



MASTER THESIS

**METHODOLOGICAL FRAMEWORK TO COLLECT,
PROCESS, ANALYZE AND VISUALIZE
CYBER THREAT INTELLIGENCE DATA**

Lucas José Borges Amaro

Brasília, October 2022

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**METHODOLOGICAL FRAMEWORK TO COLLECT,
PROCESS, ANALYZE AND VISUALIZE
CYBER THREAT INTELLIGENCE DATA**

***FRAMEWORK METODOLÓGICO PARA COLETAR,
PROCESSAR, ANÁLISAR E VISUALIZAR
DADOS DE *CYBER THREAT INTELLIGENCE****

LUCAS JOSÉ BORGES AMARO

**ORIENTADOR: WILLIAM FERREIRA GIOZZA, DR.
COORIENTADOR: ROBSON DE O. ALBUQUERQUE, DR.**

**DISSERTAÇÃO DE MESTRADO PROFISSIONAL EM
ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO: PPEE.MP.022
BRASÍLIA/DF: OUTUBRO – 2022**

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

**METHODOLOGICAL FRAMEWORK TO COLLECT,
PROCESS, ANALYZE AND VISUALIZE
CYBER THREAT INTELLIGENCE DATA**

Lucas José Borges Amaro

*Submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Mestre*

Banca Examinadora

Prof. William Ferreira Giozza, Ph.D, FT/UnB _____
Orientador

Prof. Robson de O. Albuquerque, Ph.D, _____
FT/UnB
Coorientador

Prof. Adriano Mauro Cansian, Ph.D, UNESP _____
Examinador externo

Prof. João José Costa Gondim, Ph.D, UnB _____
Examinador interno

FICHA CATALOGRÁFICA

AMARO, LUCAS JOSÉ BORGES

METHODOLOGICAL FRAMEWORK TO COLLECT, PROCESS, ANALYZE AND VISUALIZE CYBER THREAT INTELLIGENCE DATA [Distrito Federal] 2022.

xvi, 45 p., 210 x 297 mm (ENE/FT/UnB, , Engenharia Elétrica, 2022).

- Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Cyber Threat Intelligence

2. Sharing

3. Vulnerabilities

4. Framework

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

AMARO, L. J. B. (2022). *METHODOLOGICAL FRAMEWORK TO COLLECT, PROCESS, ANALYZE AND VISUALIZE CYBER THREAT INTELLIGENCE DATA* . , Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 45 p.

CESSÃO DE DIREITOS

AUTOR: Lucas José Borges Amaro

TÍTULO: METHODOLOGICAL FRAMEWORK TO COLLECT, PROCESS, ANALYZE AND VISUALIZE CYBER THREAT INTELLIGENCE DATA .

GRAU: MESTRE ANO: 2022

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Do mesmo modo, a Universidade de Brasília tem permissão para divulgar este documento em biblioteca virtual, em formato que permita o acesso via redes de comunicação e a reprodução de cópias, desde que protegida a integridade do conteúdo dessas cópias e proibido o acesso a partes isoladas desse conteúdo. O autor reserva outros direitos de publicação e nenhuma parte deste documento pode ser reproduzida sem a autorização por escrito do autor.

Lucas José Borges Amaro

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

DEDICATION

First of all, I would like to thank God, who enabled me to achieve this achievement and gave me the strength to walk this long path.

To my dear and beloved wife, Maria Luíza Amaro, all my gratitude for the support she has given me since my engineering degree until now.

For my parents, I wouldn't be here without them and without all the effort they've made throughout their lives to raise me as a man who pursues his dreams and doesn't give up on them.

I would like to thank my supervisors, William Ferreira Giozza and Robson de Oliveira Albuquerque, for their attention, dedication and support, without which this work would not have been possible.

I also thank my friends and colleagues who helped me during the development of the research, especially my dear friends Fillipe Barros and Trajano Melo.

Finally, I would like to thank the University of Brasília and, in particular, the faculty of the Department of Electrical Engineering, for their exceptional quality and for all the excellence in the training of engineers, masters and doctors.

"Those who teach learn by teaching. And those who learn teach by learning".

ACKNOWLEDGEMENTS

First of all, I would like to acknowledge the University of Brasília, which raised me since my 17 years old as a Networking Engineer, until now at my 25 years old, a technical specialist in AIOps and Telco at IBM and almost a Master.

Finally, I acknowledge IBM that took me in as a trainee in 2020 in the beginning of the covid-19 pandemic and gave me almost everything I have today.

ABSTRACT

Cyber attacks have increased in frequency in recent years, affecting small, medium and large companies, creating an urgent need for solutions capable of helping on the mitigation and response of such threats. Thus, with the increasing number of cyber attacks, we have a large amount of threat data from heterogeneous sources that needs to be ingested, processed and analyzed to obtain useful insights for their mitigation. This work proposes a methodological framework to collect, organize, filter, share and visualize cyber-threat data to mitigate attacks and fix vulnerabilities, based on an eight-step Cyber Threat Intelligence model with timeline visualization of threats information and analytic data insights. We developed a tool to address those needs in which the cyber security analyst can insert threat data, analyze them and create a timeline to get insights and a better contextualization of a threat. Results show the facilitation of understanding the context in which the threats are inserted, making the mitigation of vulnerabilities more effective.

Keywords: Cyber Threat Intelligence; sharing; vulnerabilities; visualization; analytics; temporal; framework

RESUMO

Os ataques cibernéticos têm aumentado em frequência nos últimos anos, afetando pequenas, médias e grandes empresas, criando uma necessidade urgente de soluções e ferramentas capazes de auxiliar na mitigação e resposta às ameaças. Assim, com o aumento do número de ataques cibernéticos, temos uma grande quantidade de dados de ameaças de fontes heterogêneas que precisam ser ingeridos, processados e analisados a fim de obter percepções úteis para sua mitigação. Este trabalho propõe uma estrutura metodológica para coletar, organizar, filtrar, compartilhar e visualizar dados de ameaças cibernéticas para mitigar ataques e corrigir vulnerabilidades, com base em um modelo de inteligência de ameaças cibernéticas de oito etapas com visualização em linha do tempo de informações de ameaças e insights de dados analíticos. Desenvolvemos uma ferramenta para atender a essas necessidades em que o analista de segurança cibernética pode inserir dados de ameaças, analisá-los e criar uma linha do tempo para obter insights e uma melhor contextualização de uma ameaça. Os resultados mostram a facilitação da compreensão do contexto em que as ameaças e os atacantes estão inseridos, tornando o uso de Cyber Threat Intelligence mais amigável ao usuário final.

Palavras-chave: Cyber Threat Intelligence; Compartilhamento; Vulnerabilidades; Visualização; Analytics; Temporal; Framework

CONTENTS

DEDICATION	VI
ACKNOWLEDGEMENTS	VII
ABSTRACT	VI
RESUMO	VI
LIST OF FIGURES	X
LIST OF TABLES	XI
LIST OF ACRONYMS	XII
1 INTRODUCTION	1
1.1 MOTIVATION AND OBJECTIVE	2
1.2 RESEARCH CONTRIBUTIONS	3
1.3 OUTLINE OF THIS WORK	3
2 BACKGROUND AND RELATED WORKS	4
2.1 CYBER THREAT INTELLIGENCE	4
2.2 THREAT HUNTING	5
2.3 CTI STANDARDS AND FRAMEWORKS	7
2.3.1 MITRE ATT&CK	7
2.3.2 CYBOX	9
2.3.3 STIX	9
2.3.4 TAXII	10
2.4 RELATED WORKS	10
2.4.1 CTI PROTOCOLS AND STANDARDS	10
2.4.2 CTI MAIN CHALLENGES AND TECHNICAL IMPROVEMENTS	11
3 PROPOSED SOLUTION AND MODEL	13
3.1 METHODOLOGY	13
3.2 CONTEXTUALIZATION	14
3.3 SOLUTION OVERVIEW	17
3.3.1 MANAGEMENT STEP	17
3.3.2 INDEXER STEP	19
3.3.3 COLLECT STEP	19

3.3.4	GENERATE STEP	19
3.3.5	SEARCH STEP	20
3.3.6	SHARE STEP	20
3.3.7	VISUALIZATION STEP	20
3.3.8	ANALYSIS STEP	21
3.4	FEEDS COLLECTION.....	21
3.4.1	VISUALIZATION STEP	21
3.4.2	ANALYSIS STEP	23
3.5	TIMELINE VISUALIZATION.....	23
3.5.1	DIFFERENCES BETWEEN MODELS.....	25
4	CASE STUDIES AND DISCUSSION	26
4.1	FEEDS SELECTION AND EXTRACTION.....	26
4.1.1	TRANSFORMATIONS APPLIED TO STORE FEEDS	27
4.1.2	DATA EXTRACTION FROM FEEDS	29
4.2	PEGASUS SPYWARE ANALYSIS AND TIMELINE	32
4.2.1	MITRE ATT&CK ANALYSIS	34
4.3	SOLARWINDS SUNBURST TIMELINE	36
4.3.1	MITRE ATT&CK ANALYSIS	36
4.4	CHAPTER SUMMARY	39
5	CONCLUSIONS AND FUTURE WORK	40
5.1	FUTURE WORK	40
	BIBLIOGRAPHY	42

LIST OF FIGURES

1.1	Common Cyber Threat Intelligence steps	1
2.1	Desirable steps and objectives for CTI	5
2.2	Threat Hunting framework	6
2.3	CybOX core objects categories	9
2.4	Client-server TAXII model.....	10
3.1	Biggest challenges faced by Threat Hunter teams.....	14
3.2	CTI 8-step proposed model	16
3.3	Tool architecture and technologies	18
3.4	CTI process to visualize data	22
3.5	Developed CTI timeline visualization.....	23
4.1	Feeds List.....	27
4.2	Collections List	31
4.3	Threat analysis	32
4.4	Pegasus analysis	33
4.5	Pegasus timeline	34
4.6	SolarWinds SunBurst trojan timeline.....	39

LIST OF TABLES

2.1	MITRE ATT&CK Enterprise Matrix	8
3.1	RedEcho attack timeline	15
3.2	List of imported Feeds	22
4.1	List techniques used by Pegasus Spyware for Android	35
4.2	List techniques used by SolarWinds SunBusrt Trojan.....	37
4.3	Continuation of list techniques used by SolarWinds SunBusrt Trojan	38

LIST OF ACRONYMS

General Acronyms

APT	Advanced Persistent Threat
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
CTI	Cyber Threat Intelligence
CyBOX	Cyber Observable eXpression
DNS	Domain Name System
HTTP	HyperText Transfer Protocol
IOCs	Indicators of Compromise
iOS	iPhone operating system
IoT	Internet of Things
JASON	Javascript Object Notation
PDF	Portable Document Format
SSL	Secure Sockets Layer
STIX	Structured Threat Information eXpression
TAXII	Trusted Automated eXchange of Intelligence Information
TCP	Transmission Control Protocol
TH	Threat Hunting
TI	Threat Intelligence
URL	Uniform Resource Locator
XML	eXtensible Markup Language

1 INTRODUCTION

Cyber-attacks have become more and more common with a considerable increase in the number of reported attacks, such as ransomware, phishing, social engineering, outdated systems exploit and others [1, 2]. Attackers have increasingly exploited weaknesses in large corporate systems, government agencies and individual vulnerabilities, so much that private and public institutions have been concerned about the lack of professionals to meet the increasingly urgent demand for cyber protection and Cyber Threat Intelligence (CTI).

Allied to this, data indicates that 62% of professionals in this correlated area do not receive the proper training from his companies to update themselves with the most current risks [3] - weakening cyber-attacks mitigation and vulnerabilities avoidance. Thus, the creation of methodologies capable of simplifying and adding intelligence to cyber-threat data is essential for the latent demands of the market, making the work of cyber analysts more efficient, consequently increasing the defenses of the organizations.

Cyber Threat Intelligence is ususally characterized by the common steps illustrated in Figure 1.1. CTI must explore collection, filtering, sharing and analysis of vulnera-bilities intel and threat data regardless of vendor, technology or source, and keep the cycle of using shared intel to create usefull anaylsis. For that, researchers are being developed to address the collection and filtering of threat intelligence [4, 5], as well as sharing and using that data to mitigate threats [6, 7].

