STIX-based theoretical framework for prioritizing relevant CTI Information

Victor Nazianzeno–Le Jamtel

Internship Report

Peter Hagstrom
Thesis Supervisor

RHEA SYSTEM B.V
JONCKERWEG 18
2201 DZ NOORDWIJK
The Netherlands

September 7, 2023

Glossary

asset Something of value for a company, including a software, hardware, program, database, critical information or research sample. While supporting assets are IT-specific infrastructures or software, primary assets are sensitive information and data. . 6, 13

CPE Common Platform Enumeration (CPE) is a structured naming scheme for information technology systems, software, and packages. . 13, 19

CVE The Common Vulnerabilities and Exposures (CVE) system provides a reference method for publicly known information-security vulnerabilities and exposures.. 13

Target of Evaluation In accordance with Common Criteria, an information system, part of a system or product, and all associated documentation, that is the subject of a security evaluation. [4] . 12

threat Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. Also, the potential for a threat-source to successfully exploit a particular information system vulnerability. [5] . 13

Abstract

The ever-evolving landscape of security threats requires a proactive approach to security management. This paper delves into the critical role of risk assessment in prioritizing threats within operational contexts. This study examines how the output of a threat risk assessment software can be leveraged to prioritize new threats and rank their relevancy in operational settings. The design, development and analysis of this framework is based on the RHEA-led SSE4Space risk assessment software for spatial missions. This tool is used as a preliminary step to determine the security and business objectives, as well as to define threat scenarios that can be assessed in the context of the mission. The framework presented in this report makes use of the data collected from the risk assessment of SSE4Space in order to evaluate the relevancy of external, STIX-formatted cyber-threat intelligence in the context of a specific mission and system. In order to demonstrate the feasibility of the theoretical framework, a proof-of-concept (PoC) is provided, making use of a probabilistic Factor Analysis of Information Risk (FAIR) model in order to map both SSE4Space and the CTI's metrics into a single relevancy score. The output of the framework is evaluated qualitatively in order to demonstrate the meaning of the relevancy scoring, as well as the uses of this framework in an operational context.

Contents

Chapter 1

Introduction

Cyber isk assessment is a crucial step in understanding the impact of certain threats on an organization primary and supporting assets, on individuals and on operations.. It provides a structured framework for identifying potential threats, vulnerabilities, and their potential impact through threat scenarios. However, despite its significance, cyber risk assessment is often underutilized in operational contexts, such as in Security Operations Center (SOC) teams. Many organizations tend to view it as a standalone activity rather than an ongoing process integrated into their daily operations. This oversight can lead to missed opportunities to prioritizing emerging threats and vulnerabilities. The manual assessment and prioritization of external Cyber Threat Intelligence can be inefficient as it might leave out important indicators, while incorporating noise into the Intrusion Detection System. As a result, there is a clear rationale for integrating cyber risk assessment as a foundational aspect of continuous security operations.

This report is built upon the RHEA-led SSE4Space risk assessment software, specifically tailored for assessing risks in spatial missions using Secure System Engineering. This tool serves as a preliminary step, establishing the security and business objectives of a mission, while also defining the potential threat scenarios that require evaluation. We further extend our investigation to leverage the data generated from the risk assessment of SSE4Space, using it as a basis to evaluate the relevancy of external cyber-threat intelligence in a mission-specific and system-centered context.

In order to create a relevancy score that takes into account both the severity of the external threat, and the different metrics from the Risk Assessment Software, we will build a scoring model based on the Factor Analysis of Information Risk (FAIR) model. While this model is mainly use for risk analysis, we used custom metrics that can be mapped from the Risk Assessment software and the external threat. The use this model is enhanced by incorporating a Bayesian Network in order to be able to provide fuzzy metrics to the model. More precisely, a metric will not have a discrete state, but rather a probabilistic vector in order to fine-tune the scoring and take into account the uncertainty of certain metrics.

To demonstrate the practicality and efficacy of our theoretical framework, we present a proof-of-concept (PoC). This PoC employs a probabilistic Factor Analysis of Information Risk (FAIR) model, which facilitates the amalgamation of metrics from both SSE4Space and the Cyber Threat Intelligence (CTI) into a unified relevancy score. Subsequently, we assess the output of this framework through qualitative analysis, shedding light on the significance of the relevancy scoring and its potential applications within operational environments.

In a first chapter, we will discuss the background needed to be familiar with the notions discussed in this paper. In a second chapter, the design of the framework will be described, alongside its control flow and the meaning of each metric of the framework. To demonstrate the practicality and efficacy of our theoretical framework, we present a proof-of-concept (PoC) implementing it. A threat model will be presented for the Risk Assessment, and a STIX-formatted Threat Intelligence bundle will be used as the external threat to be scored for relevancy. Finally, we will discuss qualitatively the use of the framework, its compatibility with OWASP and SSE4Space, as well as the relevancy of the metrics. We will also discuss improvements and future works that could lead to a development for practical use.

Chapter 2

Background

2.1 The STIX language

A definition of the STIX language can be found on their website:

> Structured Threat Information Expression (STIX™) is a language and serialization format used to exchange cyber threat intelligence (CTI).
>
> STIX is a very powerful way of sharing Cyber Threat Intelligence. The language is based on JSON format and allows sharing CTI in a consistent and machine-readable format.

There are 18 [6] STIX Domain Objects, including Attack Pattern, Threat Actor, Vulnerability or Malware. A Cybersecurity analyst can then build a report using those objects as well as specific relationships representing edges between objects. It allows a Cybersecurity expert to analyse CTI by considering data as graphs, and using graph theory to cluster, rank and explore threats when they arrive in bulk.

2.1.1 STIX and OpenCTI

OpenCTI [3] is an open-source platform designed for Cyber Threat Intelligence (CTI) management and collaboration. It provides a structured environment for collecting, organizing, and analyzing threat intelligence data, enabling organizations to better understand and respond to cyber threats. OpenCTI facilitates collaboration among security teams, supports the integration of various data sources, and enhances the overall effectiveness of threat intelligence operations in the Cybersecurity landscape.

