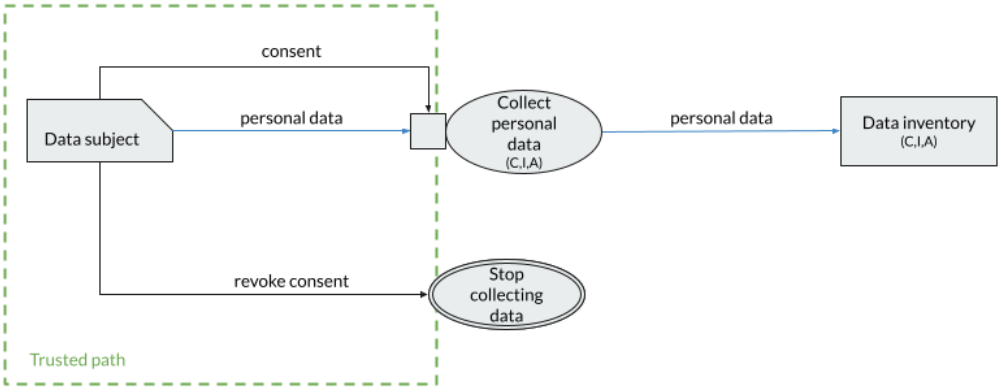


Figure A.31: DP-ID.DM-3: Sono definiti, implementati e documentati i processi di raccolta e revoca del consenso dell'interessato al trattamento di dati

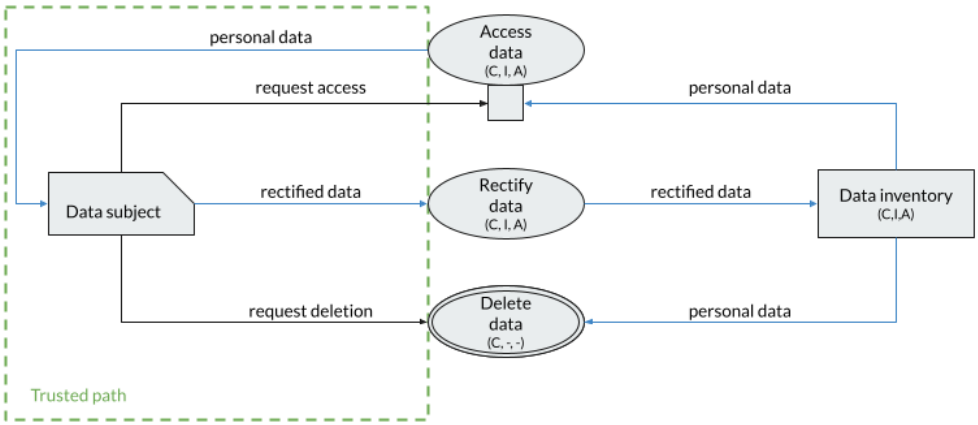


Figure A.32: DP-ID.DM-4: Sono definiti, implementati e documentati i processi per l'esercizio dei diritti (accesso, rettifica, cancellazione, ecc.) dell'interessato

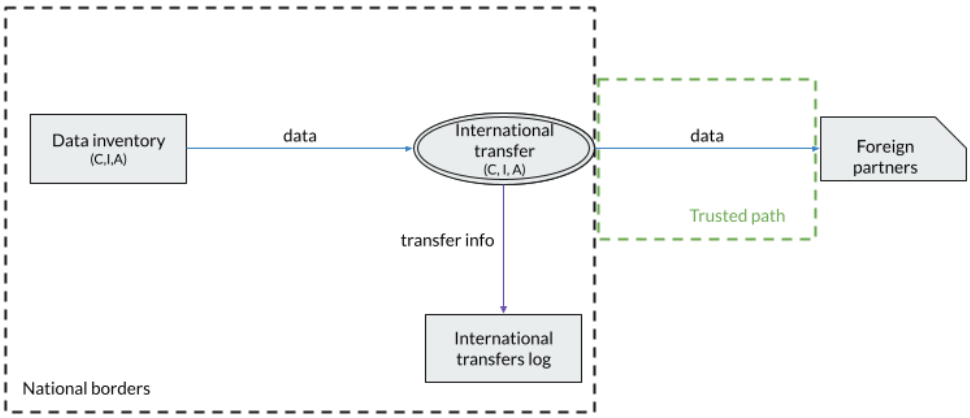


Figure A.33: DP-ID.DM-5: Sono definiti, implementati e documentati i processi di trasferimento dei dati in ambito internazionale

A.2 Protect Function

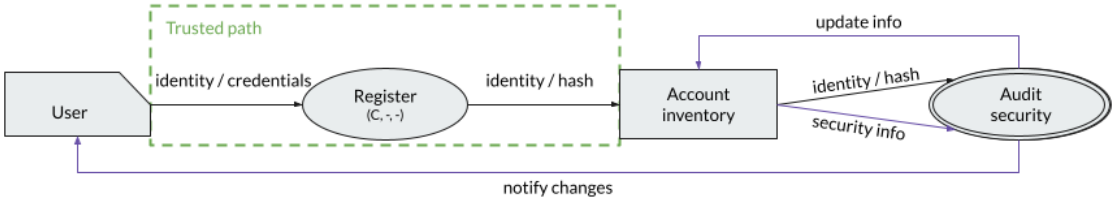


Figure A.34: PR.AC-1: Le identità digitali e le credenziali di accesso per gli utenti, i dispositivi e i processi autorizzati sono amministrate, verificate, revocate e sottoposte a audit sicurezza

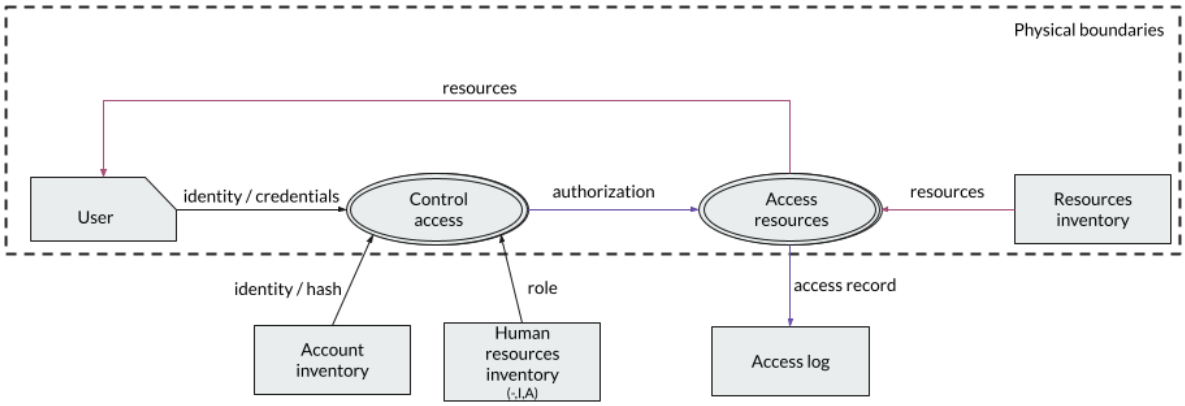


Figure A.35: PR.AC-2: L'accesso fisico alle risorse è protetto e amministrato

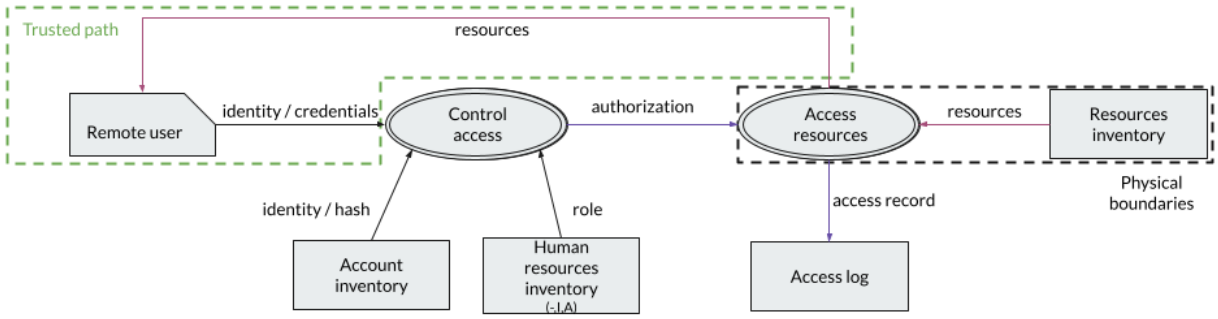


Figure A.36: PR.AC-3: L'accesso remoto alle risorse è amministrato

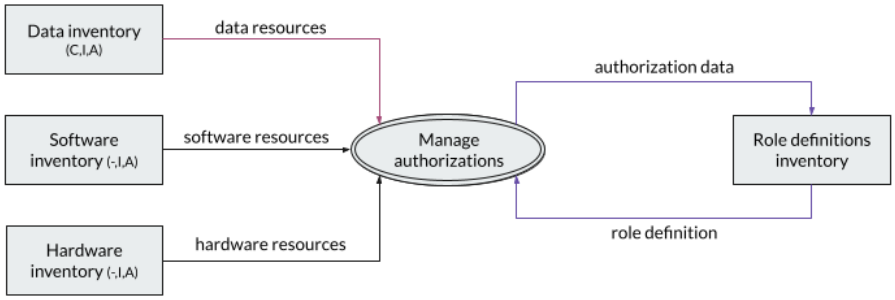


Figure A.37: PR.AC-4: I diritti di accesso alle risorse e le relative autorizzazioni sono amministrati secondo il principio del privilegio minimo e della separazione delle funzioni