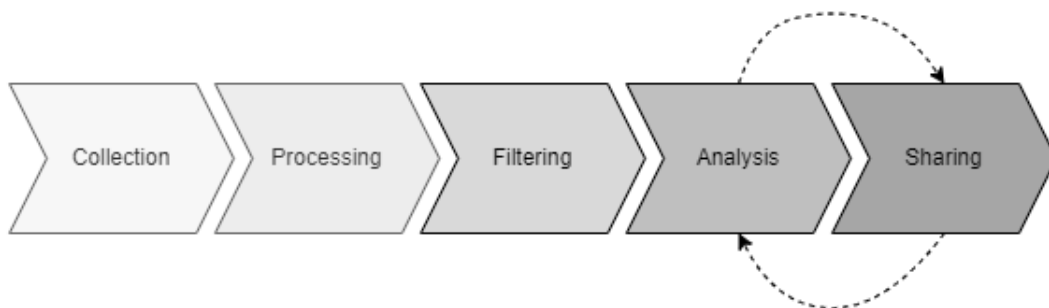


Figure 1.1: Common Cyber Threat Intelligence steps

According to [8], there are a few challenges of sharing Cyber Threat Intelligence data mainly because of the overwhelming number of threats, attacks and strategies that leads again to an profuse number of threat data as well in which a company needs to know and deal every single day. Thus, the challenges are:

- Verification: A threat actor or attacker may submit and create false reports to deceive or overwhelm threat intelligence systems, threat feeds, or a threat investigation.

- **Quality:** In case the focus of CTI is on collecting and sharing more threat data without processing it, there is a strong risk that much of it will be duplicated, wasting valuable time and effort of who will use that data in the future
- **Agility:** Security specialists want to share threat intelligence in near-real-time to follow attack speeds. The reason for this is that intelligence obtained too overdue might not be capable of helping on the prevention of an attack - however it is able to nonetheless assist recognize it better
- **Correlation:** The failure to become aware of relevant patterns, key statistics points and tendencies in threat data makes it tough to turn data into intelligence, which could and should be used to tell and direct security operations.

As said, cyber analysts need tools that help them on their daily work and on threat identification so that attackers does not stay undetected for days or even months. In this sense, a framework that properly addresses these points with the additional of an analytics interface is a desired target both academically and corporately, including issues such as threat hunting, threat intelligence, sharing protocols and data visualization.

Furthermore, it is desirable that these tools work in conformance to some common industry standards like MITRE ATT&CK and Cyber Kill Chain as example. Doing so, threats will be easily mapped and their information and footprints will be further shared through interested entities.

1.1 MOTIVATION AND OBJECTIVE

The motivation of this work is to create a framework that will address the challenges of gathering CTI data from multiple sources to create analytics capable of shortening threat mitigation time, as well as improving CTI in terms of collection, filtering, sharing, visualization and analysis with the proposal of a new 8-step model for CTI with analytics and temporal analysis.

The main objective of this work is to define a methodological CTI framework to improve threat identification and response time through the usage of an enterprise grade-level tool.

This work has its foundations based on the work of [9], and it will focus on visualizing and analyzing threat data, as well as a case study with real-world analysis example. For that, it will be explored these specific points on each step:

- **Visualization:** how to efficiently storage and correlate threat data in which it can be used to create viewable timelines to help the better understanding of how an attacker works and from how long he is working on that attack.
- **Analysis:** how to create useful analysis that can simplify and make easier to understand threat's patterns and indicators.

Also, to test and validate the proposed model, two examples of real cases will be explored in order to exemplify in a practical way our proposed methodological framework.

1.2 RESEARCH CONTRIBUTIONS

The main contributions of this work are:

- Extend the previous [9] framework for a new 8-step methodological framework to improve Cyber Threat Intelligence and threat mitigation;
- Functional implementation of the proposed 8-step framework and integration with CTI technologies.
- Validation of the proposed framework by comparison with industry-known standards and methodologies.
- Besides these contributions, this work also published the following journal article "Methodological Framework to Collect, Process, Analyze and Visualize Cyber Threat Intelligence Data"[10]

1.3 OUTLINE OF THIS WORK

Besides this Introduction, this master thesis is organized in 5 more Chapters as follows: Chapter 2 explores important background concepts that are crucial for the understanding of our research as well as the related work that was used on this work. Chapter 3 presents the results of this research, i.e., the 8-step framework and the proposed technical architecture with all the steps. Chapter 4 brings the demonstrations of the developed tool and the Proof of Concept of it with the analysis of two real threats. Finally, Chapter 5 presents the conclusions and future works.

2 BACKGROUND AND RELATED WORKS

The concept of Threat Intelligence (TI) says that it is knowledge acquired by analyzing evidence and data about possible threats from technology assets and systems. According to [11] “the set of data collected, evaluated and applied in relation to security threats, threat agents, exploits, malware, vulnerabilities and indicators of compromise” showing that all information which assists in making decisions against threats can be considered results of TI generated knowledge.

This shows how intelligent analytics is important to any organization, but in addition to the intrinsic intelligence generated from threat data, it is desirable to achieve real-time analytics in such a way that any type of attack is mitigated as soon as possible and no cyber or physical damage occurs.

2.1 CYBER THREAT INTELLIGENCE

The concept of Cyber Threat Intelligence is close to the Threat Intelligence concept: it is the knowledge generated from various sources of data and information of the cybernetic field. As defined in [12], CTI “emerged in order to help security practitioners in recognizing the indicators of cyber-attacks, extracting information about the attack methods, and consequently responding to the attack accurately and in a timely manner”. Therefore, a CTI professional must be able to deal with data extraction, data filtration, data manipulation and data standardization to achieve a very important goal: to have data intelligence-generation and visualization in (desirable) real-time speed.

Another important CTI goal is that stakeholders should share their data with each other, promoting - according to [13] - “situation awareness among stakeholders through sharing information about the newest threats and vulnerabilities, and to swiftly implement the remedies” where one should not suffer with the same threat or vulnerability that other has already suffered in the past. This objective helps to avoid rework, since one already discovered threat will be shared with all stakeholders, increasing the defense of systems against threats. Figure 2.1 shows all the steps and goals that Cyber Threat Intelligence should achieve as proposed in [14].

Companies can start and expand their "Intelligence Sharing" by choosing the right tools to share threat intelligence. For an example, emails are the most accessible place to start, but it is desirable to focus on moving to more formal exchange methods using the tools available, leveraging standards like STIX [15] and TAXII [16] to do so. Information Sharing and Analysis Center and other industry organizations are perfect communities to start with intelligence sharing and they typically have mechanisms in place for doing so.

Sharing observed adversary behaviors or details from incident response are good practices for CTI. Looking for opportunities to share with organizations outside the companies' vertical

that includes localized entities such as fusion centers and other organizations deemed a good fit for sharing intelligence too. Like always, the more organized stakeholders are individually and between each other, easier will be to mitigate threats and attacks.

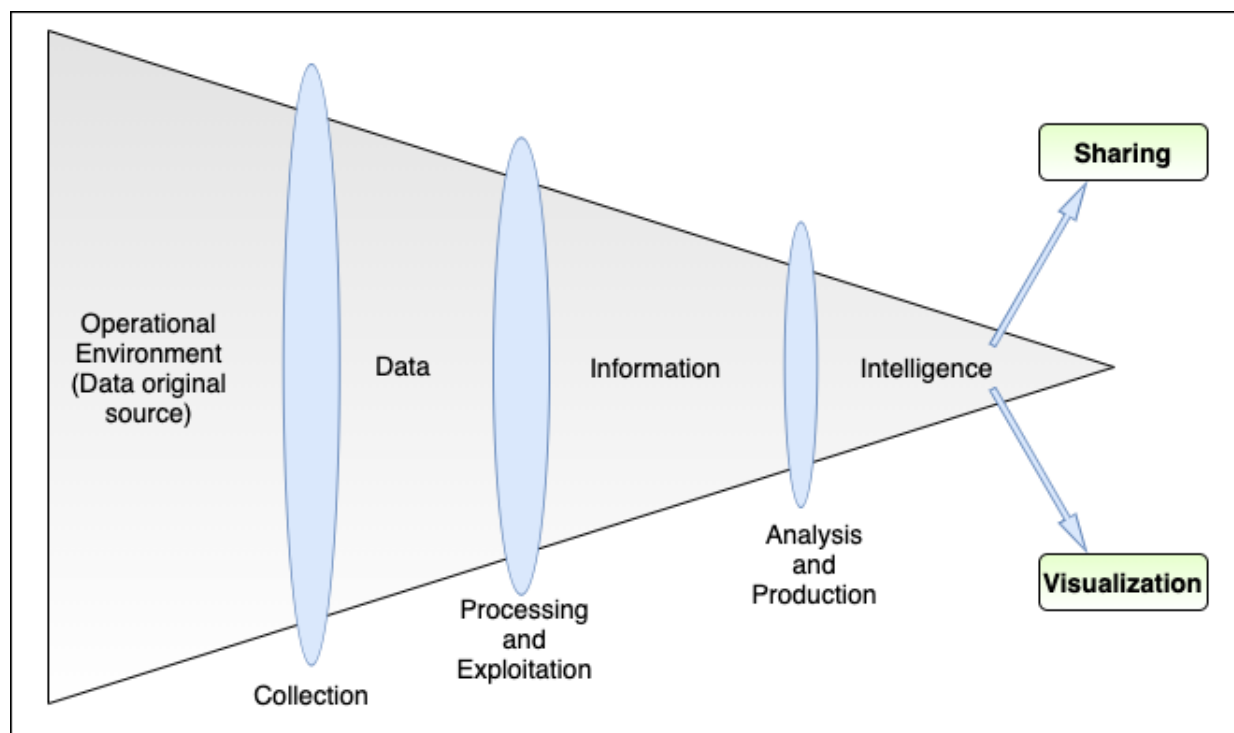


Figure 2.1: Desirable steps and objectives for CTI (adapted from [14])

This work proposes a framework for Cyber Threat Intelligence that consists of acquiring, treating and visualizing Indicators of Compromise (IOCs) and any other relevant data to identify and mitigate cyber-threats.

2.2 THREAT HUNTING

Threat Hunting (TH) is the practice of proactively searching for cyber threats that are lurking undetected in a network and it is strongly linked to the concept of Advanced Persistent Threats (APT) because when Indicators of Compromise of a particular APTs are detected within organizational and network equipment, security analysts must search and analyze logs from hosts, routers, servers and any other network-connected hardware or software to find if any of the IOCs entities appears on their logs - showing if the threat has penetrated the network or any hardware/-software. This is primarily the definition of Threat Hunting - finding and discovering threats from IOCs and any other indicators that correlate tracks and footprints to a vulnerability or attacker.

Usually, Threat Hunting represents a big challenge for those who need to deal with it. It has four main challenges [17], [18]:

- First challenge is about searching at scale, i.e., because of the attack pattern of attackers, it is

necessary to link related IOCs together even if the attacker footprints exist in a long period of time – like weeks or months. Hence, TH must be able to deal with long-time-scattered data and still be able to identify threats and vulnerabilities along its systems.

- Second challenge is about identification and correlation of threat entities, i.e., the ability to look the entire scenario of a threat campaign to fully identify all the steps of the attacker, avoiding being cheated by the attacker if he tries to masquerade his footprints.
- Third challenge is about the confidence score of threat generated data. These data should have as few false positives as possible so that cyber response operations can be done smoothly and without rework.
- Fourth and last TH challenge is how to get a complete picture of a threat when analyzing only fragmented pieces of data like IP addresses, hashes, domain names, process names or any other useful information. Weak Threat Hunting data will create a weak analysis, as well as a poor response to any threat that may benefit from that specific vulnerability.

When all these TH challenges are overcome, an organization will have a large set of data – structured, semi-structured and natural language data – composed by text files, images, documents and any other possible kind of data. This data will be ingested and processed by a CTI tool or a security professional to deliver intelligence and finally to work on threats and vulnerabilities, using frameworks [19], just as shown in Figure 2.2.



Figure 2.2: Example of Threat Hunting framework [19]

2.3 CTI STANDARDS AND FRAMEWORKS

It is important to highlight that the standards are used to make the sharing of threat information more organized, agile and possible. With the large amount of data acquired with Threat Hunting and with Cyber Threat Intelligence, having patterns of entities, objects, and their description to identify threats, actors and targets are extremely valuable for a Threat Intelligence ecosystem.

Threat frameworks allow security analysts to streamline investigations and understand the huge number of alerts streams that originate since from security logs to intelligence sources. They help to reconstruct the footprints of the attacker or the stages and steps of an attack using visualizations about it, how an attack may progress, what steps a given threat actor will take, and what countermeasures are possible. Many different types of frameworks are used on CTI, each serving different purposes. Some of the frameworks are designed to help organize, implement and manage a cybersecurity architecture and others focus on one specific area of interest.