OpenCTI works with STIX-formatted data, and allows to treat them as graphs, enabling powerful data visualization. An example of such data can be seen in Figure 2.1

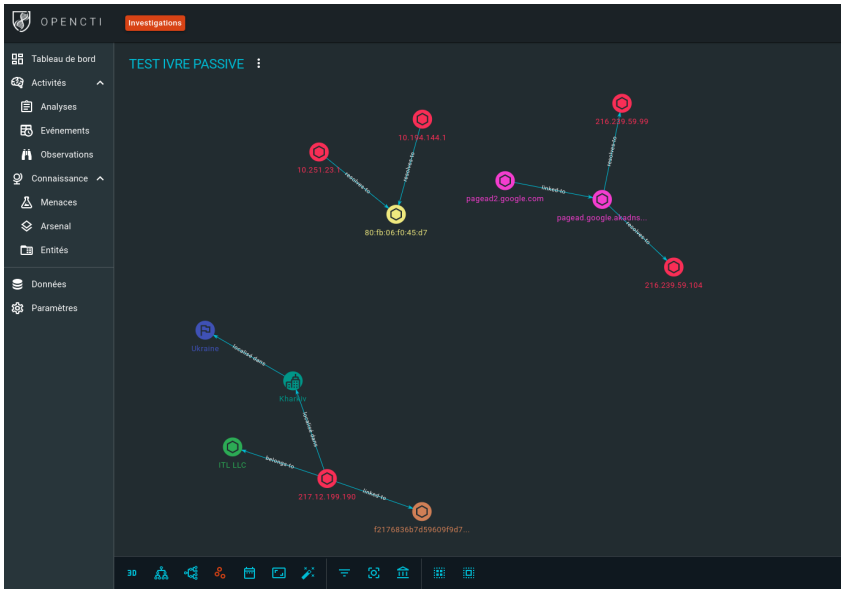


Figure 2.1: Example of an OpenCTI graph

2.2 Overview of SSE4Space

Secure Systems Engineering for Space (SSE4Space) is a framework for applying the security engineering process at a system level. In other words, SSE4Space allows to perform risk assessment on the system itself, taking into account the assets and targets of evaluations, alongside the business requirements, in order to create the security requirements and assess risks the systems. The software is designed to work on ESA unclassified spatial Missions, and performs a cycle per Mission phase.

SSE4Space performs Risk Assessment on defined threat scenarios involving different assets of the Mission. For each scenario, a risk score will be computed using the OWASP [7] risk scoring metric. The process for which risk is evaluated is not detailed in this paper, as SSE4Space only serves as background information for the scope of the framework.

Chapter 3

Design

3.1 Framework overview

A diagram representing the framework logic flow is shown on figure 3.1. The framework developed in this paper prioritizes threats on their likelihood only. This is mainly due to time constraints. The relationship between the framework scoring and the likelihood from OWASP will be detailed in the next section of the design chapter.

We can describe the high-level logic flow in the following steps:

- The framework possesses two input files. On one side, the risk assessment data consists of threat scenarios involving multiple threats on different assets targeted by defined threat actors. A list of Targets of Evaluation and assets are also listed, alongside their geographical and sectoral metrics. A list of threats related to the scenarios are mapped to attack patterns. An enriched list of vulnerabilities, weaknesses and attack patterns are mapped from the assets. It will be discussed later in this chapter. Finally, a list of security controls is provided too, containing both implemented and non-implemented mitigation to different threats.

  On the other side, the external CTI consists of a STIX-formatted bundle, containing standard STIX objects.
- The external CTI is a STIX-formatted bundle. Before mapping the metrics, we use a community detection algorithm, namely the Louvain method, in order to cluster the STIX bundle around threat actors. This allows to separate different threat actors contained in a single bundle, and score them individually.
- For each cluster, an automatic mapping is made from the TRA and the external CTI to a list of set metrics. This list will be described in details later. While some metrics only relate to the TRA data, some of them are purely exclusive to the CTI, and a few are taking into account both of them, in order to evaluate matches on different levels. Each metric possesses 5 level of severity.