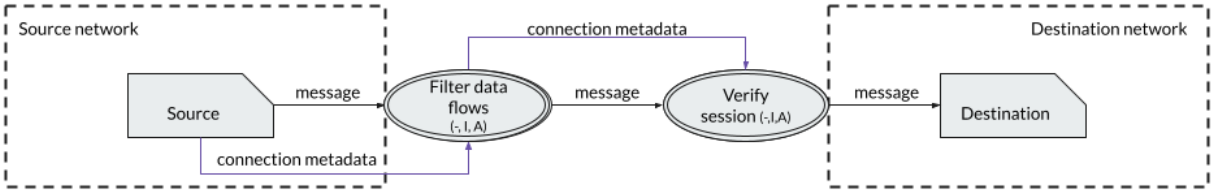


Figure A.38: PR.AC-5: L'integrità di rete è protetta (es. segregazione di rete, segmentazione di rete)

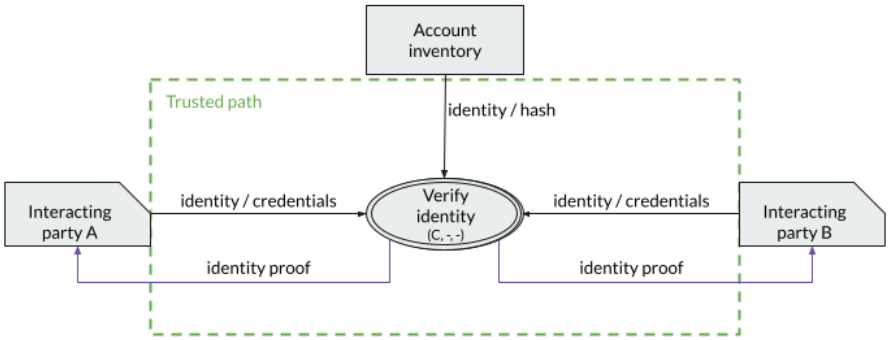


Figure A.39: PR.AC-6: Le identità sono comprovate, associate a credenziali e verificate durante le interazioni

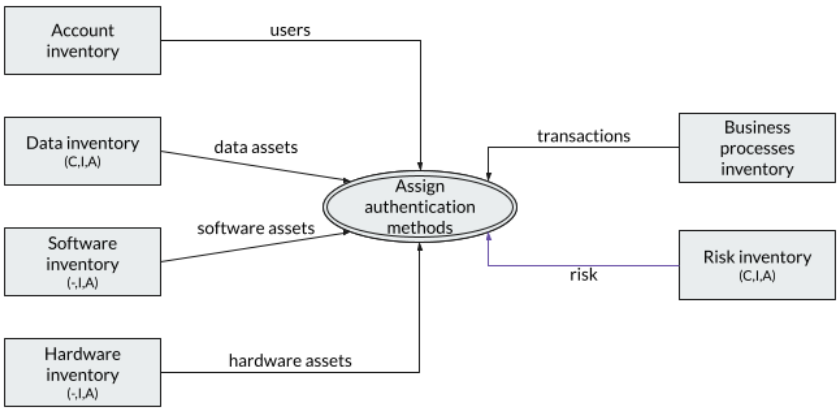


Figure A.40: PR.AC-7: Le modalità di autenticazione (es. autenticazione a fattore singolo o multiplo) per gli utenti, i dispositivi e altri asset sono commisurate al rischio della transazione (es. rischi legati alla sicurezza e privacy degli individui e altri rischi dell'organizzazione)

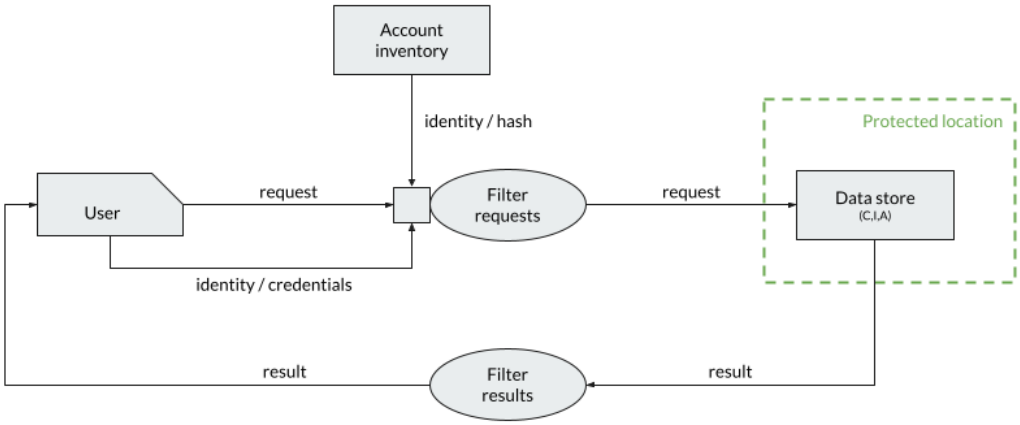


Figure A.41: PR.DS-1: I dati memorizzati sono protetti

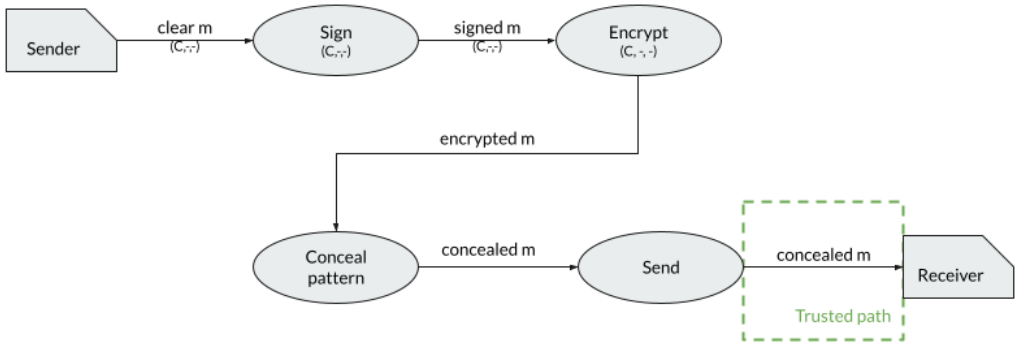


Figure A.42: PR.DS-2: I dati sono protetti durante la trasmissione

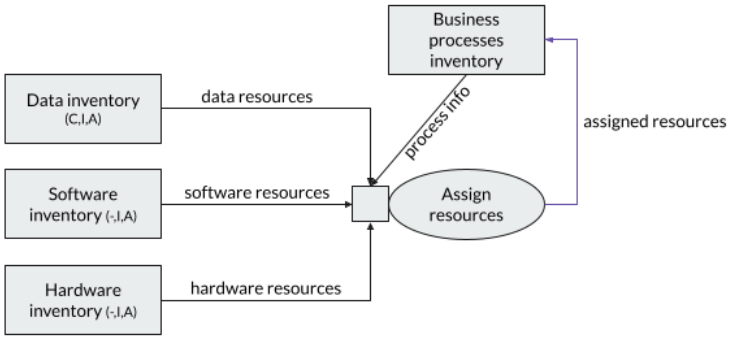


Figure A.43: PR.DS-4: I sistemi hanno adeguate risorse a disposizione per poter garantire la disponibilità

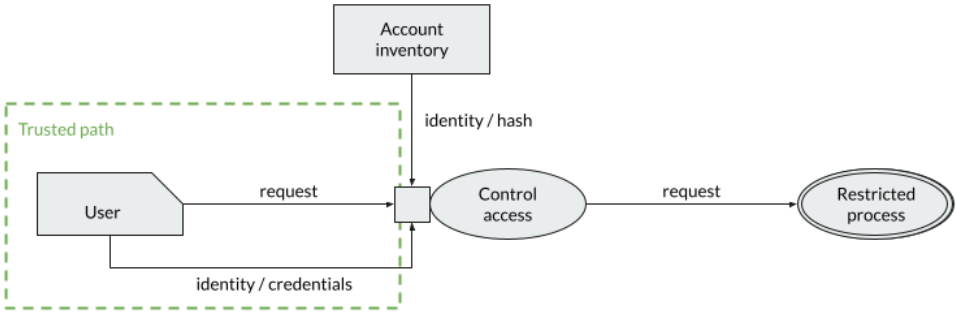


Figure A.44: PR.DS-5: Sono implementate tecniche di protezione (es. controllo di accesso) contro la sottrazione dei dati (data leak).