There are three well known Cyber Threat Intelligence frameworks that are used world wide:

The first well-known CTI framework is Lockheed Martin's Cyber Kill Chain, which is part of the Intelligence Driven Defense [20] model to identify and prevent cyber attacks. This model identifies what an adversary must do to achieve their goals and provides insight into the activities an attacker might undertake. The second one is the Diamond Model [21]. This CTI framework highlights four key aspects of an intrusion: the adversary (who), the infrastructure (what), the capabilities (how), and the victims (where). For this model, an attack event is defined as the way in which the attacker demonstrates and uses specific skills and techniques on infrastructures against a specific target. At last but not least, MITRE ATT&CK [22] stands as one of the most important and used cyber framework around the world. Its importance is so significant that it will be further explored in the next section.

2.3.1 MITRE ATT&CK

According to [22], MITRE ATT&CK is largely a knowledge base of adversarial techniques — a breakdown and classification of offensively oriented actions that can be used against particular platforms, such as Windows. Unlike prior work in this area, the focus it is not on the tools and malware that adversaries use but on how they interact with systems during an operation."

It is, therefore, an important tool to deal with threats and their mitigation, as it allows the sharing of information in a structured and standardized way, being widely used in various Threat Intelligence tools available on the market [23] [24]. MITRE ATT&CK Enterprise Matrix contains 14 Windows, macOS, Linux, Azure AD, Office 365, Google Workspace, SaaS, IaaS, Network and Containers tactics representing an adversary's tactical objective for acting.

MITRE ATT&CK Enterprise Matrix 14 tactics, representing possible attack pattern in which an attacker can use to infiltrate a system and/or network, are described on Table 2.1:

Table 2.1: MITRE ATT&CK Enterprise Matrix

Tactics	Description
Reconnaissance	This tactic consists of techniques that involve opponents in the active or passive collection of information that can be used to support targeting attacks. This data can include details of the victim's organization, infrastructure or even personal information. Such data is commonly used by the attacker in other MITRE Enterprise Matrix Tactics such as <i>Initial Access</i> for example.
Resource Development	This tactic consists of enemies building, acquiring, or compromising/stealing resources that can be utilized to facilitate targeting.
Initial Access	This tactic consists of adversary trying to get into the network.
Execution	After gaining the Initial Access to a server or computer, attackers will be able to execute malicious code to exploit the system or wait for an user interaction to gain access to the network.
Persistence	If the hacker gain access to a system, he might lose his access if a user changes his password or any access key. To avoid this, an attacker can install spyware software to keep control of user-access privileges.
Privilege Escalation	Adversaries can enter an enterprise system using an unprivileged access but they may acquire more resources within the victim system and elevate their permissions to better exploit vulnerabilities in applications and servers within the enterprise system.
Defense Evasion	Adversaries frequently cover their traces to escape detection and overcome security restrictions, allowing them to continue their malicious activities.
Credential Access	Adversaries may acquire more usernames and passwords from a compromised computer's Bash History or Keychain in order to achieve more malicious objectives and maintain access to the victim system.
Discovery	Adversaries may attempt to explore and gather more information about an enterprise system after gaining access to it in order to support their objectives. These attempts include the discovery of possible vulnerabilities to exploit, data stored in the system, and network resources.
Lateral Movement	Adversaries may switch from a compromised user account to another user account within an office area to keep control on that network.
Collection	Attackers can gather information that aids them in achieving their malicious objectives.
Command and Control	Adversaries use this technique to remotely control their operations within an enterprise system. When attackers gain control of an organization, their compromised computers may be used as botnets to increase the attack depth.
Exfiltration	After data are collected, adversaries may package it compressing data to minimize the data size transferred over the network, making the exfiltration less conspicuous to bypass detection.
Impact	Adversaries can breach the confidentiality, degrade the integrity, and limit the availability of assets within an enterprise system after achieving their objectives to make it unavailable for the company.

2.3.2 CybOX

Cyber Observable eXpression (CybOX) provides a common structure to specify, characterize and share information about cyber observables, i.e., objects like network connection, an IP address, URL and others in a standardized way [25]. Figure 2.3 shows an example of "CybOX tree" to characterize use cases.

CybOX is not targeted at a single cyber security use case but rather is designed to be flexible enough to offer a common solution for all cyber security use cases requiring the ability to deal with cyber observable. It is also designed to be flexible enough to allow both high-fidelity description of IOCs that have been measured in an specific context, as well as more abstract patterns for potential IOCs that may be threat targets in order to observe and analyse them.

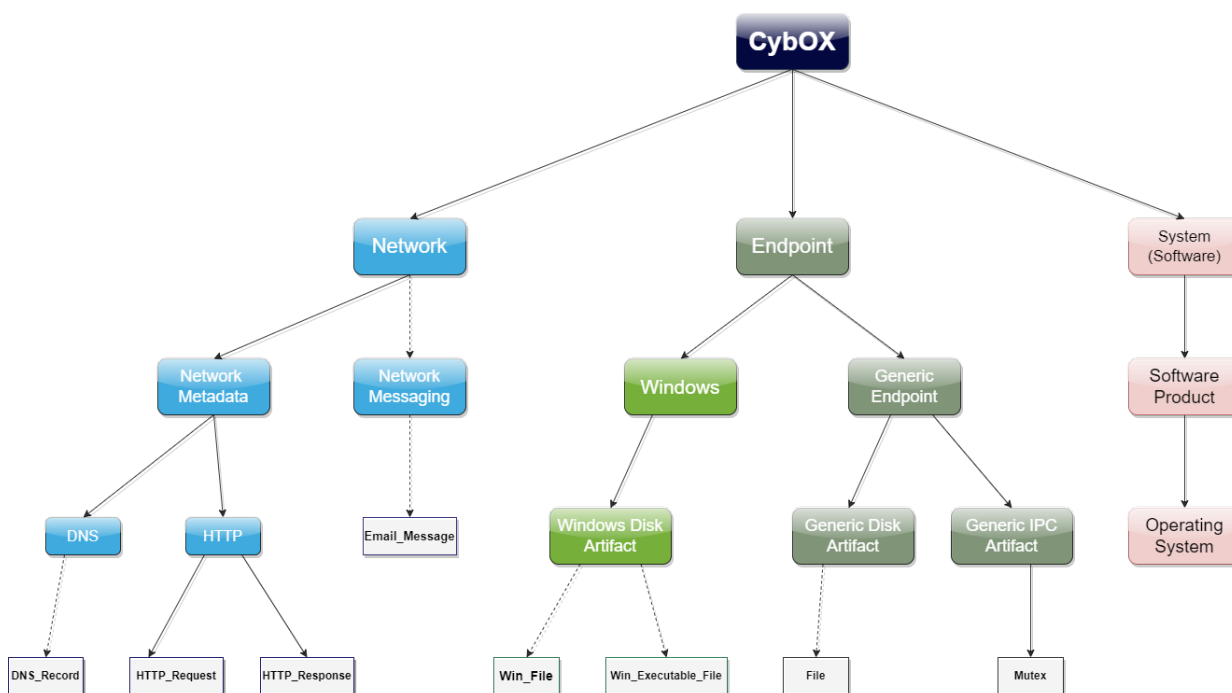


Figure 2.3: CybOX core objects categories (Adapted from [26])

Furthermore, threat assessment, log management, malware characterization, indicator sharing, and incident response can be all patronized using CybOX structure and objects. CybOX is now part of STIXv2 as Cyber Observable Objects.

2.3.3 STIX

Structured Threat Information eXpression (STIX) [15] is a language developed by MITRE to share TI information within stakeholders of an ecosystem. It is used to acquire, classify and share any data related to an attack campaign, providing tools and patterns to represent information in such a way that CTI data can be efficiently used by security professionals inside Cyber Threat Intelligence Tools.

It allows users to standardize unstructured data and individual IOCs inside a machine-readable

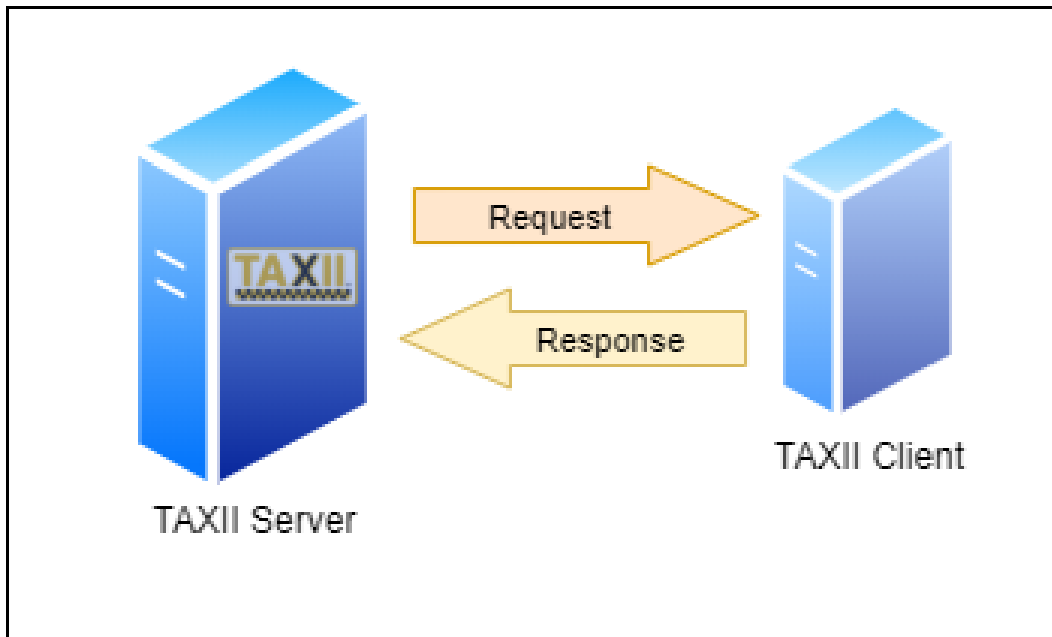


Figure 2.4: Client-server TAXII model

language so automation and interoperability can be applied. The first version of STIX (STIXv1) was XML-based and had eight cores. The second and current version of STIX (STIXv2) is a JSON-based language [27] with various architectural improvements to get close to a CTI environment.

2.3.4 TAXII

MITRE also developed TAXII (Trusted Automated eXchange of Intelligence Information), an application layer protocol that specifies a collection of services for businesses to exchange Threat Intelligence information data through a Client-Server model as it is shown in Figure 2.4. TAXII was designed specifically for the transmission of data in the STIX format, but it is not limited to that as the transport protocol used by it is Hyper Text Transfer Protocol Secure (HTTPS), offering a variety of sharing models like peer-to-peer, spoke-hub distribution and source-subscriber [16].

2.4 RELATED WORKS

Cyber Threat Intelligence is a theme extremely important for the private business area as well as in the academic area. Thus, there are many academic research being developed addressing collection, sharing, standardization and usage questions of CTI.

2.4.1 CTI protocols and standards

In [28], authors give us key context of CTI protocols and how important it is to use common standards for sharing CTI data, thus given a comprehensive evaluation methodology for Threat

Intelligence standards and how to evaluate Cyber Threat Intelligence platforms and market tools. Authors in [29] presented a survey about Threat Intelligence and open-source threat intelligence tools, showing a comparison that can be used to understand not only open source CTI tools but enterprise CTI softwares. The main points found on this research are the secure and compliance-accepted sharing of information, in addition with the possibility of common protocols like STIX and TAXII.

Another important research that compared available formats and languages to share threat intelligence data like STIX, TAXII and CybOX is presented in [30]. The authors shows that, even though sometimes STIX is poorly implemented on CTI tools, it is one of the most popular CTI standards used today. In [31], authors shows that CTI needs standardization in order to achieve an intelligent information sharing and desirable quality, doing some considerations about STIX language and how it can be used to give a holistic approach of cyber incidents.

2.4.2 CTI main challenges and technical improvements

The work in [13] presents a survey addressing possible challenges of CTI, e.g., how to pro-actively mitigate cyber-attacks using process automation on CTI sharing and how to trust on a stakeholder that have shared threat IOCs data - similar to what was researched on [17], the authors enumerated three CTI challenges, where two of them addresses the data quantity and quality, and one addresses the sharing problem between platforms.

In [7], authors shows how small and medium size enterprises suffers from the lack of CTI tools and shared information. They created a CTI Feed that can provide these companies with actionable recommendations ordered by relevance and help them avoid vulnerabilities and threats.