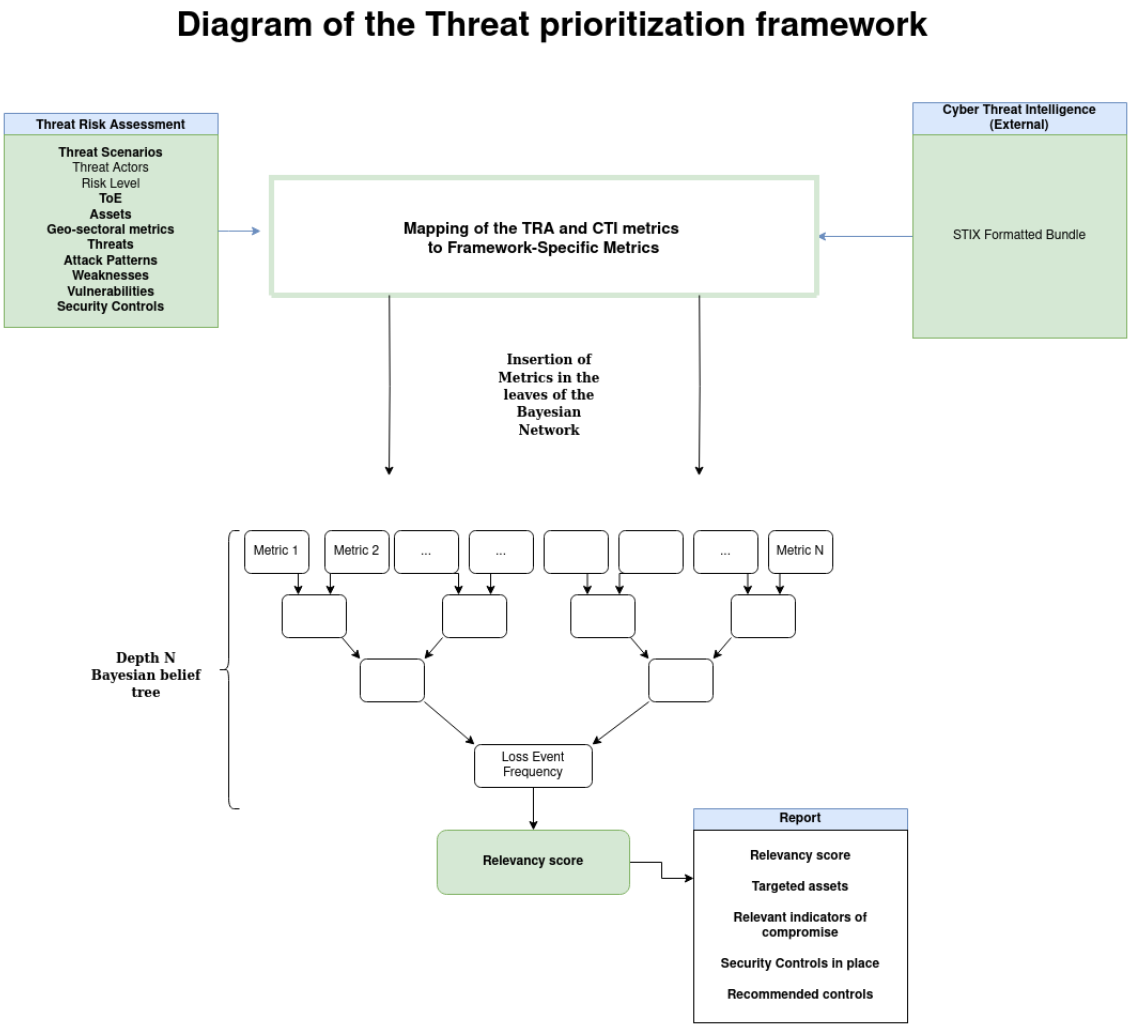


Figure 3.1: High-level diagram of the framework flow

- Those metrics then consists of the leaves of a Belief Bayesian Network, on which we can perform probabilistic inference to compute the root of the tree, which in this case will be the relevancy score.
- finally, alongside the relevancy score, a list of targeted assets, relevant Indicators of Compromise, security controls in place, and recommended controls are all added into a report which effectively prioritizes the threat by giving it a unique rank for the given mission.

3.2 Risk Assessment Metrics

The goal of the framework is to make use of the output of one **SSE4Space** Risk Assessment cycle in an operational context. More precisely, the framework aims at mapping an external threat alongside the Threat and Risk Assessment (TRA) output to a unique relevancy scoring in order to prioritize the threat according to our mission and company data and metrics.

To that extent, an important step of designing the framework consists of mapping the output of an **SSE4Space**'s Assessment to the input of our framework, in a clean, concise and practical format. As the output of the TRA is not fit to be used in such a way, it was necessary to refactor it and create a new data format and enriching it through the addition of a few attributes described in the corresponding subsection.

3.2.1 Refactoring of SSE4Space's output

The following metrics from the Risk Assessment software were used in the new data format:

- **Threat Scenarios** are part of the threat modeling procedure in Risk Assessment. They consist of attack steps taken by a threat actor target a specific asset. It is important to include those scenarios in order to later map external threats to those, and assess their relevancy against our system. A Threat Scenario consists of a unique identifier, a textual description, as well as likelihood, impact and risk metrics. Moreover, threat scenarios include a threat actor, which is defined by its type, sophistication level, motivation and capabilities.
- **Target of Evaluation**: they consists of a system, or sub-system which can be targeted by an adversary. A ToE defines relationships between assets, as well as security objectives to guarantee confidentiality, availability and integrity to the included assets.

- **Assets** are crucial elements of the Risk Assessment data. We can distinguish supporting and primary assets. Supporting assets can be software, hardware, or a database that contains, manages or handles sensitive data or information for the company, called primary assets. Assets are referring to a list of ToEs as well as a list of Threat Scenarios to which they belong to. Assets have been enriched with additional attributes, as described in the next subsection.
- **Threats** are the core components of the threat scenarios. They are defined through techniques originating from catalogues , such as ATT&CK or SPACESHIELD.

3.2.2 Enrichment of SSE4Space

During the research step, a few attributes related to the risk assessment data were added.

- **CPE mapping**: One of the core aspect of the enrichment of the TRA is the addition of product enumerations linked to each asset. The CPE provides a list of software versions that can be bilaterally linked with CVEs, i.e. known vulnerabilities. This allows to refer to an asset through its software version, as well as to link it to a known vulnerability and therefore evaluate its risk.
- **Enriched catalogue**: Having CPE-linked assets, it was then possible to enriched the risk assessment data by mapping each product version to a list of vulnerabilities. Those vulnerabilities were then mapped to known weaknesses (CWEs), providing a higher-level of abstraction as they don't relate to a specific software, but rather a faulty practice or a category of vulnerabilities. In a third entry, we added attack patterns (CAPEC), which are patterns used by malicious actors to exploit a set of weaknesses.
- **Security controls**: While security controls related to a scenario were already present in the original risk assessment data, some attributes were added in order to enrich the mapping with the external threats. More precisely, each identified threat / technique is assigned one or more security controls that mitigates it, with a *control strength* metric defining how efficient the control performs against this threat. T

3.3 Community detection

Louvain method is a community detection algorithm for extracting non-overlapping clusters from a graph [8]. While the mathematical details of the method won't be discussed here, it is important to note that Louvain method relies on the notion of modularity, which expresses the strength of division of a graph. More precisely, if a graph expresses high modularity, it means that there are highly connected nodes in each module, and every modules are sparsely interconnected.

Another key aspect is the fact that modularity takes into account the weights of the edges, which allows to center communities around specific nodes.