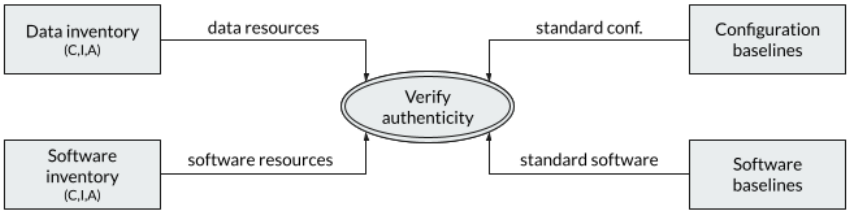


Figure A.45: PR.DS-6: Sono impiegati meccanismi di controllo dell'integrità dei dati per verificare l'autenticità di software, firmware e delle informazioni

Figure A.46: PR.DS-7: Gli ambienti di sviluppo e test sono separati dall'ambiente di produzione

Figure A.47: PR.DS-8: Sono impiegati meccanismi di controllo dell'integrità per verificare l'integrità del hardware

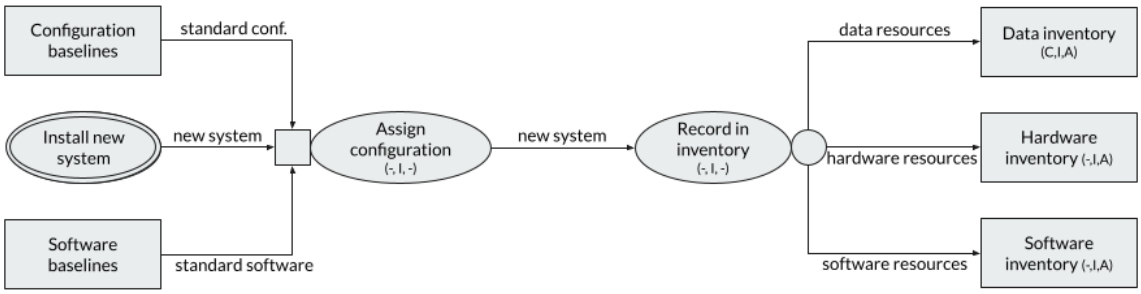


Figure A.48: PR.IP-1: Sono definite e gestite delle pratiche di riferimento (c.d. baseline) per la configurazione dei sistemi IT e di controllo industriale che incorporano principi di sicurezza (es. principio di minima funzionalità)

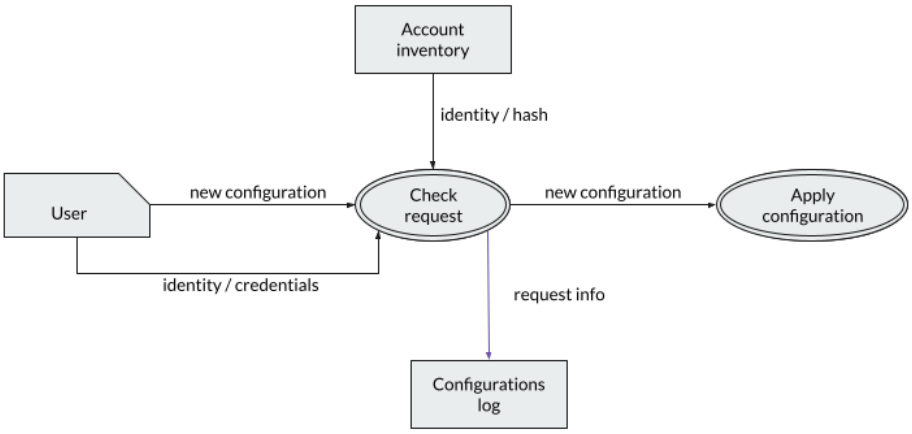


Figure A.49: PR.IP-3: Sono attivi processi di controllo della modifica delle configurazioni

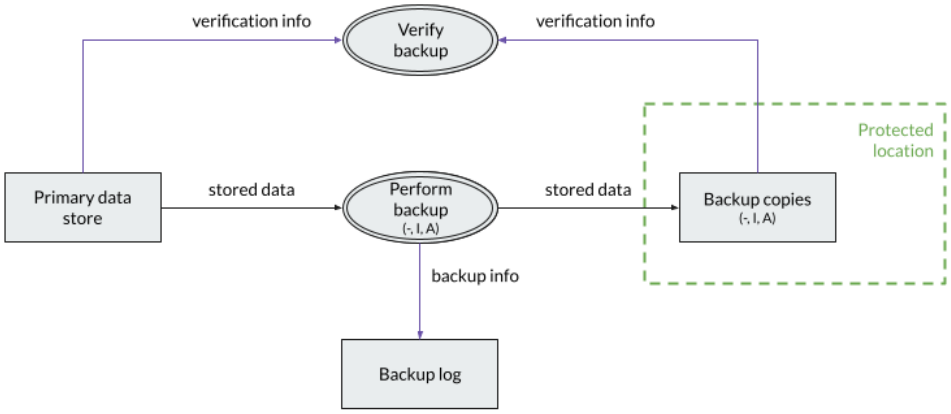


Figure A.50: PR.IP-4: I backup delle informazioni sono eseguiti, amministrati e verificati

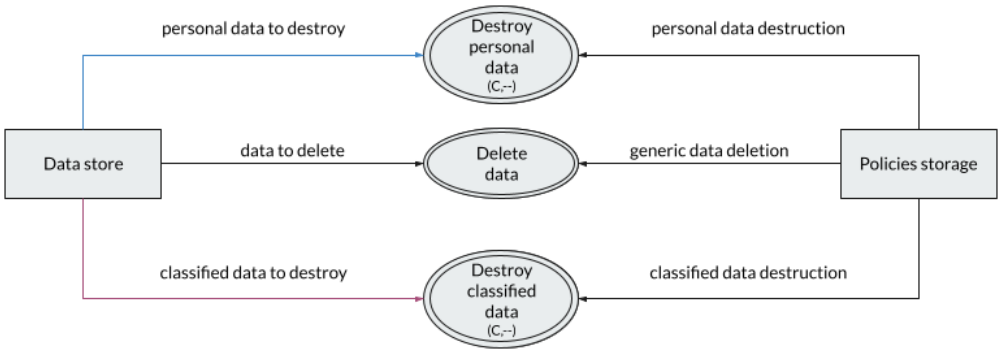


Figure A.51: PR.IP-6: I dati sono distrutti in conformità con le policy

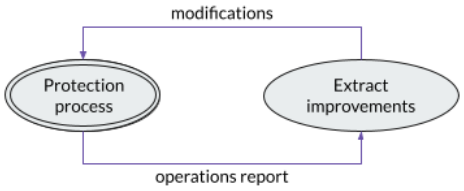


Figure A.52: PR.IP-7: I processi di protezione sono sottoposti a miglioramenti

Figure A.53: PR.IP-8: L'efficacia delle tecnologie di protezione viene condivisa

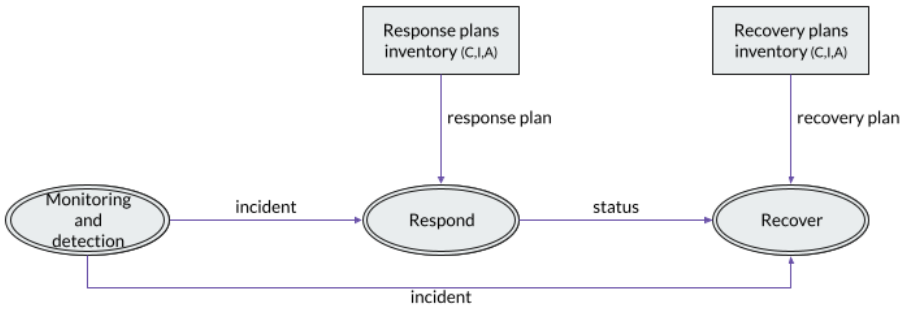


Figure A.54: PR.IP-9: Sono attivi ed amministrati piani di risposta (Incident Response e Business Continuity) e recupero (Incident Recovery e Disaster Recovery) in caso di incidente/disastro