In the context of intelligent CTI data extraction, [4] proposes an automatic approach to generate CTI data from open-source threat intelligence publishing platforms (i.e., any platform that shares IOCs or any other useful information about threats) using Machine Learning and Natural Language Processing together with known threat intelligence background in order to achieve accurate and detailed CTI data that can, for example, feed a tool like ours in order to help cyber security analysts on threat mitigation. In Preuveneers & Joosen [6], similar problematic is considered by authors when they propose a solution to complement the sharing of IOCs using Machine Learning based threat detection and demonstrate their proposed solution implementing it on state-of-practice open source CTI sharing and incident response platforms.

In [32], authors proposed a CTI model based on heterogeneous information network, which helps to integrate various types of cyber-threats nodes. The authors came out with a threat type identification system that significantly improved the performance of state-of-the-art's baseline, confirming the main challenge in which this work seeks to address: a CTI model to shorten threat answer time with user-useful tools.

In [33], the authors propose a paradigm shift of cybersecurity information exchange using a blockchain-based model in which every stakeholder will have the responsibility to process and validate all shared data within the network. They developed an Ethereum Blockchain Smart

Contract Marketplace and tested some scenarios to prove that their developed system is valid and a possibility to future CTI platforms.

In the same line of reasoning, [34] touches the point of how can we share CTI if all stakeholders have their own systems and their own methodologies to collect and use CTI data. In this problematic, authors suggest the use of blockchain to allow the world-wide sharing of CTI data. The work embraces its arguments through some attacks examples and how a blockchain-based system could help with the standardization of collected data.

Authors in [35] propose again a blockchain model to help improve security of CTI in the context of IIoT, that is Industrial Internet of Things, i.e., IoT devices for large companies focus. Using blockchain, their model allows integrity auditing of threat intelligence using ciphertext state to ensure the confidentiality protection requirements of threat intelligence data, as well as a double chain model to store ciphertext in one and the other to audit stored data inside the first chain.

Another interesting analysis was made in [36], where authors proposed a psychological analysis of threat actors in order to better understand and mitigate their actions. In this work it is used both technical knowledge about Threat Intelligence and psychology knowledge to define the attacker personality just by looking at the logs he leaves behind when exploiting some system or network. By that, the necessary cyber security precautions can be taken in time even if this hacker has never participated in any survey or test or has never appeared on that particular server before. For further improvement, the authors placed that their psy-cyber system will be integrated with SIEM software to real-time review intrusions and detect attackers' patterns.

Therefore, based on the context of all these researches, it is possible to see how CTI is urging for new methodologies and tools in which helps achieve better sharing, trusting and threat identification as possible - and how organizations are thrilled to have more research results inside this area.

Thus, we used their conclusions as a bias for our decisions on which technologies should be used on the implementation of our tool such as CTI standards and programming languages, as well as ideas for our 8-step methodological framework.

3 PROPOSED SOLUTION AND MODEL

In this chapter, the methodology behind our work will be explored and detailed. Also, it will be introduced the early CTI framework developed in [9] as well as our novel 8-step model that will be technically explained step by step within its architecture. It is important to emphasize that even though the name "steps" suggests something sequential, one step does not necessarily have to come before the other and vice versa. Also, the way we propose to collect threat data from websites is an example of how it can be done and will be explained in the end of this chapter with all its code in detail.

3.1 METHODOLOGY

The methodological approach used in the research and development of our 8-step model consisted of carrying out bibliographic research seeking references that describe the use, applicability, functioning and behavior of methods and tools of Cyber Threat Intelligence, as well as looking for industry-known standards to better understand which one could have a best fit with our model.

We researched about other famous models like MITRE ATT&CK, Diamond Model, CybOX, MISP, etc., before we started to develop our 8-step model, and MITRE's model was the one that we considered most on our work since it is more complete and widespread by CTI community. For that, some academic research engines were used, like Google Scholar and Springer. The following keywords were used to find relevant results: "Threat Intelligence OR Cyber Threat Intelligenc" AND "CTI standards OR CTI procotols" AND "CTI tools".

For acknowledgment, the development of our model and proof-of-concept tool were based on the following steps:

- Bibliographic research of CTI tools, standards and related works;
- Usage of a previously develop tool that had only 6 steps of our desired 8 steps model;
- Using all this previous knowledge, propose a more complete model of Cyber Threat Intelligence;
- Selection of two real-case examples of threats to insert in the tool and to test and exemplify our 8-step model;
- Testing of Analysis step and Visualization step.

3.2 CONTEXTUALIZATION

Cyber Threat Intelligence has become increasingly present in corporate security systems as it has great importance in mitigating threats and consequently reducing damage from attacks. According to 1.098 IT and IT security practitioners in North America and the United Kingdom [37], the biggest challenges faced by Threat Hunters in the search and collection of useful data for a Threat Intelligence analysis are shown in figure 3.1:

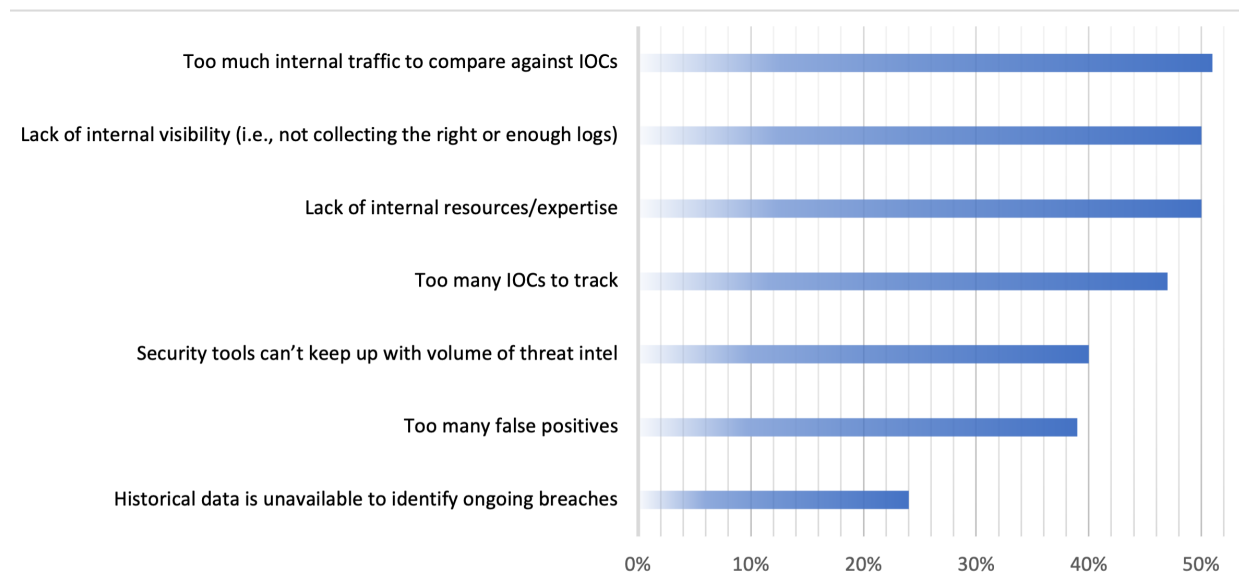


Figure 3.1: Biggest challenges faced by Threat Hunter teams (Adapted from [37])

This illustrates how CTI is inherently dependent on issues such as the collection of large amounts of unstructured data, lots of IOCs and the quality of collected data itself.

Another big challenge for Cyber Threat Intelligence is acting fast to understand, neutralize the threat and share it info with other stakeholders. Consequently, real-time tools are needed [12] [30], as well as user-friendly interfaces to simplify the security analysts analyses and the reaction to threats.

An example of the problematic behind CTI is the attack named RedEcho that started in 2019 and targeted the Indian Power Sector [38] to exploit Indian servers networks, spoofing over SSL via TCP port 443, HTTP over TCP 8080 and DNS over UDP 53. All this to ex-filtrate critical data. As can be seen in Table 3.1, from the first trace of the attack until the counter-attack measures were taken, it took almost 2 years. A lot of research, network traffic analysis, and expert analysis was required to achieve correlation between data and attack pattern, but even so, it took a long time to map and understand the dynamics of the attack, allowing it to penetrate the systems of Indian energy companies and devastating their targets.

This is a long time to mitigate a threat and shows the value of a tool that can reunite different threat Feeds and IOCs to show all their timestamped footprints to neutralize or avoid it as soon as possible. Thus, the RedEcho attack example given above illustrates the problem addressed in

Table 3.1: 2019 RedEcho attack timeline [38]

Timeline	Event description
Apr 2, 2019	The first attacker domain www.smartdevoe.com was registered
Apr 26, 2019 to Jul 11, 2020	Seven other domains were registered
Sep 22, 2020	First IP Address (218.255.77.52) was detected as part of the attack common pattern
Dec 30, 2020	Potential Data Ex-filtration Observed
Feb, 2021	Recorded Future released a domain blacklist and security advice to countermeasure this attack

our work: a lot of unstructured data from different sources which is difficult to unify into a single CTI database to create a global understanding of the threat. This emphasizes the importance of linking threats footprints and IOCs to more quickly mitigate them - avoiding resources loss and gaining advantage in the threats countermeasures.

The work presented in [9] proposed a six-step CTI model and a proprietary CTI tool. All functionalities related to the context of threat intelligence and each of the six proposed steps were described, as well as the CTI tool implementation details and interconnections. The overall result of that work was a tool that allows threat information collection, preparation and sharing.

Thus, based on the model proposed at [9] and the state-of-the-art of the CTI works compiled about collection, sharing, visualizing and analyzing threat data, this work proposes an eight-step CTI model as shown in Figure 3.2.

The two additional steps proposed are Visualization and Analysis. These two steps as well as all the other 6 steps are intrinsically related since one does not exist without the other and vice versa. Thus, there is no hierarchy between steps, i.e., they are all important for the correct application of our methodological framework and are describe as follows:

- Step 1 (Management) is responsible for managing the interaction of users with each functionality that the application offers and the interaction between them. It is also responsible for controlling the flow of data and its access, through the management of access permissions. This functionality must be linked to all actions that can be performed by users, such as creating requests to import data, manipulate stored data, share feeds and collections. That's why this step is at the base of our model.
- Step 2 (Indexer) must be supported by a storage structure capable of supporting the amount of data input and output on our tool. Relational databases are the main storage structures used on our model to support the ingestion of collected data and the feeding of upper steps.
- Step 3 (Collect) is responsible to provide external data gathering and insertion on our model. It needs to collect data in different formats and standardize them on the same format required on Step 2.

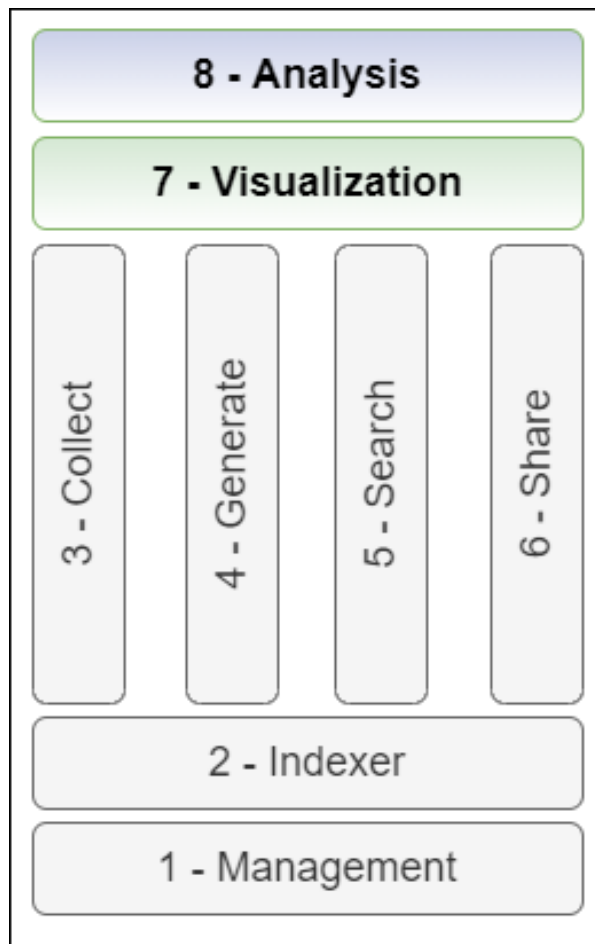


Figure 3.2: CTI 8-step proposed model

- 4 (Generate) must provide the normalization of internal data using the same Step 2 and 3 pattern so that ingested data can be converted into Feeds for later use in our model.
- Step 5 (Search) must provide mechanisms and methods to allow data manipulation and exploration in an effective way, permitting fast data indexing and full visibility of stored data.
- Step 6 (Share) must allow the sharing of internal data such as Feeds, Collections and Threat Campaigns between users, as well as the sharing with third parties tools that uses STIX language.
- The Visualization step (Step 7) introduces the visualization of the CTI data in a temporal format, to create a threat timeline so one can have a “complete picture” view of the threat footprints, IOCs and any useful information that was enriched and shared by other interested parties.
- The Analysis step (Step 8), which is intrinsic to the Step 7, implements functionalities allowing to analyze data, manipulate and get the best information from the available threat data.