In the case of our framework, we chose to center the clusters around threat actors, malware, and campaigns. To do so, we gave more weight to relationships between such nodes and other objects. A visual explanation of Louvain method is given in the Proof-Of-Concept chapter.

3.4 Framework Metrics

This section aims at providing a detailed description of each metric of the framework, and how they are mapped from the risk assessment and the external threat data.

3.4.1 Overview

Figure 3.2 gives an overview of the metrics tree used for computing the relevancy scoring.

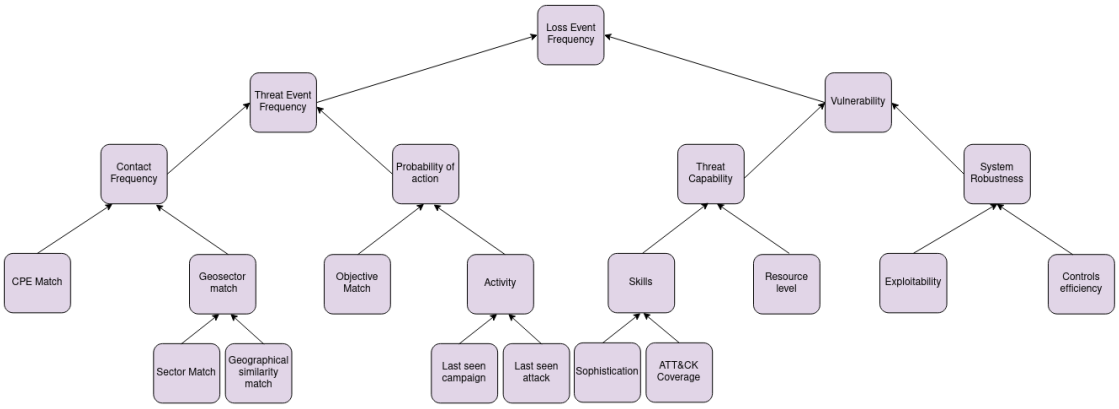


Figure 3.2: Overview of the metrics tree

We can distinguish metrics by their depth in the tree. The root, namely the Loss Event Frequency, corresponds to the relevancy score for the likelihood. The next two levels of the tree correspond to the upper metrics, which are mapped from the FAIR model. Finally, the lower metrics are customized and mapped from the Threat risk assessment data and the external threat bundle.

3.4.2 Lower metrics

We can distinguish 14 lower metrics in our framework.

- **CPE Match**: relates to whether the external CTI contains a threat targeting a software version present in the RA data.
- **Geosector Match**: it is comprised of two submetrics, namely the sector match and the geographical match. The first one checks for sectoral keywords in the description of threat actors, campaigns or malware in the CTI, which the second one checks for linguistic or regional similarities between a threat's target and the RA data.
- **Objective match**: this metrics compares the goals of threat actors and campaigns in the external threat with the threat scenarios defined in the risk assessment software
- **Activity**: consisting of the last seen campaign and the last seen attack. This metrics extract timestamps and evaluate how recent the campaign, attack, or threat actor's activity is.
- **Skill**: this metric describes the skill level of a threat actor detected in the external CTI. It looks at it sophistication level, as well as the ATT&CK Coverage. The latter corresponds to the number of techniques employed by a threat actor in term of the ATT&CK catalogues. It is an indicator of the capabilities of the malicious actor.
- **Resource level**: the resource level determines the size of a threat actor, or its scales. It is often related to the identity which relates to the threat actor, ranging from an isolated individual to a nation-wide hacker team.
- **Exploitability**: exploitability corresponds to a CVSS metric taken from CVEs. It indicates how realistic the exploitation of a certain vulnerability is.
- **Controls strength**: this metric takes into account the implemented security controls in the risk assessment software for the given threat model. It takes into account the controls applied to matching threats also found in the external CTI bundle.

3.4.3 Upper metrics

Upper metrics have a much stronger semantic meaning. They are taken from the FAIR model, as well as a from a paper which implements evidence-based threat prioritization using FAIR [2]. We can use the definition given in this article as their meaning and usage in the framework are identical.

- **Contact frequency**: The probable frequency with which a threat actor will come into contact with an asset. It takes into account the CPE match and the Geosectoral match.
- **Probability of action**: The probability that a threat agent will act against an asset once contact has occurred. It includes the objective match and the activity metric.

- **Threat Event Frequency**: The probability that a threat actor will act against an asset. It is composed of the contact frequency and the probability of action.
- **Threat Capability**: The probable level of force a threat agent is capable of applying against an asset. It uses the skill and resource levels of threat actors identified in the bundle.
- **System Robustness**: this metric has been customized in order to accommodate the mapping from both the risk assessment and the external threat. It corresponds to the strength of the system in term of its exploitability and the implemented controls.
- **Vulnerability**: The probability that the threat actor's actions will be successful. It depends on the threat actor's capabilities, as well as how robust the system is.

3.4.4 Relevancy score

The relevancy score corresponds to the root of the tree. The metric in place equivalent this score is called the *Loss Event Frequency*, which is defined in [2] as

> the probable frequency within a given time frame with which a threat agent will inflict harm on an asset.
> It is important to note the probabilistic dimension here, which referes to likelihood. This is the subject of the next section.

3.5 Scoring

3.5.1 Relevancy score and likelihood

Limitations to likelihood

As stated at the beginning of this chapter, the development of the relevancy score in this framework is based on a notion of likelihood only. A first reason is that the work done in the evidence-based prioritization article [2] is also centered around the notion of *Loss Event Frequency*, which determines the probability of an asset being successfully targeted in a given time frame. Another reason was due to temporal constraints, as the development of the framework as well as the proof of concept took place during 7 weeks.