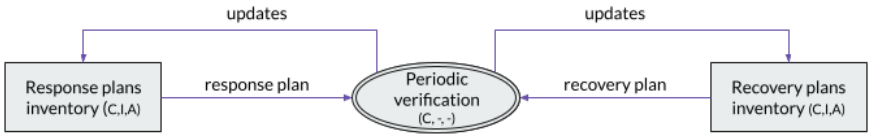


Figure A.55: PR.IP-10: I piani di risposta e recupero a seguito di incidenti/disastri sono verificati nel tempo

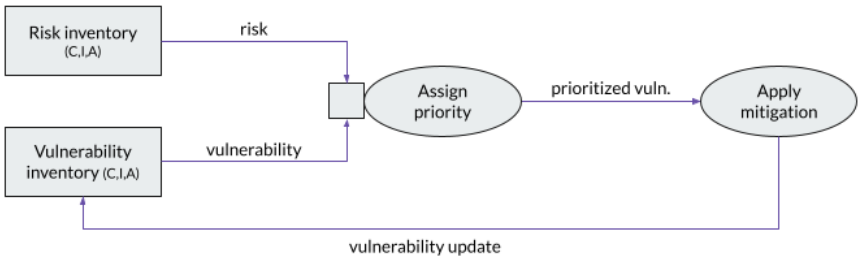


Figure A.56: PR.IP-12: Viene sviluppato e implementato un piano di gestione delle vulnerabilità

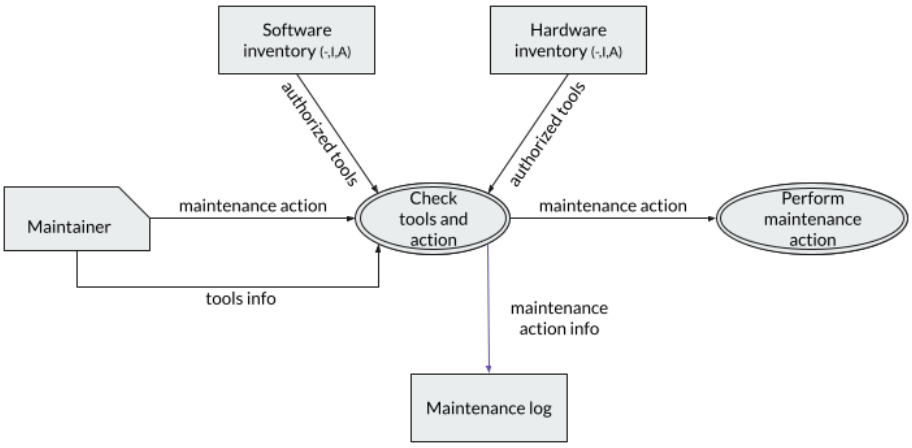


Figure A.57: PR.MA-1: La manutenzione e la riparazione delle risorse e dei sistemi è eseguita e registrata con strumenti controllati ed autorizzati

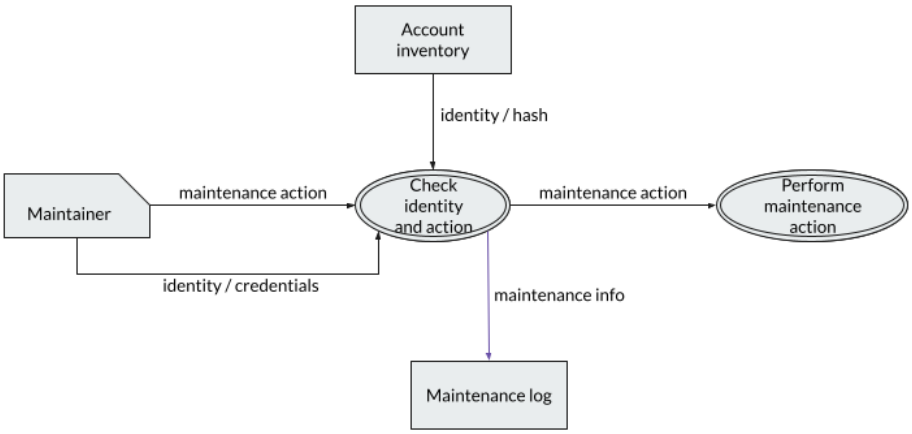


Figure A.58: PR.MA-2: La manutenzione remota delle risorse e dei sistemi è approvata, documentata e svolta in modo da evitare accessi non autorizzati

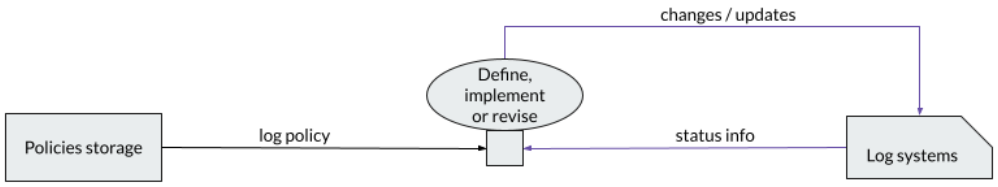


Figure A.59: PR.PT-1: Esiste ed è attuata una policy per definire, implementare e revisionare i log dei sistemi

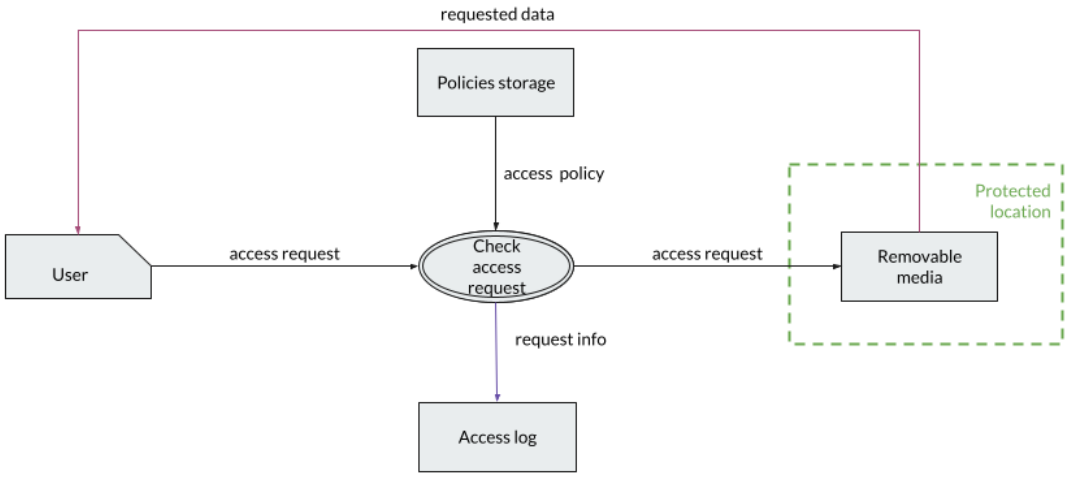


Figure A.60: PR.PT-2: I supporti di memorizzazione rimovibili sono protetti ed il loro uso è ristretto in accordo alle policy

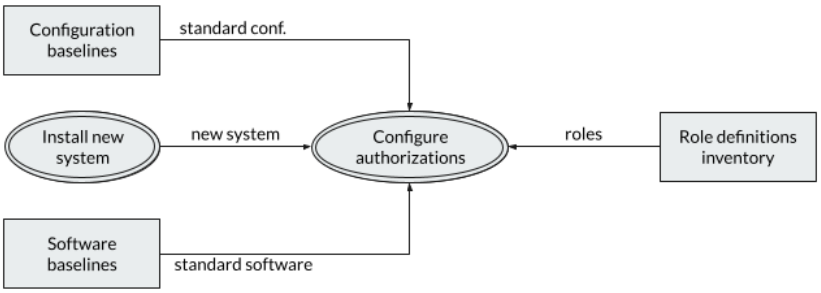


Figure A.61: PR.PT-3: Viene adottato il principio di minima funzionalità configurando i sistemi in modo che forniscano solo le funzionalità necessarie