3.3 SOLUTION OVERVIEW

In this section, we will present our 8-step model that has its foundations within the work of [9]. For better understanding, originally this tool was a Python-based tool that mainly collected, organized and aggregated threat data (IOCs) to show to the user in an efficient and fast way in order to help with threat mitigation.

Our novel 8-step model is based on this tool and aggregated 2 more steps in order to improve the end-user (i.e., the cyber-security analyst) experience in the sense of making his/her life easier to understand threat patterns and how they correlate with each other.

Thus, the main idea is to deliver a "user friendly" interface to deal with all the data and information that can be collected from threats inside the Internet. So, Figure 3.3 describes the technologies and products used to implement the proposed 8-step model. All steps are explained in detail in the next subsections with all technical solutions behind and the functionality that commit within this methodological framework.

The decision points for joining the 8 steps must be taken by the developer in such a way to facilitate their development and integration, desirably taking into account these points:

- High-performance indexable database so it is possible to search stored data quickly
- Robust relational database to store all kind of data (cold or hot data) in order to keep track of all information that was/is stored in the tool.
- Controlling tools to organize the flow of information through the steps of the methodological framework.
- Modern programming languages so it is possible to collect and extract IOCs and threat data across different internet websites.

3.3.1 Management step

The Management step is responsible for controlling the application and its functionality. Its attributions include the control of the users that access the application, the control aimed at the manipulation of the offered functionalities and the management of the interaction between these functionalities.

It relates itself to all other steps, supporting the control input data next to the Collection step and the Generation step. It also helps the management of data stored next to the Indexer step and works with Search step generating access control to data. Furthermore, it helps achieve user access to determined functions of sharing and indexing, something similar to an Authentication Directory with users permissions.

The implementation of this step must support permissions management. This functionality must be linked to all actions that can be performed by users, such as creating requests to collect data, choosing data storage location, manipulation of stored data, sharing data, among others. The

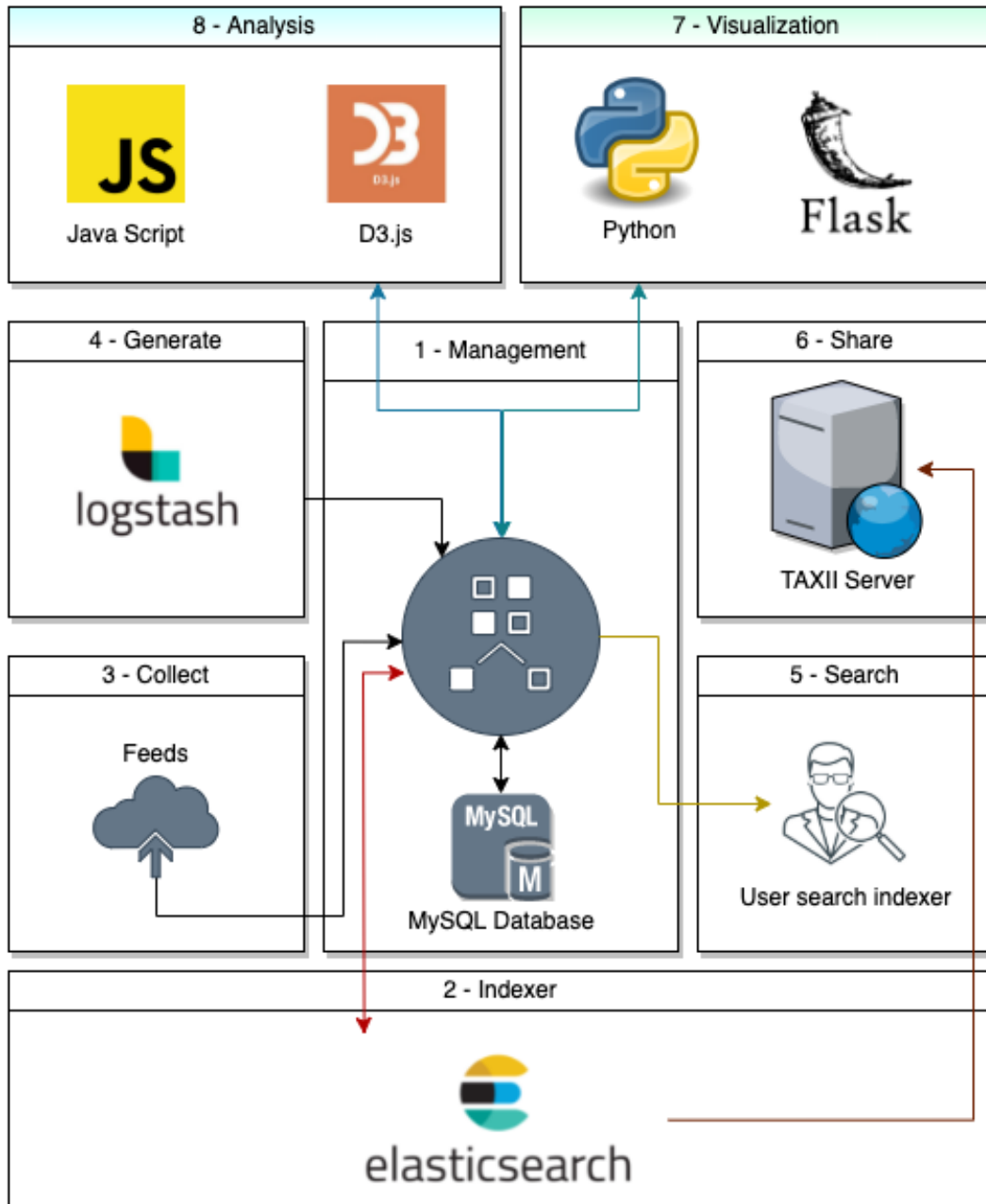


Figure 3.3: Tool architecture and technologies

database in which data will be stored should be exclusive to this step so that the inter-dependency of resources are free.

3.3.2 Indexer step

The Indexer step must work together with the Management step, and perform control of the CTI data that enters, leaves and remains in the application. To bear the data storage functionality, the application must have a storage structure capable of sustaining the desired amount of data input and output for good application operation. The main storage structures today are banks of relational data, non-relational databases and custom structures used on proprietary software. The structure implemented to support the Indexer step must be able to store data sent by the Collect and Generate steps, and make data available in the required by the Search and Share steps. The best types of databases for CTI tools are non-relational databases. The Indexer step is commonly implemented by all applications, but the availability of its resources to end users is bequeathed only by applications that provide the possibility of inserting data (Collect and Generate step).

3.3.3 Collect step

Collect step encompasses the ability to collect data from sources outside the infrastructure that supports the application of CTI. It needs to normalize them (when necessary) and send them to the Indexer step to be internalized. The Collect step must work together with the Management step to better control the feeds of interest, along with the consistency of collection. Applications that implement the Collect step must offer features that use mechanisms aimed at data collection, which are made available in various structures and standards, such as wikis, security communities and TAXII standard servers.

They must also offer methods that standardize this data in a single format, being mandatory compatibility with the format required by the structure implemented by the storage, such as the STIX standard. These functionalities proposed in the Collect step are vital for the use of an application of CTI, as it allows contextualizing the information collected with the assets of the infrastructure to be monitored.

The way in which the collection should be carried out in order to take better advantage of data must be a conjecture between the infrastructure and the collection techniques chosen by the developer of the application, such as the use of robots, feed collection in different patterns, etc. Most tools add the proposed features in the Collect step into its list, but there are few that allow the end user to manipulate it, having the ability to enter feeds that are of particular interest to the analyst.

3.3.4 Generate step

The Generate step includes resources that are needed for data collection generated by assets that belong to the internal infrastructure of organizations that make use of the CTI tools. This step

must work together with the Management step to control the assets in the best way possible to send data to the application, which improves with the frequency of its receipt. Applications that implement features of the Generate step must have mechanisms the structure of the received data to standardize the data, unifying the data in standardize format with the standard required by the compatible storage capacity.

It aims to provide value generation for the organization that implements it, using existing computing infrastructure assets such as IOCs feeds. This step uses Logstash to control data ingestion and data destination inside our tool. With that, it is created a pipeline to standardize feed ingestion and its correct storage inside our databases.

3.3.5 Search step

Search step is where the user can search for indexed data. It uses all the technology embedded under it to allow fast and efficient search from the perspective of the user.

This step must offer mechanisms which are able to read the data in the patterns provided by the Indexer step, to generate adequate visibility of this data, in a understandable pattern to be used by analysts,, according to the need for visualization within a scope chosen by the manufacturer/-developer. The visibility referred to is not only graphically based, but also in detailing the data in tabular form, self-management feedbacks and alerts.

3.3.6 Share step

Sharing step has the objective of grouping functionalities that aims the sharing of data that comes from outside the tool framework. The integration between Management step is fundamental to this step because the creation of business rules is essential for the correct management of the tool. The developer must pay attention to this step because of its importance to CTI. Sharing is the most crucial part of Cyber Threat Intelligence since we all live on a globalized world and attackers often spread their attacks through several countries and networks, following the principle "if you waver, it's over". So, like humanity evolved through shared knowledge, the same analogy works here. It is mandatory to share information about an specific attack, threat or attacker, so everyone can avoid the same situation you have already passed by. Moreover, sharing IOCs and threat info allows stakeholders to perceive that a particular threat is spreading more strongly than another, for example - making threat's importance detection more efficient and intelligent.

Alongside this, share step uses TAXII protocol to achieve information sharing. It is used a TAXII Server to allow standardized STIX sharing making our tool integrable with any other market tool.

3.3.7 Visualization step

Visualization step is built to run on any modern system that uses web-browsers, allowing an "all time"availability. Its importance to this methodological frameworks drives from the ability to

see threat data and IOCs in a human-understandable way, in which and cyber analyst can better and faster understand and detect patterns on data. It works really close to Analysis step in a way that one depends on another so analysis can have a complete tool to do their daily work. Just like information was extremely important in old war times, visualizing the attackers footprints through a temporal way drives a more intuitive understanding by the analyst in a way he can remember when something or some action was taken by the attacker or threat. Even if he/she can't remember something, it is just needed to look the past of the timeline to visualize threats data - if there is data, of course. So, this step is developed using Python and Flask, a micro-framework for web developing that allows efficient and robust applications.

3.3.8 Analysis step

Finally, Analysis step is built on top of all architecture because it is used to create analysis of ingested threat data and IOCs, as well as data created originated from Visualization step insights. This step is very important to all methodological framework because it is where data is enriched and cyber analysts can use all their knowledge and intelligence to create useful insights to fight against attackers and mitigate their tactics and techniques.

For that, this step is developed using Java Script language because of the *D3.js* library that allow the work with large data-sets of Feeds and the creation of dynamic visualizations of IOCs.

3.4 FEEDS COLLECTION

Steps 7 and 8 must be fed with treated and standardized data in order to friendly provide to the user threat information and analytics. However, before that, it is necessary to correctly collect IOCs from different Internet sources - like forums, sites, archives (PDF, word, freetext, etc.) and Feeds. Among these, Feeds usually are the most desirable information collection source, because many cyber threat vendors or specialists release their threat research and hunting data in this way. Feeds are a set of indicators of a specific threat type, e.g., malicious IP addresses that should be blacklisted on firewalls or phishing URLs that must not be accessed by common users. In this work, seven known Feeds were used to acquire threat information and IOCs. All this information is listed in Table 3.2.

Furthermore, it is possible to add manual Feeds or IOCs on the tool as long as the data is written in the STIX/TAXII format.

3.4.1 Visualization Step

As mentioned above, for a correct execution of the proposed Visualization step, it is necessary a very well determined CTI process of collection and data organization, which will allow the correct IOCs and any other information linkage as illustrated in Figure 3.4. This process illustrates the number of data silos where useful information can be extracted - such as PDFs, blogs, tweets,

Table 3.2: List of imported Feeds

Feed name	Brief description
Blocklist.de [39]	Lists of malicious IPs to be blacklisted
Firehol Blacklisted IPs [40]	Blacklist of malicious IPs that should be blocked on servers directly connected to the Internet.
Mirai Security [41]	List of the last 1000 likely IPs of machines infected with Mirai Botnet ransomware.
Openphish [42]	List of phishing URLs
Pan-unit42 [43]	List of malicious masked URLs
Vxvault [44]	List of malicious and downloadable <i>.dll</i> and <i>.exe</i> files.
Zerodot1 [45]	List of blockchain mining bot domains to be blocked by a network admin.

websites, text files, etc. - which often contain IOCs and threat footprints that can be useful in the CTI analysis.