Relevancy defined with likelihood

Relevance, in the context of an cyber threat intelligence, refers to how pertinent a specific threat is to the system in question, taking into account the

system's unique characteristics, assets, and vulnerabilities and the threat scenarios defined preemptively. A relevant threat is one that poses a direct and significant risk to the system's integrity, confidentiality, or availability. It is also one that is tied to the organization through the motivation of the threat actor, its targeted sector, or its geographical characteristics. On the other hand, likelihood assesses the probability of the threat materializing, considering various factors such as historical data, threat actors' capabilities, and the robustness of the system. There is a clear intersection between relevance and likelihood, even though it might not form an exhaustive relevancy assessment.

3.5.2 Bayesian Belief Networks

The use of the FAIR model is not necessarily essential to the development of the framework. Indeed, FAIR is simply an ordering of metrics which allow risk to be analysis with respect to an asset. However, if FAIR was chosen, it is because some research has been done in threat prioritization using FAIR by incorporating Bayesian networks in the procedure.

To begin with, we can define a Bayesian network as a probabilistic graphical models were we define variables that are related by conditional dependencies (conditional probability table). In the case of this framework, we apply the idea of a Bayesian Network to the FAIR model by having a directed binary tree topology, starting from the leaves to the root (see figure 3.2).

There are two advantages to using Bayesian networks in this context.

- Given a subset of variables having a state, we can infer the other variable's states probabilistically. In the context of this paper, it means that starting from the leaves, we can probabilistically infer the value of the relevancy score.
- Each variable / metric has a state. We consider 5 states, namely (very) low, medium and (very) high. Using a bayesian network, it is possible to enhance this discrete characterization by using multiple state for a single metric. For example, a metric can now be 60% low, and 40% very low. This fine tuning can increase the accuracy of the final score as this allows to refine the mapping from the risk assessment metrics as much as we want with respect to this 5-states scale.

3.5.3 Probability table

In order for the Bayesian network to be functional, it needs to rely on conditional probability tables. In the case of a binary-tree topology, conditional tables are defined for a variable and its two parents. For example, a table is

defined for the activity, and its two parents, namely the last seen campaign and the last seen attack.
The probability table then represents the influence of each of the parent metric on its child metric. Modifying those tables results in a different scoring, as they are setting weights for computing the score itself.
Those tables are specific to a mission, and need to be set up by an security expert beforehand, in order to specify how influential each metric is in the tree. This is one of the only step of this framework which needs some expert interaction, as it doesn't rely on the risk assessment data, but rather exists as a complement of it to rank threats relevancy.

3.6 Output

The output of the framework consists of 4 elements. The main attribute is the relevancy score for each detected community in the STIX-bundle. The other attributes were deemed useful for an operational context. This is discussed in later chapters.

3.6.1 Relevancy score

After obtaining a probability vector from the Loss Event Frequency, we can perform a weighted sum to obtain a unique score.
Formally, we have the probability vector $\mathbf{v} = [v_l, v_l, v_m, v_h, v_h]$ and the weight vector $\mathbf{w} = [1, 2, 4, 8, 16]$. The weighted sum is given by:

$$\sum_{i=1}^{5} v_i \cdot w_i = v_l \cdot 1 + v_l \cdot 2 + v_m \cdot 4 + v_h \cdot 8 + v_h \cdot 16$$

.
This score is then normalized to a scale from 0 to 10, taking into account the distribution of the Bayesian network. This is discussed in details in the Proof-Of-Concept chapter.

3.6.2 Indicators of Compromise

Each bundle consists of 0 or more indicators of compromises. They consists of hashes, ips, or URL that indicates a threat or malware present in the bundle. In STIX, they come with a pattern that can be used in an Intrusion Detection System in order to spot suspicious IP addresses, or files going through an organization network.
The IoCs of each cluster are added to the output alongside the relevancy score.

3.6.3 Products enumeration

As discussed earlier, CPEs are software versions linked to an asset in the risk assessment data. Therefore, if a product version was identified in the external threat, either through a vulnerability or an attack pattern, the corresponding asset will be included in the output.

3.6.4 Mitigations

In the risk assessment output, there a controls listed for each threat. When finding a matching threat or technique in the external CTI, it is then possible consider which controls have been implemented to mitigate the threat, and which controls can be recommended in order to provide the operational engineer a course of action.

In order to match a control to a threat, it is possible to refer to the NIST 800-53 framework [1], which provides controls that can be mapped to the ATT&CK catalogue. While some controls might not be applicable to our systems, it still provides a course of action that will guide the operational team as it can link the mitigation to an asset.

Chapter 4

Proof-of-Concept

The Proof-Of-Concept (PoC) presented in this report is essentially an application of the framework with an example Risk Assessment JSON file and a example STIX-formatted CTI bundle. We will begin by presenting the scenario and the context of the PoC, before describing the threat scenarios and the threat modeling phase. We will then view the CTI bundle and observe how each metric is mapped, and how each cluster within the bundle is scored. We will briefly go through the output of the framework as well, and discuss the scoring distribution and how to normalize it.

4.1 Scenario

4.1.1 Description

AstraCorp is a R&D company dedicated to space engineering and space-compatible nano-components. It collaborates with public institutions on a research-level, as well as for database management. The company's goal is to perform research on different fields related to space engineering, as well as to store sensitive mission data for processing and testing. AstraCorp also works with private stakeholders investing money in Space R&D. AstraCorp possesses a Cybersecurity department, with a Risk Assessment as well as an Operational teams.

4.1.2 Context

Astracorp works with the public institution GovSpace for R&D purposes. The following primary and supporting assets are used in their collaboration:

Primary assets

- Sensitive data about mission samples
- Medical histories of astronauts

Supporting assets

- Webserver used for querying data in HTTP.
  - cpe:2.3:a:hypr:keycloak-authenticator:-:*:*:*:*:*:*:*
  - cpe:2.3:a:f5:nginx:1.1.1:*:*:*:*:*:*:*
- Web application made with node.js.
  - cpe:2.3:a:openjsf:express:4.3.1:*:*:*:*:node.js:*:*
  - cpe:2.3:a:nodejs:node.js:6.2.1:*:*:*:lts:*:*:*
- Database for AstraCorp sensitive data storage
  - cpe:2.3:a:oracle:mysql:8.0.18:*:*:*:*:*:*:*
- Cloud for Medical histories
  - cpe:2.3:a:microsoft:azure-open-management infrastructure:-:*:*:*:*:*:*:*

4.1.3 Threat Actors

Advanced

We define an advanced threat actor whose goal is to access and leak sensitive data from GovSpace's cloud.