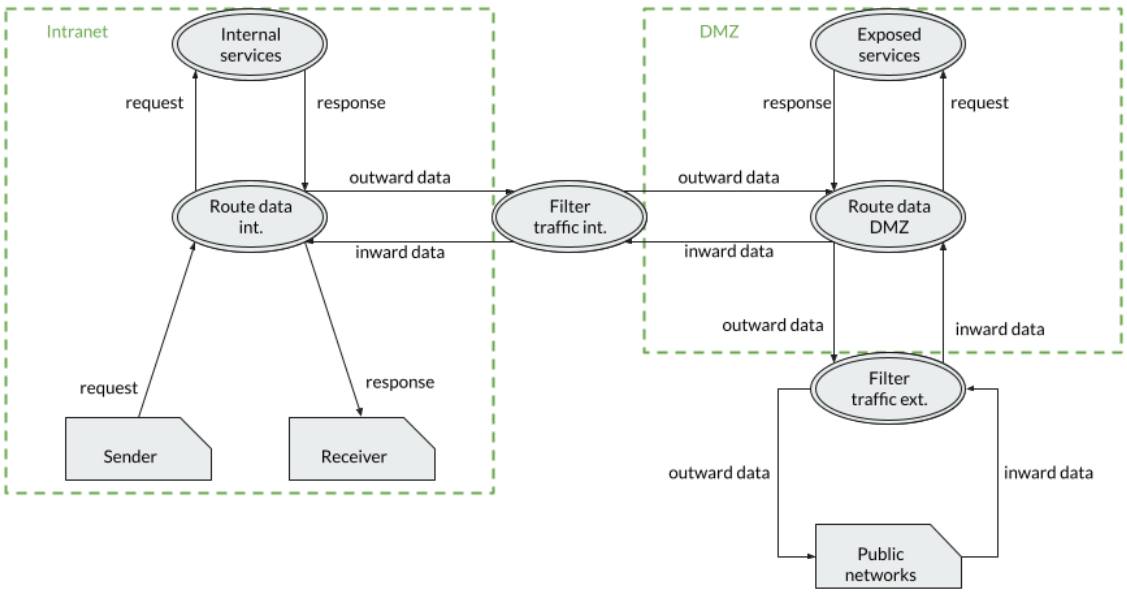


Figure A.62: PR.PT-4: Le reti di comunicazione e controllo sono protette

A.3 Detect Function

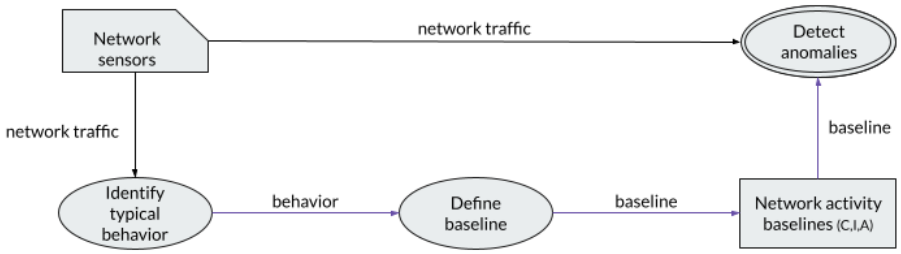


Figure A.63: DE.AE-1: Sono definite, rese note e gestite delle pratiche di riferimento (c.d. baseline) inerenti l'utilizzo della rete ed i flussi informativi attesi per utenti e sistemi

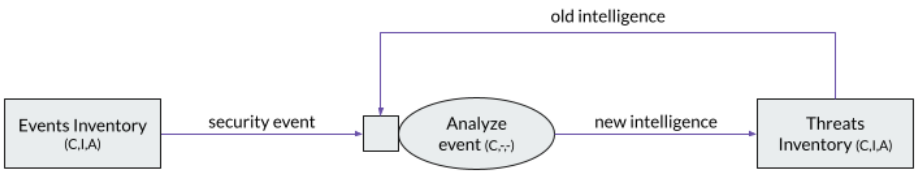


Figure A.64: DE.AE-2: Gli eventi rilevati vengono analizzati per comprendere gli obiettivi e le metodologie dell'attacco

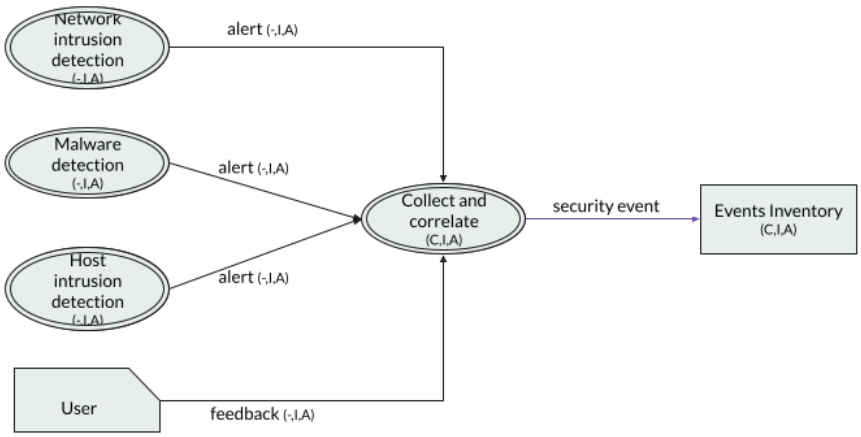


Figure A.65: DE.AE-3: Le informazioni relative agli eventi sono raccolte e correlate da sensori e sorgenti multiple

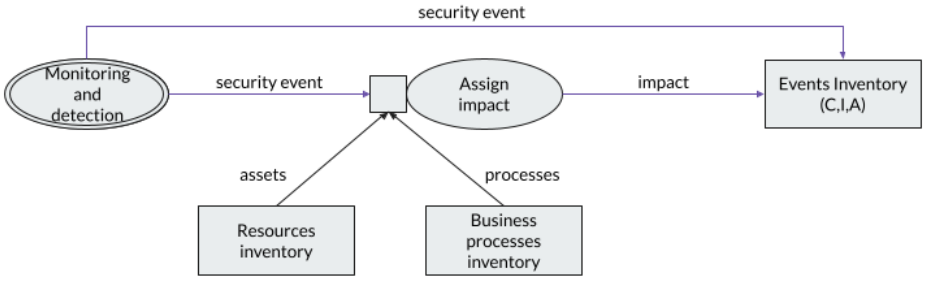


Figure A.66: DE.AE-4: Viene determinato l'impatto di un evento

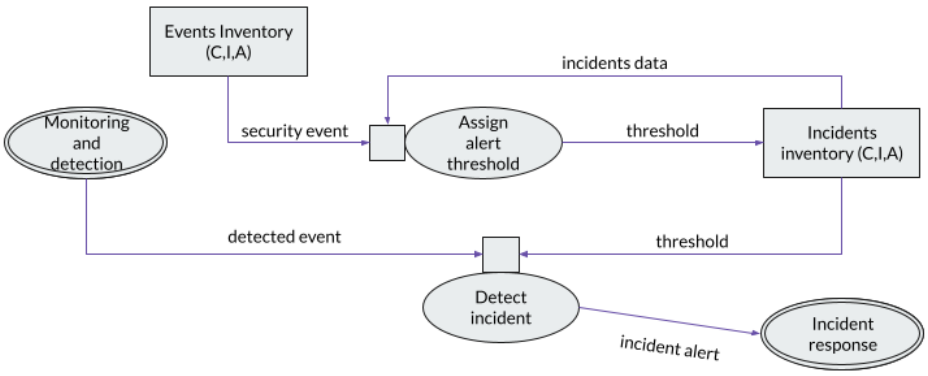


Figure A.67: DE.AE-5: Vengono definite delle soglie di allerta per gli incidenti

Figure A.68: DE.CM-3: Viene svolto il monitoraggio del personale per rilevare potenziali eventi di cybersecurity

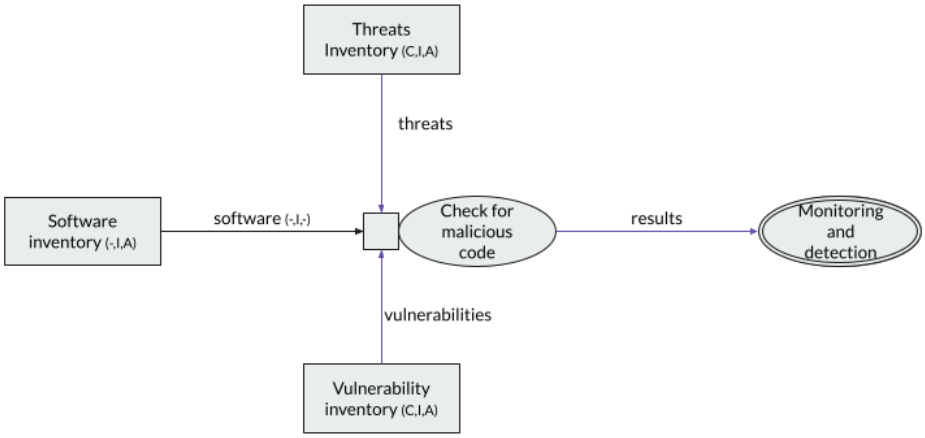


Figure A.69: DE.CM-4: Il codice malevolo viene rilevato

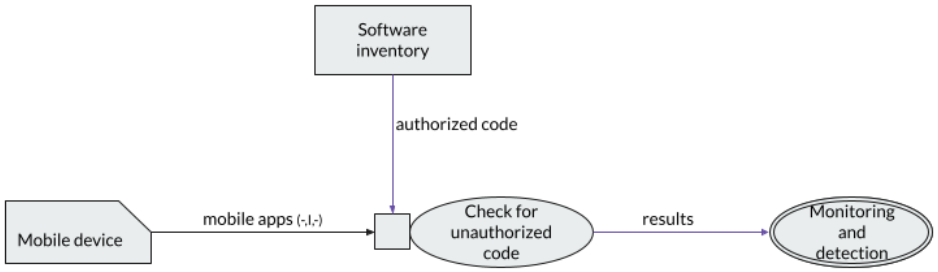


Figure A.70: DE.CM-5: Il codice non autorizzato su dispositivi mobili viene rilevato