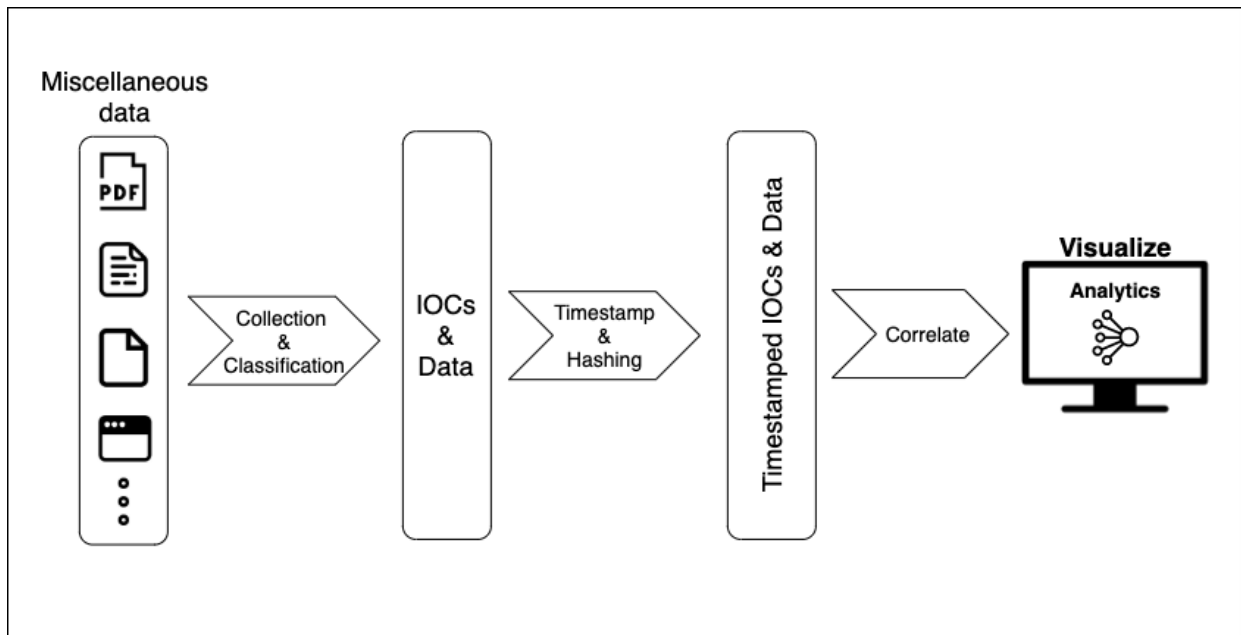


Figure 3.4: CTI process to visualize data

The first important consideration is how to efficiently ingest all that data in a way it can be further used on the CTI process. Therefore, when collecting a piece of information from any source, a timestamp must be created to uniquely identify that data for further validation, indexing and the exhibition of it in the timeline itself.

Another important point is to create a hash signature for that data, so it is possible to avoid collecting an already existent data in the CTI tool repository, thus duplicating it. When these two points are implemented, the temporal visualization is possible, providing the user with a useful interface that will enhance Cyber Threat Intelligence results.

3.4.2 Analysis Step

With Step 7 (Visualization), the tool will allow the exhibition of analytics from the CTI acquired data - like IOCs relationship, quantity, timestamps from when the IOCs was added to the platform or discovered by Threat Hunting and any other possible analytic idea.

Because of this, the Analysis step (Step 8) has challenges similar to the business intelligence area, i.e., it will depend upon the cybersecurity engineer to create analytics that make sense for the business situation, or the scope of the challenge being addressed at that specific time - choosing which analytics are best suitable.

3.5 TIMELINE VISUALIZATION

Analyzing IOCs and threat information is usually a complex task because it is needed to analyze a large amount of text: IPs, URLs, domains, IPs geolocation and any other textual information. The solution proposed in this work helps this task, providing a timeline-model in which the user will literally look at threat data in a temporal format, facilitating the understanding and perception of patterns such as the origin of the threat, interconnected domains and IPs, and even possible attack groups. Figure 3.5 shows an example of the proposed timeline with generic IOCs added to illustrate.

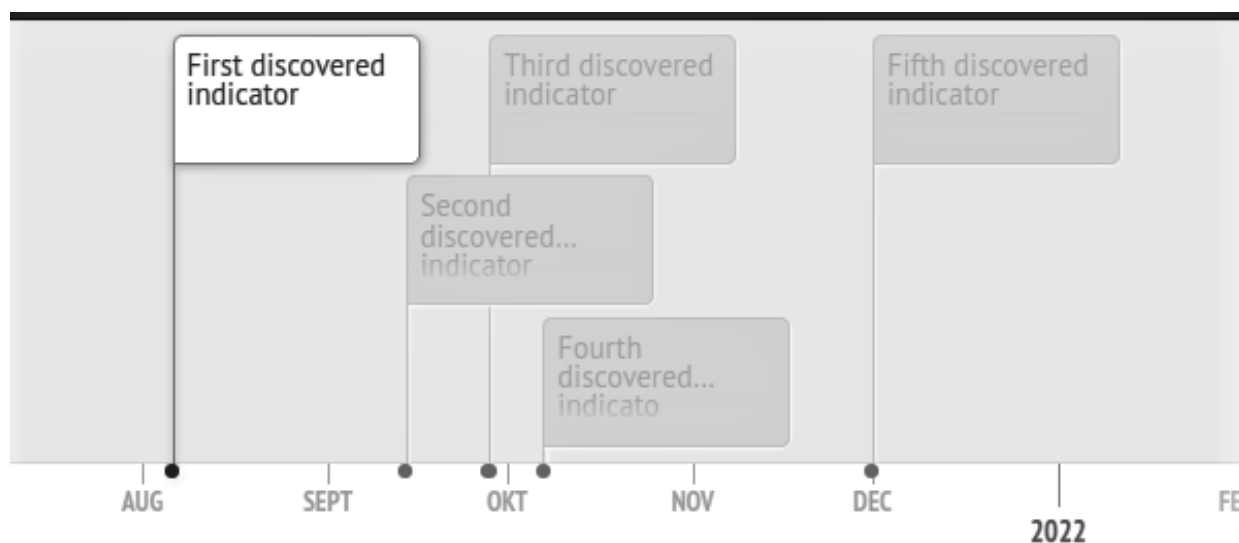


Figure 3.5: Developed CTI timeline visualization

The CTI timeline is customizable and need only to be fed with events. Users can add images to characterize individual IOCs, like an IP, domain and URL, or leave blank as seen in Figure 3.5. It is a very easy and lightweight visualization platform, built to run on browsers, so one can use on any modern OS.

The stored data is saved on a JSON file so it can be easily indexed and searched by the front-end layer of timeline. Listing 3.1 is an example of the code skeleton and how the developer must create and standardize the JSON so it can better fit the tool:

Listing 3.1: JSON code

```
1 {
2   "title": {
3     "text": {
4       "headline": "Example timeline",
5       "text": "Events in the 7th century from https://\
        en.wikipedia.org/wiki/Timeline_of_the_Middle_Ages"
6     }
7   },
8   "events": [
9     {
10      "start_date": {
11        "year": "2015",
12        "month": "01"
13      },
14      "media": {
15        "url": "static/img/stick-man.png"
16      },
17      "text": {
18        "headline": "Example with own image",
19        "text": "Long conflict leaves both empires exhausted and \
        unable to cope with the newly united Arab armies under \
        Islam in the 630s"
20      }
21    },
22    {
23      "start_date": {
24        "year": "2015",
25        "month": "02"
26      },
27      "end_date": {
28        "year": "2015",
29        "month": "03"
30      },
31      "media": {
32        "url": "https://en.wikipedia.org/wiki/Grand_Canal_(China) "
33      },
34      "background": {
35        "opacity": "50",
36        "url": "https://upload.wikimedia.org/wikipedia/commons/a/ad/\
        Sui_Wendi_Tang.jpg"
37      }
38    },
```

```
39     "text": {
40         "headline": "Grand Canal in China is fully completed",
41         "text": "Its main role throughout its history was the \
                transport of grain to the capital."
42     }
43 }
44
45 ]
46 }
```

As can be seen in Script 3.1 JSON can be divided in two main sections: title (line 2) and events (line 8). The first is about the name of that specific timeline, it can be something like "Pegasus Spyware timeline" or "Case #2033 timeline", this option is free for the cyber analyst to imagine and should be selected to help the easier understanding as possible since from the analyst it self, as well as from other interested parties that might come to use this timeline. Perceive that timeline data must desirable be stored in fast indexing format. JSON format is recommended but developer can easily choose a storage format to work with.

The second part (line 8) it about handling with events it self. How to save and aggregate information fields so an event gets completely recorded to be used in the future. For this, developer must save the start date of the event, where it came from (URL, physical document, etc.) and its description with an headline and the text it self. Like this, the minimum identification criteria are met and events can now be stored and exhibited in a temporal format inside the timeline. As this is a JSON file, the bigger the time line is, bigger will be this file, so it is desirable to backup it periodically so any corruption, data loss or even an attack does not loses all data of the timeline. Again, developer is free to add more fields on JSON file (or in any type of file he chooses) to further increase the quantity of information saved on the tool. On this case, sometimes more information is not the best situation because people will look at it, so, like a Business Intelligence analyst have to always think: which is the best and most efficient approach to show all the information stored in a human-understandable way, that will allow users to look at that information and understand the most of it; having insights and more.

3.5.1 Differences between models

As seen, the model proposed in [9] with a 6-step division was used on this work to create the 8-step model that was explained earlier in this chapter. Regarding the Visualization and Analysis step, it can be seen that the IOCs aggregation with timestamps and timeline visualization helps a lot cybersecurity analyst's work since it shows in a more human friendly interface all threat data collected by the tool. Just as said in the beginning of this chapter, this is the main difference between our model and the other one.

4 CASE STUDIES AND DISCUSSION

In this chapter, two examples of real cases of threats will be studied, which will then be inserted into the model proposed by this work in such a way as to exemplify how such threats could be analyzed in the context of the 8-step model. For that, the procedure is to collect the IOCs of these two threats from the Internet, insert them into the tool and create a timeline visualization of their footprints in order to create a more "human friendly" approach. Also, it will be explained how robots, i.e., automated scripts were created to extract feed information from their respective web domains. For example, Fig 4.1 show a list of collected feeds that were used on this work that it will be explored further.

4.1 FEEDS SELECTION AND EXTRACTION

As discussed in chapter 3, these feeds can be free text that are available on some websites, and robots are, in this work case, piece of python codes that crawls through those feeds web-pages to gather all useful information inside them. And for the case studies, an analysis of two real cases will be explored to confirm the 8-steps methodological framework (focusing on the last 2 steps - Visualization and Analysis steps) and how it can improve CTI efficiency regardless threat identification and, further well, their mitigation.