- adversaryType: spy
- sophistication: strategic
- opportunity: high
- motivation: critical
- capabilities: critical

We define an advanced threat actor whose goal is to access and leak sensitive data from AstraCorp's database.

- adversaryType: crime-syndicate
- sophistication: expert
- opportunity: high
- motivation: critical
- capabilities: high

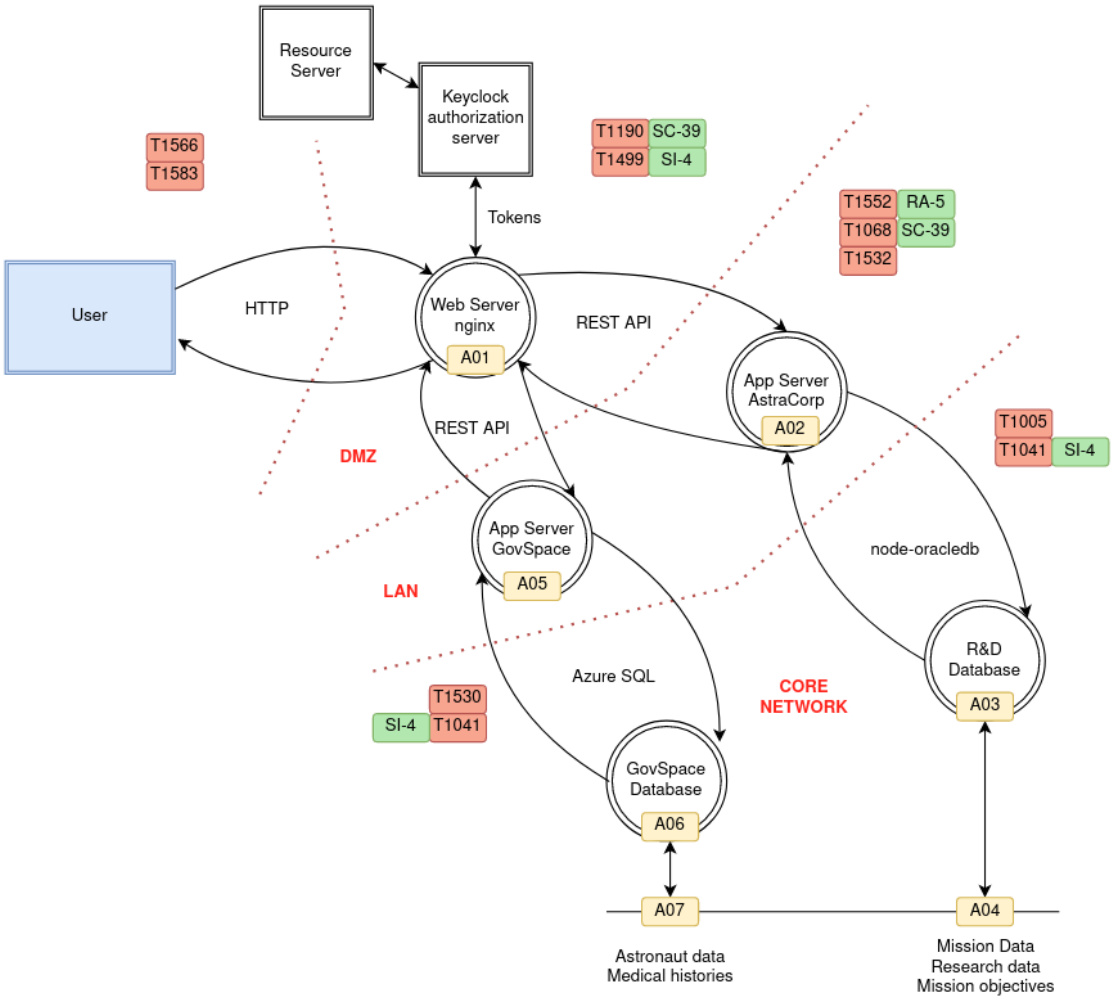
Intermediate

We define an intermediate threat actor whose goal is to prevent accessibility through denial of service of the nginx webserver.

- adversaryType: hacker
- sophistication: advanced
- opportunity: medium
- motivation: high
- capabilities: medium
- threatVectors: DENIAL-OF-SERVICE

4.2 Threat Model

The Threat Model can been seen in the following figure:



The red labels designate techniques from the ATT&CK catalogue, while the green labels are controls from NIST 800 – 53. Finally, the yellow labels are assets. The name and description of each threat,control and asset can be found in the risk assessment JSON file on the github repository (see Appendix).

4.3 CTI Prioritization

4.3.1 STIX graph

Figure 4.1 shows the CTI bundle.