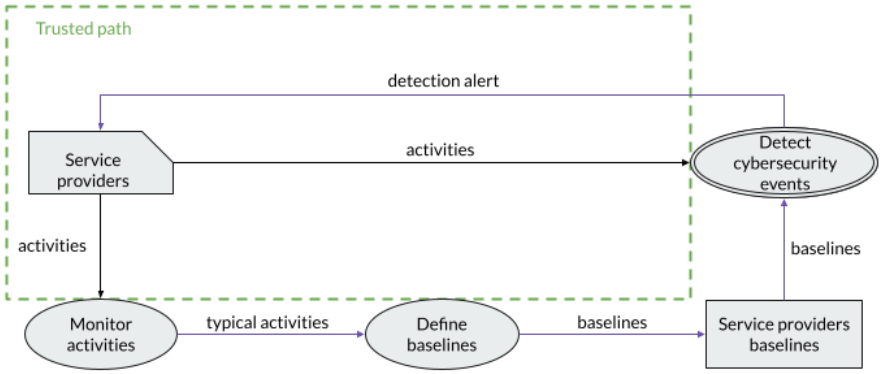


Figure A.71: DE.CM-6: Viene svolto il monitoraggio delle attività dei service provider esterni per rilevare potenziali eventi di cybersecurity

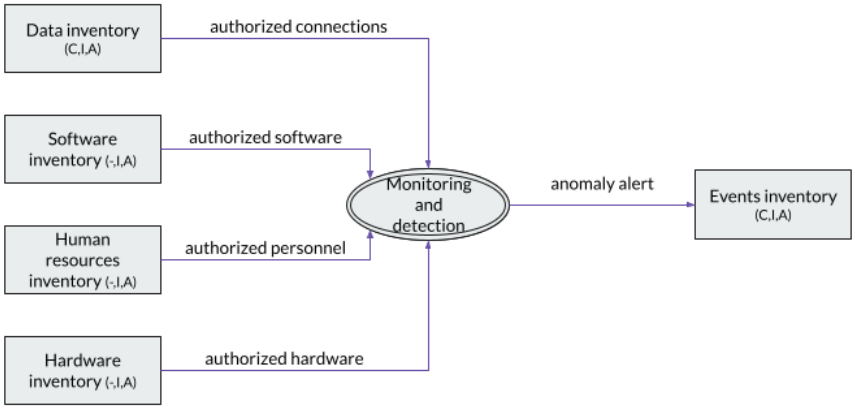


Figure A.72: DE.CM-7: Viene svolto il monitoraggio per rilevare personale, connessioni, dispositivi o software non autorizzati

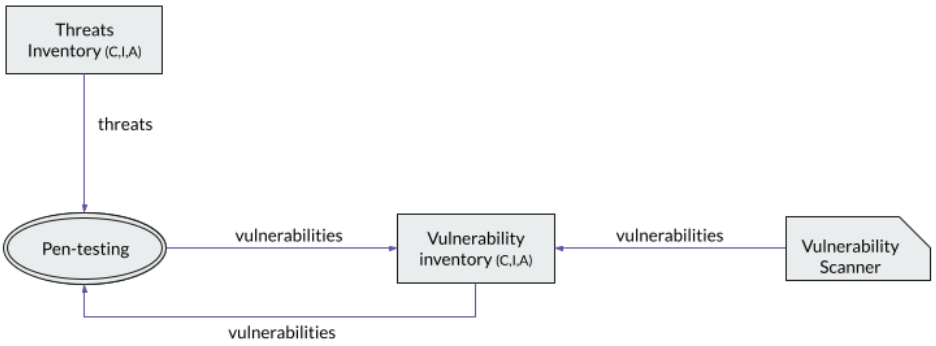


Figure A.73: DE.CM-8: Vengono svolte scansioni per l'identificazione di vulnerabilità

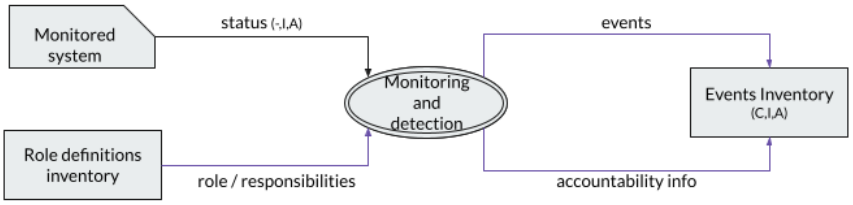


Figure A.74: DE.DP-1: Ruoli e responsabilità per i processi di monitoraggio sono ben definiti al fine di garantire l'accountability

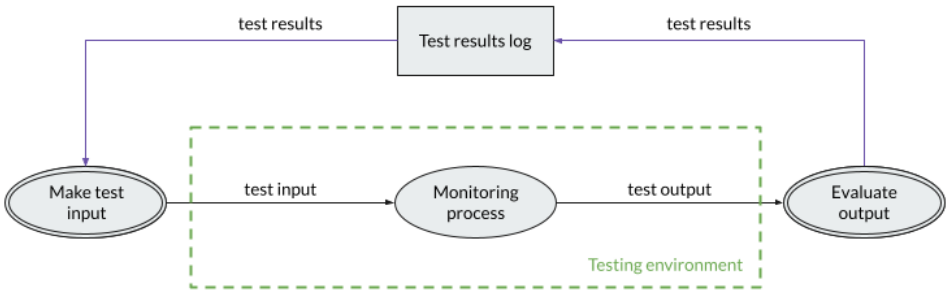


Figure A.75: DE.DP-3: I processi di monitoraggio vengono testati

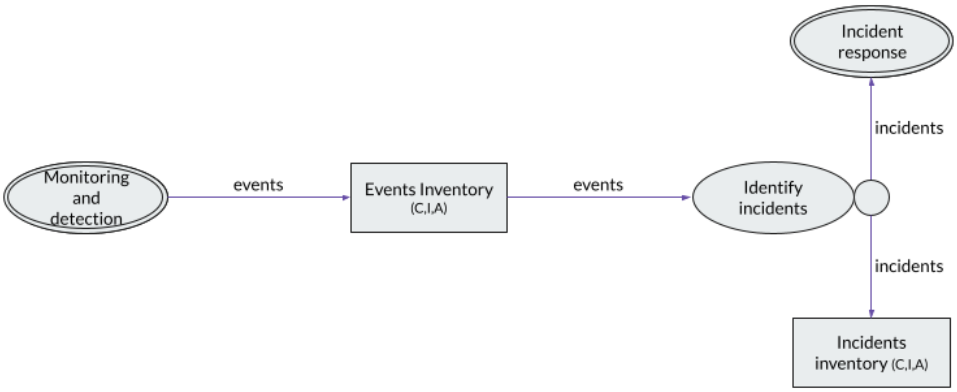


Figure A.76: DE.DP-4: L'informazione relativa agli eventi rilevati viene comunicata

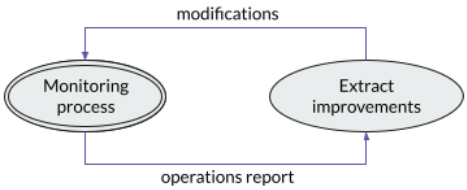


Figure A.77: DE.DP-5: I processi di monitoraggio sono oggetto di periodici miglioramenti e perfezionamenti

A.4 Respond Function

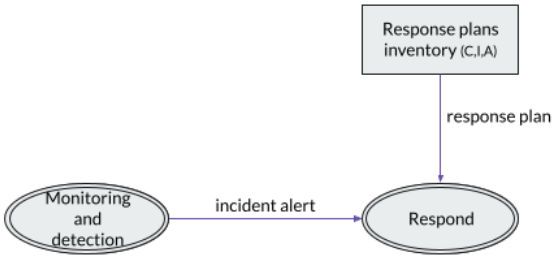


Figure A.78: RS.RP-1: Esiste un piano di risposta (response plan) e questo viene eseguito durante o dopo un incidente