First, the "feed collectors", as known as "robots" will be demonstrated and listed here, as well as the analysis view of a threat with its objects, relationships and other information. The list of Feed aggregated inside our tool are shown in Table 3.2. All of them are STIXv2/TAXII standardized. To do so, it's necessary to import some python libraries as shown in Listing 4.1:

Listing 4.1: Python Libraries

```
1 from app.classes.feed import Feed
2 from app.db.sqlite.feed import registerfeed
3 from app.db.sqlite.feed import findfeedbytitle, registerversion
4 from app.db.elastic.feed import registerfeedelk
5 from app.db.elastic.feed import selectfeedbyidsql
6 from app.db.elastic.collection import registercollectionelk
7 from app.classes.collection import Collection
8 from datetime import datetime
9 from flask import session
10 import hashlib
11 import requests
12 import json
13 from app.db.elastic.registerstix import insertoneobject
14 from stix2 import AttackPattern, Indicator, Relationship
```

List of feeds				
Show <input type="text" value="10"/> entries		Search: <input type="text"/>		
Title ↑↓	Description ↑↓	URL ↑↓	Shared with ↑↓	Action ↑↓
blocklist.de-680	lists of the attackers IP addresses of the last 48 hours pro service or all addresses	http://127.0.0.1:8999/blocklist.de-680	admin	
Enterprise-Attck-680	Att&ck - Enterprise feed	http://127.0.0.1:8999/Enterprise-Attck-680		
ICS -Attck-680	Att&ck - ICS feed	http://127.0.0.1:8999/ICS-Attck-680	admin	
mirai-5819	lists of possible Mirai botnet IP addresses	http://127.0.0.1:8999/mirai-5819		

Figure 4.1: Feeds List

4.1.1 Transformations applied to store Feeds

Then, the developer can create a function to store and collect all information from an specific feed - in this case "blocklist.de". He should specifies feed name and save it with a timestamp so it can be better organized inside the application. It is very important to hash each individual extracted information so it can check if any other ingested data in the future is the same that it's already present on the database. This is exemplified in Listing 4.2:

Listing 4.2: Python code

```

1 now = datetime.now()
2 feed_name = "blocklist.de-" + str(now.day) + str(now.month) + str(now.
  hour)
3 feed = Feed(0, feed_name, "lists of the attackers IP addresses of the
  last 48 hours pro service or all addresses", "", datetime.today().
  strftime('%Y-%m-%d'), "NO",
4     session.get('user_id'), "")
5 feed.hash = hashlib.md5((feed.title + str(feed.registered)).encode('utf-8
  ')).hexdigest()
6 feed.verifyNone()
7 registerfeed(feed)
8 feed = findfeedbytitle(feed.title)

```



```

9  registerversion("", feed.id)
10 feedback_feed = registerfeedelk(feed)
11
12 # InsertCollection - SSH
13 col = Collection(0, "ip_ssh", "Attacks on SSH", "All IP addresses which
      have been reported within the last 48 hours as having run attacks on
      the service SSH",
14     "", "false", "false", "", feedback_feed['_id'], datetime.today().
      strftime('%Y-%m-%d'))
15 col.verifycollection()
16 feedback = registercollectionelk(col, feedback_feed['_id'])

```

Then the user can start collecting individual objects from feed and save them at the tool. For that, it is necessary to get the response from the feed URL so it can be used inside python code. For this tool context, some fields are named to fit the database, just as shown in Listing 4.3:

Listing 4.3: Python code

```

1  #insert objects
2  response = requests.get('https://lists.blocklist.de/lists/ssh.txt')
3
4  report = AttackPattern(name="Attacks on SSH",
5      description="All IP addresses which have been
      reported within the last 48 hours as having
      run attacks on the service SSH. reference:
      https://lists.blocklist.de/")
6  report_j = json.loads(report.serialize(sort_keys=True))
7  report_j.update({"_collection": feedback['_id'], "_share": "NO", "_class":
      "OBJECT", "_iteration": "0",
8      "_inserted": datetime.today().strftime('%Y-%m-%d')})
9  insertoneobject(report_j)
10
11 for resp in response.iter_lines():
12     ip = "ipv4-addr:value=" + resp.decode("utf-8")
13     ind = Indicator(name="attack attemptive", description="evidence from
      attack attemptive", pattern=[ip],
14         pattern_type="ipv4-addr")
15     ind_j = json.loads(ind.serialize(sort_keys=True))
16     ind_j.update({"_collection": feedback['_id'], "_share": "NO", "_class":
      "OBJECT", "_iteration": "0",
17         "_inserted": datetime.today().strftime('%Y-%m-%d')})
18     insertoneobject(ind_j)
19
20     rel = Relationship(relationship_type="indicates", source_ref=ind_j['_
      id'], target_ref=report_j['_id'])
21     rel_j = json.loads(rel.serialize(sort_keys=True))
22     rel_j.update({"_collection": feedback['_id'], "_share": "NO", "_class":

```

```

    ": "OBJECT", "_iteration": "0",
23         "_inserted": datetime.today().strftime('%Y-%m-%d'))}
24     insertoneobject(rel_j)

```

4.1.2 Data extraction from Feeds

To extract data from "mirai" feed page, developer can use the code shown in Listing 4.4, which is very similar to the previous code. With this, it is possible to understand the pattern of feed extraction using python and all other already cited technologies.

Listing 4.4: Python code

```

1  from app.classes.feed import Feed
2  from app.db.sqlite.feed import registerfeed, findfeedbytitle
3  from app.db.elastic.feed import registerfeedelk, selectfeedbyidsql
4  from app.db.elastic.collection import registercollectionelk
5  from app.classes.collection import Collection
6  from datetime import datetime
7  from flask import session
8  import hashlib
9  import requests
10 import json
11 from app.db.elastic.registerstix import insertoneobject
12 from stix2 import AttackPattern, Indicator, Relationship
13
14
15 def collectdatafrommirai():
16
17
18 #InsertFeed
19 now = datetime.now()
20 feed_name = "mirai-" + str(now.day) + str(now.month) + str(now.hour)
21 feed = Feed(0, feed_name, "lists of possible Mirai botnet IP addresses", \
    "", datetime.today().strftime('%Y-%m-%d'), "NO",
22         session.get('user_id'), "")
23 feed.hash = hashlib.md5((feed.title + str(feed.registered)).encode('utf-8\
    ')).hexdigest()
24 feed.verifyNone()
25 registerfeed(feed)
26 feed = findfeedbytitle(feed.title)
27 registerversion("", feed.id)
28 feedback_feed = registerfeedelk(feed)
29
30
31 # InsertCollection - SSH
32 col = Collection(0, "ip", "Mirai botnet IPs", "Last 1000 IPs addresses \

```

```

        reported to run Mirai botnet attacks",
33         "", "false", "false", "", feedback_feed['_id'], datetime\
            .today().strftime('%Y-%m-%d'))
34 col.verifycollection()
35 feedback = registercollectionelk(col, feedback_feed['_id'])
36
37 # inser objects
38
39 response = requests.get('https://mirai.security.gives/data/ip_list.txt')
40
41 report = AttackPattern(name="ip-",
42                       description="Last 1000 IPs addresses reported to \
                           run Mirai botnet attacks")
43 report_j = json.loads(report.serialize(sort_keys=True))
44 report_j.update({"_collection": feedback['_id'], "_share": "NO", "_class"\
        : "OBJECT", "_iteration": "0",
45                 "_inserted": datetime.today().strftime('%Y-%m-%d')})
46 insertoneobject(report_j)
47
48 for resp in response.iter_lines():
49     if resp.decode("utf-8")[0] != "#":
50         ip = "ipv4-addr:value=" + resp.decode("utf-8")
51         ind = Indicator(name="botnet attack attemptive", description="\
            evidence from botnet attack attemptive", pattern=([ip]),
52                       pattern_type="ipv4-addr")
53         ind_j = json.loads(ind.serialize(sort_keys=True))
54         ind_j.update({"_collection": feedback['_id'], "_share": "NO", "\
            _class": "OBJECT", "_iteration": "0",
55                     "_inserted": datetime.today().strftime('%Y-%m-%d')\
            })
56         insertoneobject(ind_j)
57
58         rel = Relationship(relationship_type="indicates", source_ref=\
            ind_j['_id'], target_ref=report_j['_id'])
59         rel_j = json.loads(rel.serialize(sort_keys=True))
60         rel_j.update({"_collection": feedback['_id'], "_share": "NO", "\
            _class": "OBJECT", "_iteration": "0",
61                     "_inserted": datetime.today().strftime('%Y-%m-%d')\
            })
62         insertoneobject(rel_j)

```

With that, users will be able to download feeds to the platform and feeds will be available for exploration. It is important to reinforce that the process of timestamping is automatically done by this tool and the Feed is exhibited with a short timestamp (e.g., “blocklist.de-680”) indicating the time they they were downloaded into the tool, as shown in Figure 4.1.

Then, the cyber threat analyst will be able to look at the list of Collections of a specific Feed. These collections can be a specific type of vulnerability exploitation like botnets, *ssh* exploits and

any other known vulnerability, just like is shown in Figure 4.2. The user can analyze the Objects itself to see some text-information about them. It is possible to have some simple analytics views, like objects quantity and type and a raw view of the data collected.

Furthermore, an analyst can get information about a specific Collection such as numbers of Objects, with whom is it shared with, type of Objects and IOCs relationship view (Figure 4.3).

Therefore, to illustrate the proposed framework aggregated to the visualization step, the complete CTI process will be demonstrated - collection, treatment and visualization of IOCs arising from two recent attacks at the time of this writing: Pegasus Spyware (2021) and SolarWinds Orion Trojan (2020).









































List of collections			
Show <input type="text" value="10"/> entries		Search: <input type="text"/>	
ID	Shared with	Action	
apache		    	
bot		    	
bruteforcelogin		    	
ftp		    	
imap		    	
ip_ssh		    	
ircbot		    	
mail		    	

Figure 4.2: Collections List

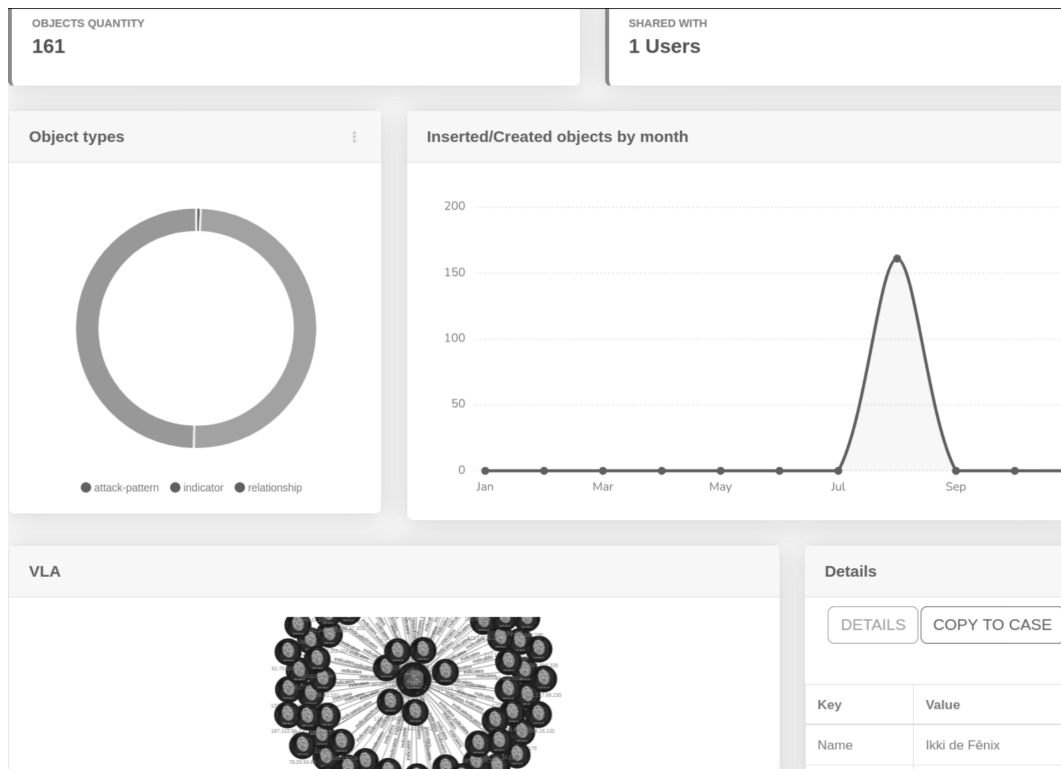


Figure 4.3: Threat analysis

4.2 PEGASUS SPYWARE ANALYSIS AND TIMELINE

Pegasus malware is an attack that affected personal Android and/or iOS smartphones with phishing clickbait or even zero-click attack, which do not require any interaction from the phone's owner to succeed. This malware was initially reported in early 2016 and might have gained access to good amount of smartphones. It was also used to target political users, disclosing confidential political information. It is still being reported at the year of 2022 with new zero-click smartphones exploits [46]

For this analysis, it is important to highlight that some Feed creators do not create the timestamp for each IOCs according to the period in which it was discovered. In fact, many Feeds have a timestamp for when the IOCs were added to that feed. The problem is that since they are added in one batch, they are timestamped on the same day, hour and minute, changing only the milliseconds in which a computer script took to add all IOCs to the Feed.