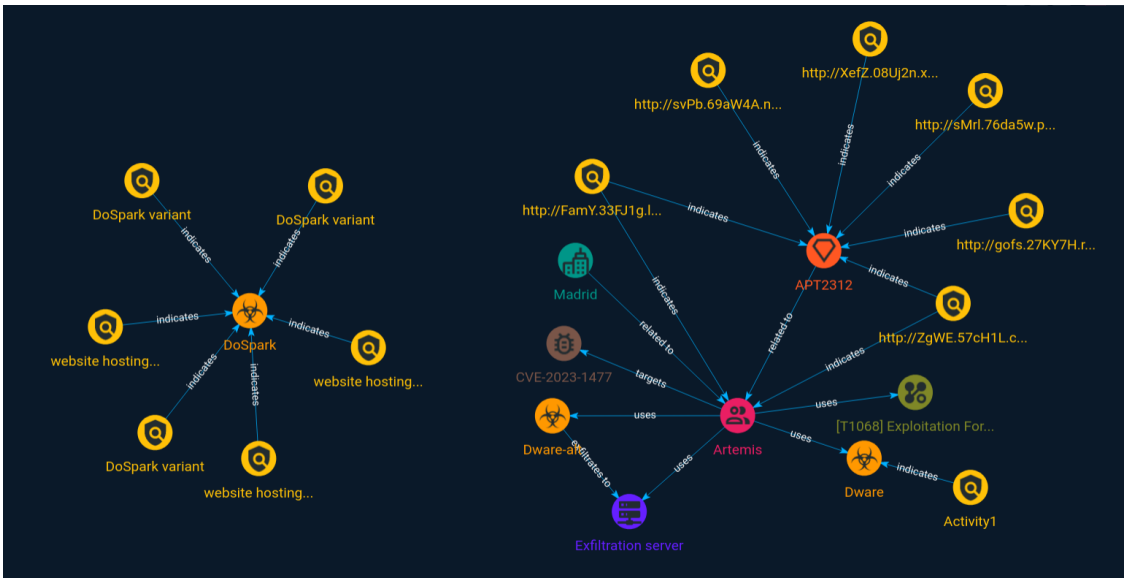


Figure 4.1: CTI graph for the Proof Of Concept

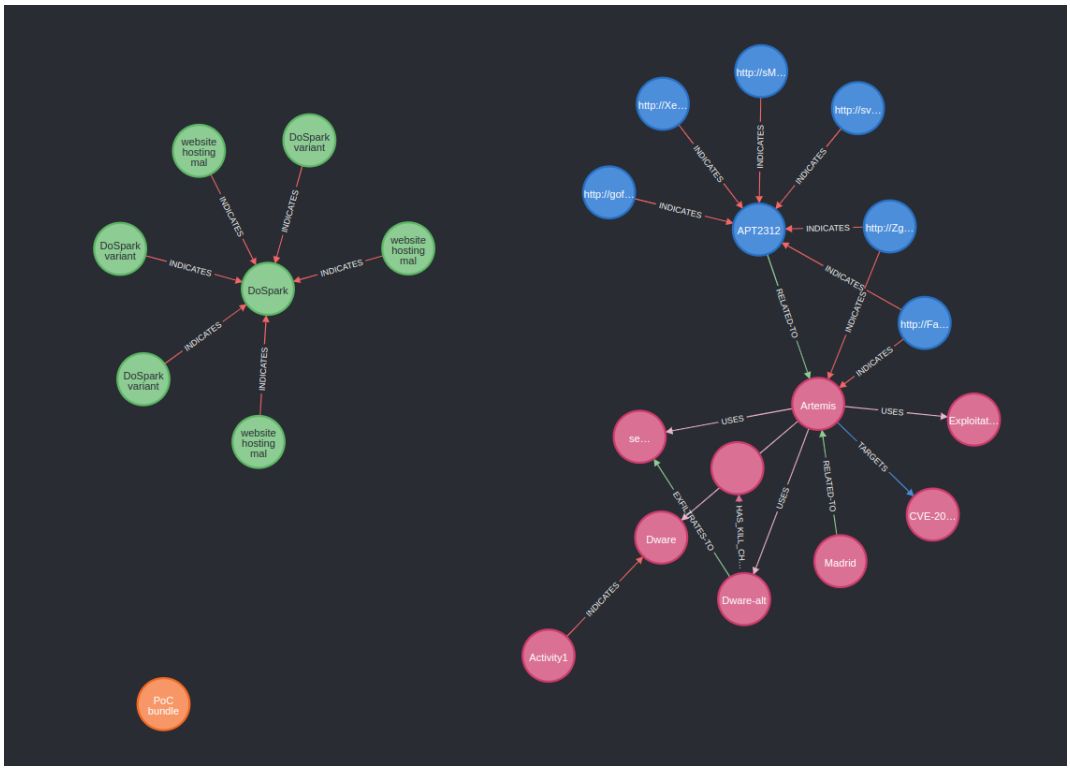
We can extract the following information from it.

- The graph is partitioned.
- The first subgraph contains a malware and IoCs.
- The second graph is more complex, and contains a threat acotr (Artemis), an intrusion set (APT), a few malware linked to the threat actor, a vulnerability and attack pattern, an infrastructure hosting a malware, and many IoCs.

It would seem like the bundle could be separated into 3 communities based a first view of the components and their relationships, as the intrusion set and the threat actors seem to have an associative relationships, but not any kind of inheritance.

4.3.2 Clustering

In order to cluster the graph using Louvain method, we mapped the STIX bundle to a neo4j graph using a script. This script can also be found on the repository with a ".cypher" extension, and can be applied to any graph. Indicator relationships have lower weights than the rest, in order to center the clusters around threat actors and malware. Having mapping the bundle to neo4j, we obtain the following clusters:



We can see that there are 3 clusters, without counting the isolated bundle object itself, which is discarded before further analysis.

4.3.3 Scoring of each bundle

The framework for automating the scoring of the each cluster is developed in Python with a command-line interface.