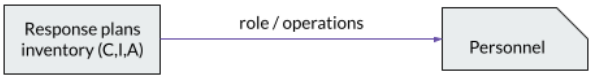


Figure A.79: RS.CO-1: Il personale conosce il proprio ruolo e le operazioni che deve svolgere in caso sia necessaria una risposta ad un incidente

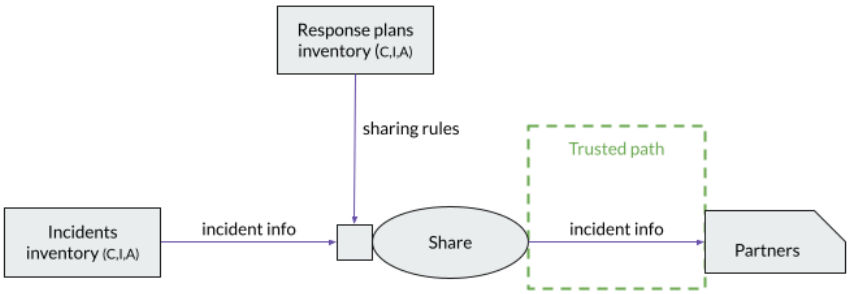


Figure A.80: RS.CO-3: Le informazioni sono condivise in maniera coerente con il piano di risposta

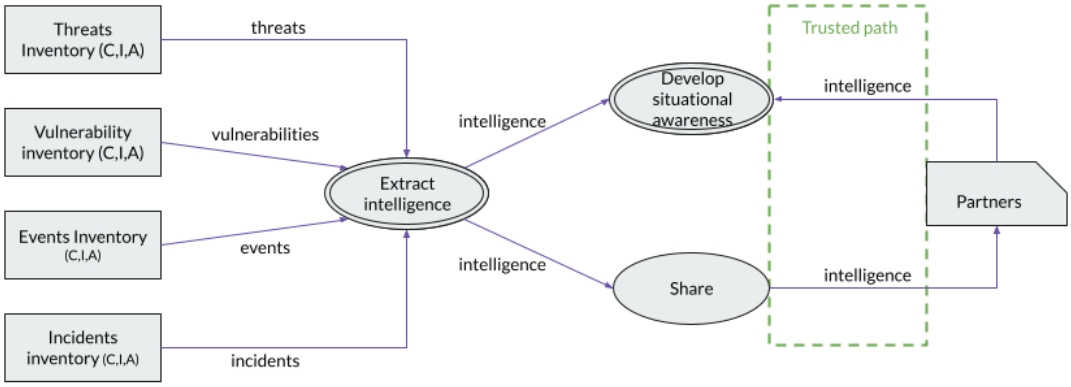


Figure A.81: RS.CO-5: È attuata una condivisione spontanea delle informazioni con le parti interessate esterne all'organizzazione (information sharing) per ottenere una maggior consapevolezza della situazione (c.d. situational awareness)

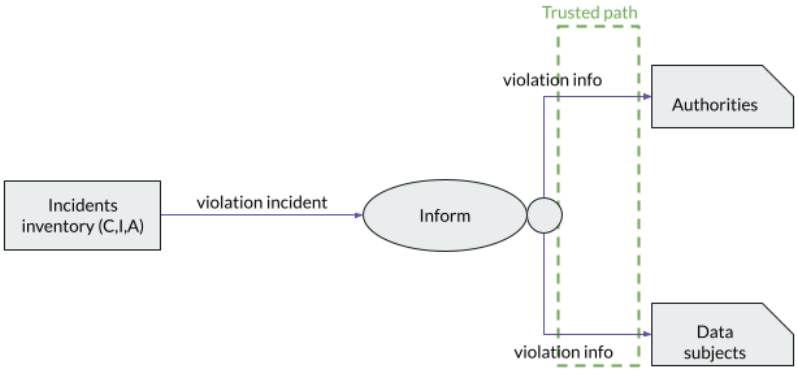


Figure A.82: DP-RS.CO-6: Gli incidenti che si configurano come violazioni di dati personali sono documentati ed eventualmente vengono informati le autorità di riferimento e gli interessati

Figure A.83: RS.AN-1: Le notifiche provenienti dai sistemi di monitoraggio vengono sempre visionate e analizzate

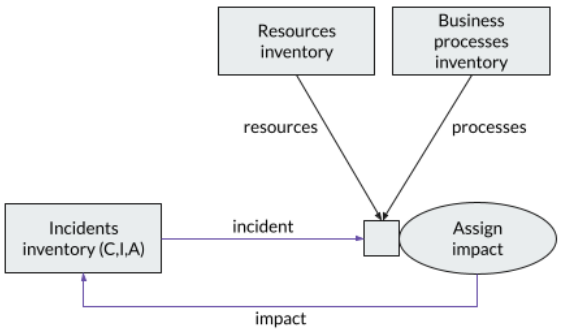


Figure A.84: RS.AN-2: Viene compreso l'impatto di ogni incidente

Figure A.85: RS.AN-3: A seguito di un incidente viene svolta un'analisi forense

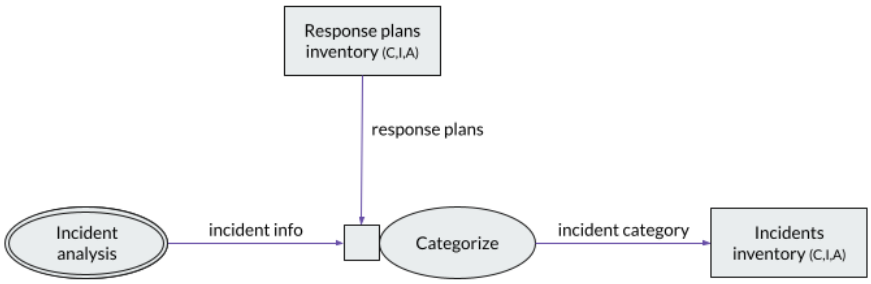


Figure A.86: RS.AN-4: Gli incidenti sono categorizzate in maniera coerente con i piani di risposta

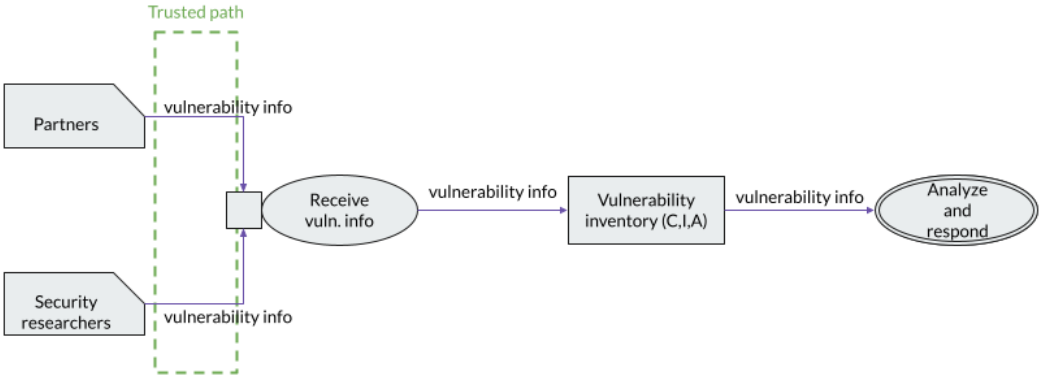


Figure A.87: RS.AN-5: Sono definiti processi per ricevere, analizzare e rispondere a informazioni inerenti vulnerabilità rese note da fonti interne o esterne all'organizzazione (es. test interni, bollettini di sicurezza, o ricercatori in sicurezza)

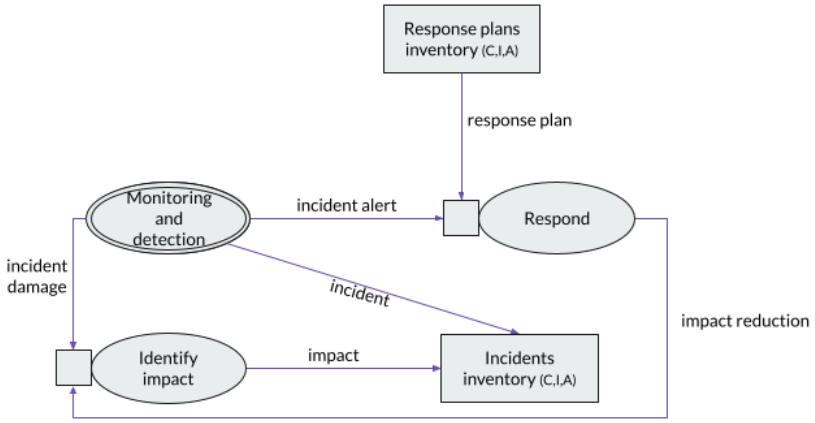


Figure A.88: RS.MI-1: In caso di incidente vengono messe in atto procedure atte a contenerne l'impatto