4.4 Output and Scoring

4.4.1 Format

4.4.2 Scoring distribution

In order to to map a score to a relevancy level, it is important to understand the distribution of the Bayesian Tree outputs, as it is unlikely they will follow a normal distribution. The Bayesian network inference was ran on 50,000 random samples to approximate the distribution of values for the final LEF score. In other words, the base metrics received a random score between 0 and 4, and Bayesian Inference was performed to compute the *LEF* score based on those metrics. The resulting approximated distribution is shown on Figure 4.2

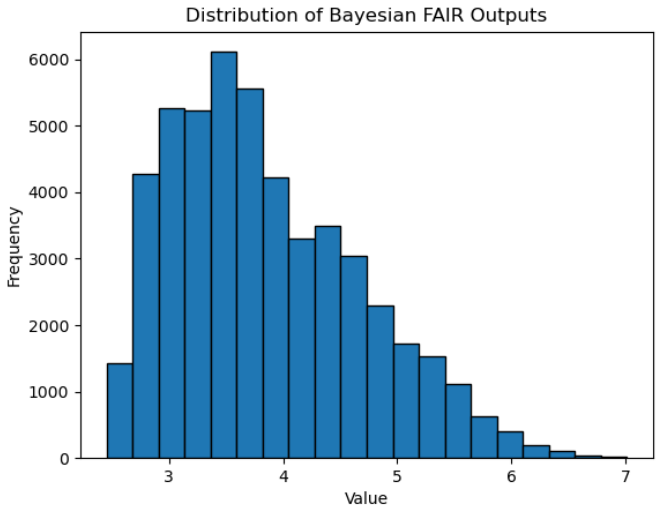


Figure 4.2: Distribution of values for the LEF score

It is clear that the distribution is centered around 3.5, and is not uniform, especially on higher values. The lower and upper bounds are respectively 2.5 and 7. In order to convey a meaningful mapping to relevancy, the scoring output was normalized before being assigned a relevancy label.

This can be approximately achieved using percentile ranking [9], which, for a given value in a distribution, calculates the percentage of scores that are less than that value. The percentile graph can be seen on Fig 4.3.

In order to convince ourselves that this technique does approximate to a uniform distribution, we can plot the percentiles in term of their frequency, shown on Fig 4.4

Figure 4.3: Percentile rank for LEF values

The normalized output of the Bayesian Tree then allows to define meaningful thresholds for relevancy. For simplicity, we defined 4 thresholds at $0.2, 0.4, 0.6$ and 0.8. They create 5 intervals mapped to relevancy labels, ranging from **Very Low Relevancy** to **Very High Relevancy**. For example, if a bundle is ranked at 0.53, it will receive the **Medium Relevancy** label.

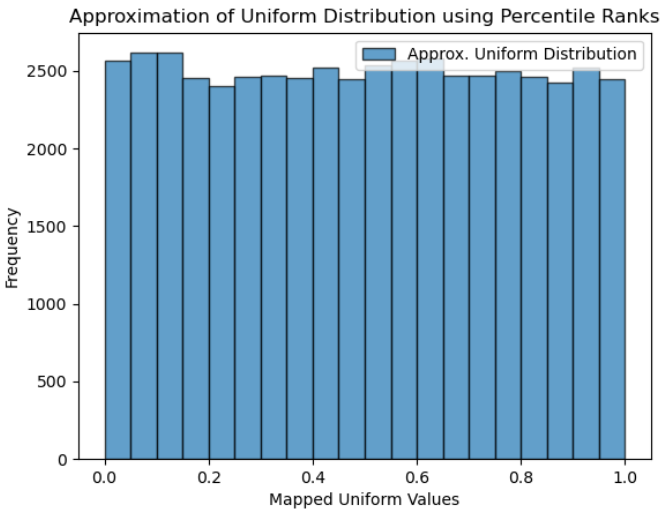


Figure 4.4: Percentile frequencies

Chapter 5

Discussions

5.1 Qualitative Assessment of the Proof Of Concept

More qualitative Evaluated on the base on real operations: in that case its complicated * few test cases (hypothesis -> experiment -> observation) * identify criterion in the hypothesis * analysis comes after

5.2 Utility of the model for Operational Use

* how to apply the output to an operational phase (mitigation, courses of action, targeted assets, IoCs per "community"

5.3 Comparison with other Threat Prioritization Frameworks

Chapter 6

Future Work and Improvements

also add: * future work on kill chain phase * future work on NLP * future work on owasp-compatible framework * future work on strategic

6.1 Changes to SSE4Space

6.2 Fine-tuning of the Metrics

* here talk about the scoring, thresholding, evidence-based vs machine learning-based.

6.3 Feedback loop

* reinforcement learning on the influence matrix *reinforcement learning on the cluster algorithm by checking previous bundles and creating connections to new ones to have more meaningful clusters at the end, therefore the relevancy is more accurate.

6.4 Related Work

Chapter 7

Conclusion

* bridges between risk assessment and CTI

Bibliography

[1] *Complete 8500 Control List — stigviewer.com*. https://www.stigviewer.com/controls/800-53. [Accessed 07-09-2023].

[2] *Evidence-Based Prioritization of Cybersecurity Threats — isaca.org*. https://www.isaca.org/resources/isaca-journal/issues/2021/volume-6/evidence-based-prioritization-of-cybersecurity-threats. [Accessed 07-09-2023].

[3] *GitHub - OpenCTI-Platform/opencti: Open Cyber Threat Intelligence Platform — github.com*. https://github.com/OpenCTI-Platform/opencti. [Accessed 05-09-2023].

[4] NIST. *target of evaluation (TOE) - Glossary | CSRC*. [Online; accessed 4-September-2023]. 2023. URL: https://csrc.nist.gov/glossary/term/target_of_evaluation.

[5] NIST. *threat - Glossary | CSRC*. [Online; accessed 1-September-2023]. 2023. URL: https://csrc.nist.gov/glossary/term/threat.

[6] OASIS Cyber Threat Intelligence (CTI) TC. *STIX™ Version 2.1*. [Online; accessed 2-September-2023]. 2019. URL: https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html.

[7] *OWASP Risk Rating Methodology | OWASP Foundation — owasp.org*. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology. [Accessed 05-09-2023].

[8] Wikipedia contributors. *Louvain method — Wikipedia, The Free Encyclopedia*. [Online; accessed 7-September-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=Louvain_method&oldid=1173553358.

[9] Wikipedia contributors. *Percentile rank — Wikipedia, The Free Encyclopedia*. [Online; accessed 31-August-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=Percentile_rank&oldid=1158568027.

Appendix A

Project repository

The Proof-of-Concept is hosted in a private repository available here. In case you need access to the repository, you can contact the author at **v.nlejamtel@gmail.com**.