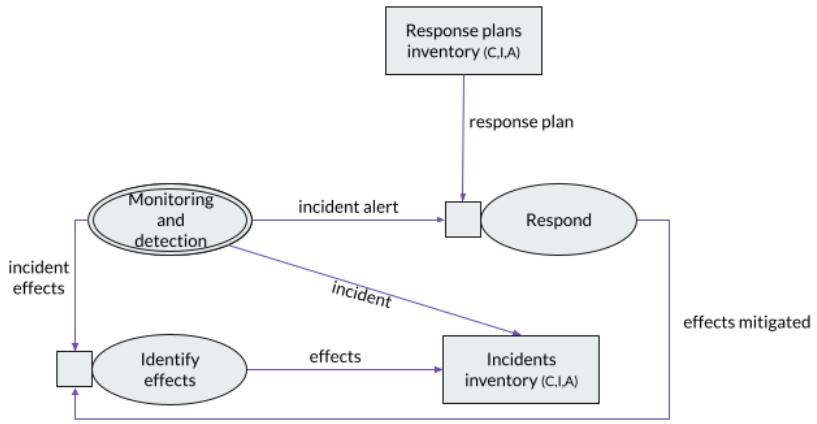


Figure A.89: RS.MI-2: In caso di incidente vengono messe in atto procedure atte a mitigarne gli effetti

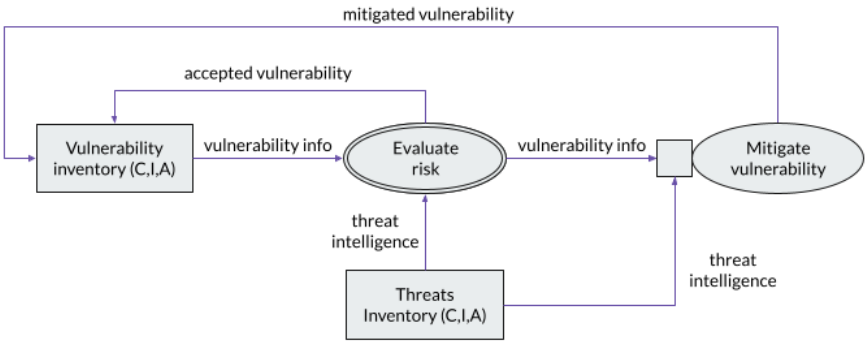


Figure A.90: RS.MI-3: Le nuove vulnerabilità sono mitigate o documentate come rischio accettato

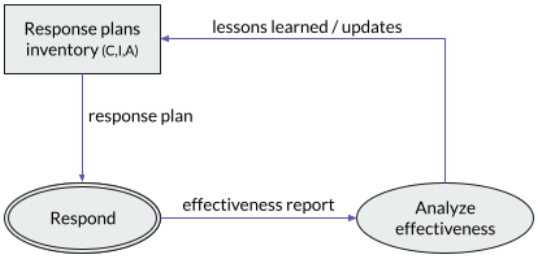


Figure A.91: RS.IM-1: I piani di risposta agli incidenti tengono in considerazione le esperienze passate (lesson learned)

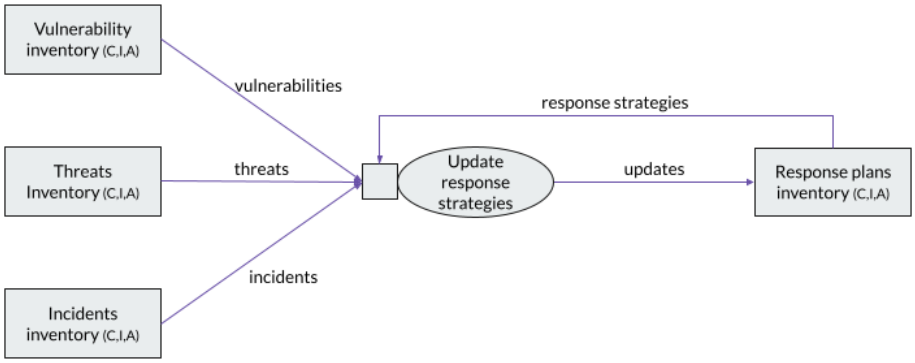
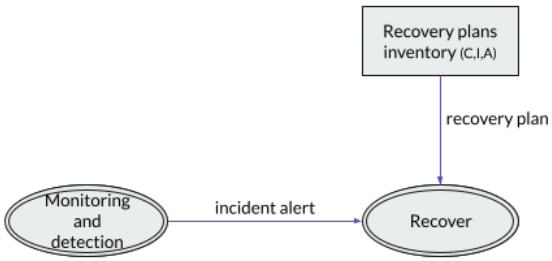


Figure A.92: RS.IM-2: Le strategie di risposta agli incidenti sono aggiornate

A.5 Recover Function



RC.RP-1: Esiste un piano di ripristino (recovery plan) e viene eseguito durante o dopo un incidente di cybersecurity

Figure A.93: RC.RP-1: Esiste un piano di ripristino (recovery plan) e viene eseguito durante o dopo un incidente di cybersecurity

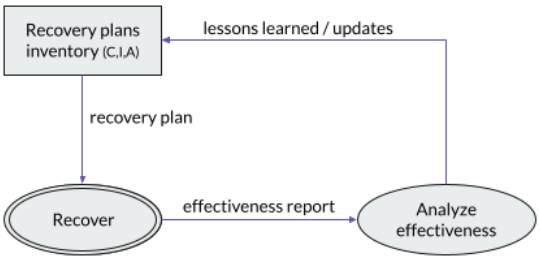


Figure A.94: RC.IM-1: I piani di ripristino tengono in considerazione le esperienze passate (lesson learned)

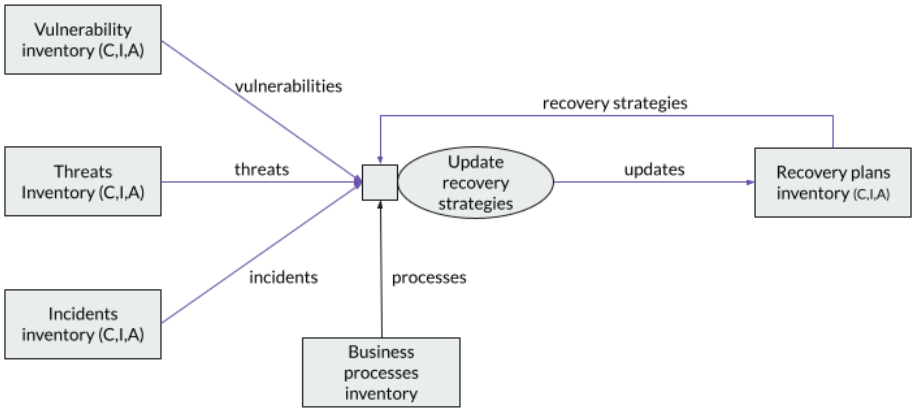


Figure A.95: RC.IM-2: Le strategie di recupero sono aggiornate

Figure A.96: RC.CO-1: A seguito di un incidente vengono gestite le pubbliche relazioni

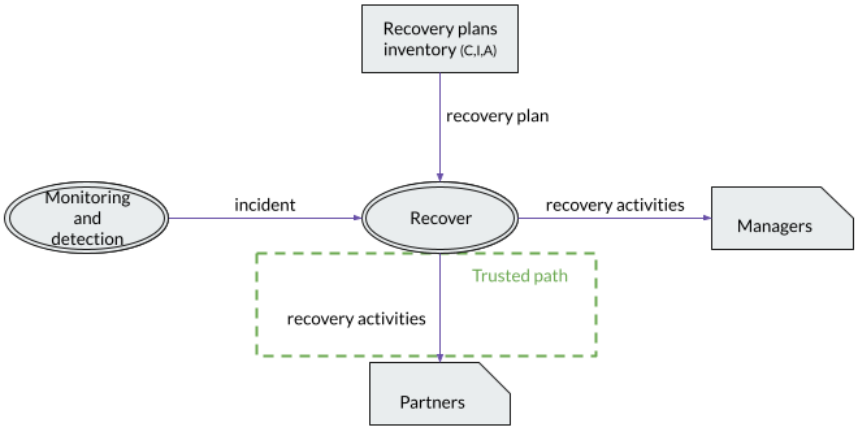


Figure A.97: RC.CO-3: Le attività di ripristino condotte a seguito di un incidente vengono comunicate alle parti interessate interne ed esterne all'organizzazione, inclusi i dirigenti ed i vertici dell'organizzazione