This problem makes an analysis of the threat in the timeline unfeasible since it makes no sense to display a timeline with all the IOCs at the same moment of time, especially when this moment is the time when the computer added the IOCs to the Feed and not the actual time that those IOCs were discovered by a Threat Hunter. The idea of the timeline visualization is to create a better temporal understating about that threat, helping the understanding of the CTI engineer about the campaign of that threat. So, for this demonstration, it was acquired Pegasus IOCs from a Feed

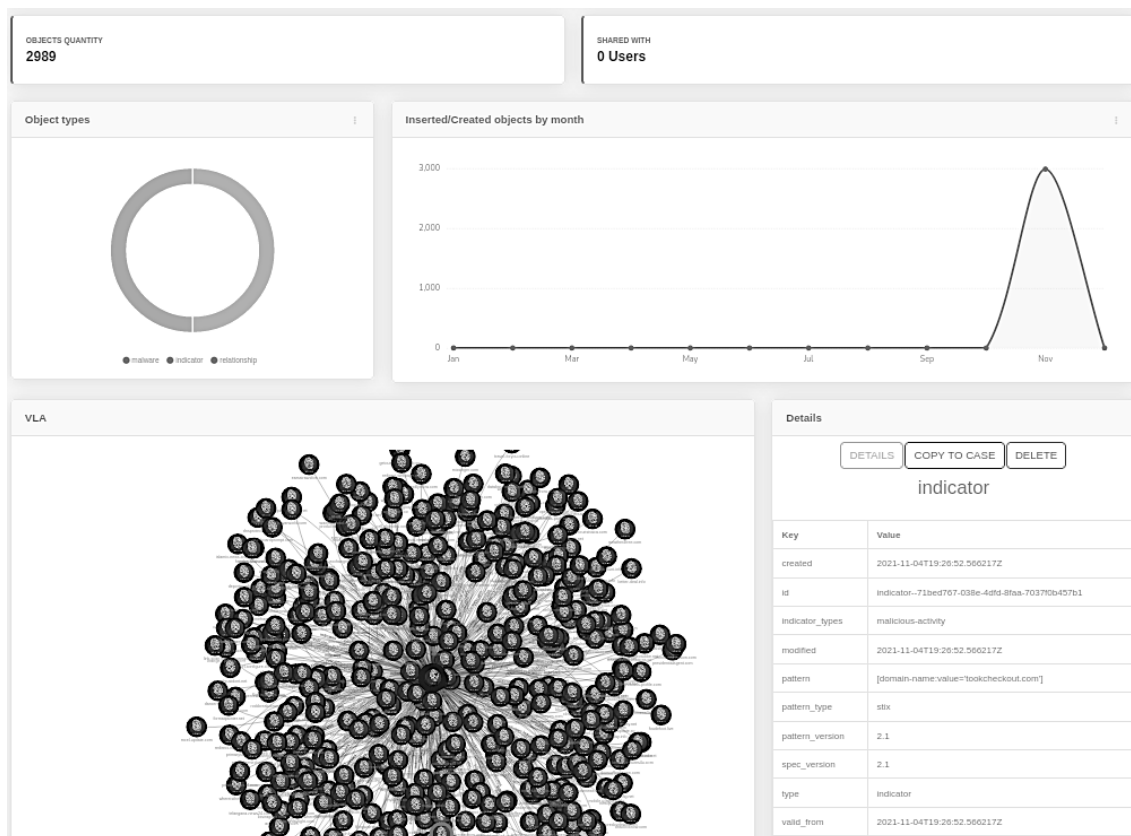


Figure 4.4: Pegasus analysis

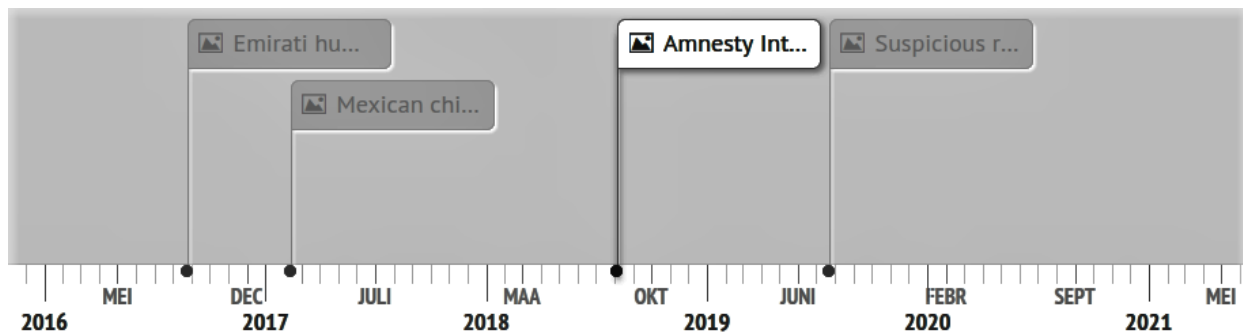
and cyber threat specialized sites as well.

Pegasus feed was found at Amnesty Tech (the same company that had their employees smartphones hacked by Pegasus) Github repository [47] and was STIXv2 formatted, so it was inserted inside thistool. After the insertion, the tool formatted all ingested data and a Feed called “Pegasus” was created. As shown in Figure 4.4, there are 2989 objects that are correlated with Pegasus, most of them being Indicators of Compromise of the spyware.

Figure 4.5 shows Pegasus timeline with 4 events that was registered by two different organizations: Citizen Lab and Amnesty International. Those events were independently related and were not correlated at the time. With this timeline view, it is possible to see how the spyware spread out to his targets - most of all iPhone users that were lured to click on a link that downloaded the malicious content to their IOS phone system. This attack was brought to light in 2021 but has been reported since 2016 as it is possible to see in the timeline view.

This kind of information could have helped a security analyst to mitigate Pegasus damage earlier, as he would be able to see that the spyware was making more and more victims from different nationalities.

It is important to highlight that the example of Pegasus used in this case has a more textual description of the attack, and the next SolarWinds timeline analysis was created using a more "IOCs focused" analysis. This was purposefully done to demonstrate that thistool supports both types of analysis and that it is possible to create any type of temporal visualization, depending on



6 AUGUSTUS 2018

AMNESTY INTERNATIONAL'S STAFF MEMBERS

Amnesty International said one of its staff members, as well as several Saudi human rights defenders, had been targeted with Pegasus software, using text messages with links

Figure 4.5: Pegasus timeline

which is best for a specific threat campaign.

4.2.1 MITRE ATT&CK analysis

Going further on this analysis, it is possible to use MITRE ATT&CK framework to correlate Pegasus' footprints with the data on the timeline. For that, Table 4.1 shows all techniques that MITRE mapped when talking about Pegasus for Android.

To further demonstrate how this analysis is following MITRE ATT&CK best practices, it will be done a comparison to how the Pegasus attack would be used inside MITRE MATRIX. Figure 4.5 shows that Amnesty International staff members were hacked using text messages with links. We can correlate this with the MITRE ATT&CK "Alternate Network Mediums" technique, that Pegasus used SMS to command and control Android and iOS phones.

Another possible correlation can be the last data inserted in the timeline that correlates with "Capture Audio" MITRE ATT&CK technique and could be linked to further enrich the CTI data about Pegasus Spyware for Android phones.

The above examples and correlation with MITRE ATT&CK shows how a cyber analyst can innovate and use this methodological framework alongside with other known-market frameworks like ATT&CK to constantly upgrade the quality of TI data about every known threat. If the timeline is continuously updated and fed, it can store a lot of important information to help understand attacker's pattern or how it is spreading through networks, users and even countries.

Table 4.1: List techniques used by Pegasus Spyware for Android (Adapted from([48])

Technique name	How it is used?
Access Calendar Entries	Pegasus for Android accesses calendar entries
Access Call Log	Pegasus for Android accesses call logs
Access Contact List	Pegasus for Android accesses contact list information
Access Stored Application Data	Pegasus for Android accesses sensitive data in files, such as messages stored by the WhatsApp, Facebook, and Twitter applications. It also has the ability to access arbitrary filenames and retrieve directory listings
Alternate Network Mediums	Pegasus for Android uses SMS for command and control
Application Discovery	Pegasus for Android accesses the list of installed applications
Broadcast Receivers	Pegasus for Android listens for the BOOT_COMPLETED broadcast intent in order to maintain persistence and activate its functionality at device boot time
Capture Audio	Pegasus for Android has the ability to record device audio
Capture Camera	Pegasus for Android has the ability to take pictures using the device camera
Deliver Malicious App via Authorized App Store	Pegasus for Android attempts to detect whether it is running in an emulator rather than a real device
Exploit OS Vulnerability	Pegasus for Android attempts to exploit well-known Android OS vulnerabilities to escalate privileges
Modify System Partition	Pegasus for Android attempts to modify the device's system partition
System Network Configuration Discovery	Pegasus for Android checks if the device is on Wi-Fi, a cellular network, and is roaming

The analysis using our 8-step model alongside other known frameworks like MITRE ATT&CK improves the quality of threat understanding and enriching possibilities to defend against it. Cyber analyst can now create a threat footprint using many different sources within the tool, visualize and analyze all this data and indicators, and finally connect it all with MITRE ATT&CK techniques, enriching even more all available information. Moreover, it helps in the field of sharing threat information through all means and tools mainly because MITRE is a very used framework by all companies.

4.3 SOLARWINDS SUNBURST TIMELINE

In early December 2020 [49], the security company FireEye announced that they were attacked by a supposedly nation state-backed attack group (probably a Russian group named Cozy Bear [50] [51]) that stole tools that were used by FireEye red-hackers team. The attackers exploited a vulnerability in Orion updates – a popular IT infrastructure management software that is distributed by SolarWinds [52] - and infiltrated through FireEye network, inserting their malware on their servers and opening a backdoor to steal data.

For this analysis, it was used IOCs data acquired from FireEye GitHub repository [53]. IOCs were not formatted on STIX format, so they were manually added to the timeline. This example shows that the cybersecurity engineer will not always find threat feeds formatted in STIX/TAXII standards, but that, with this tool, he will still be able to create a timeline with any information found on websites and documents that contain useful indicators for a more complete analysis of the threat. Figure 4.6 shows part of the timeline created for SunBurst trojan.

The data breach occurred on December 8th of 2020. Analyzing the timeline, it is possible to see that “strange” domain traffic flows were noticed on FireEye network logs. These domains were not related to and attack because probably they did not realize the increasing frequency of unknown domains connections inside their network traffic. So, with the timeline analysis, the Cyber Analyst could notice the increase in these domains and perhaps become more alert or even start investigating this case more closely before the damage was done.

4.3.1 MITRE ATT&CK analysis

Again, going further on this analysis context using MITRE ATT&CK framework to correlate SolarWinds SunBurst’s footprints with the data on the timeline, Table 4.2 and Table 4.3 shows all techniques that MITRE mapped when talking about SolarWinds SunBurst. Figure 4.6 shows that a Malicious Domain identified as IP: *34.203.203.23* and *websitetheme.com* was detected as an indicator of the until unknown Sunburst trojan. We can correlate this with some MITRE ATT&CK techniques like, e.g., Application Layer Protocol: Web Protocols and Application Layer Protocol: DNS. The above example and correlation with MITRE ATT&CK shows how a security engineer can perceive a traffic pattern and then open a case to study that traffic and maybe identify exploit attempts early. Imagine if in this case the analyst opened a case that used the malicious domain and IP and shared it with other stakeholders saying that this might be an attack technique based on one (or all) of ATT&CK technique(s). This could help improve the time to detect and discover the trojan.

Table 4.2: List techniques used by SolarWinds SunBurst Trojan (Adapted from [54])

Technique name	How it is used?
Application Layer Protocol: Web Protocols	SUNBURST communicated via HTTP GET or HTTP POST requests to third party servers for C2
Application Layer Protocol: DNS	SUNBURST used DNS for C2 traffic designed to mimic normal SolarWinds API communications
Command and Scripting Interpreter: Visual Basic	SUNBURST used VBScripts to initiate the execution of payloads
Data Encoding: Standard Encoding	SUNBURST used Base64 encoding in its C2 traffic
Data from Local System	SUNBURST collected information from a compromised host
Data Obfuscation: Junk Data	SUNBURST added junk bytes to its C2 over HTTP
Data Obfuscation: Steganography	SUNBURST C2 data attempted to appear as benign XML related to .NET assemblies or as a faux JSON blob
Data Obfuscation: Protocol Impersonation	SUNBURST masqueraded its network traffic as the Orion Improvement Program (OIP) protocol
Dynamic Resolution	SUNBURST dynamically resolved C2 infrastructure for randomly-generated subdomains within a parent domain
Encrypted Channel: Symmetric Cryptography	SUNBURST encrypted C2 traffic using a single-byte-XOR cipher
Event Triggered Execution: Image File Execution Options Injection	SUNBURST created an Image File Execution Options (IFEO) Debugger registry value for the process dllhost.exe to trigger the installation of Cobalt Strike
File and Directory Discovery	SUNBURST had commands to enumerate files and directories
Impair Defenses: Disable or Modify Tools	SUNBURST attempted to disable software security services following checks against a FNV-1a + XOR hashed hardcoded blacklist
Indicator Removal on Host	SUNBURST removed IFEO values to clean up traces of execution
File Deletion	SUNBURST had a command to delete files
Ingress Tool Transfer	SUNBURST delivered different payloads, including TEARDROP in at least one instance
Masquerading: Match Legitimate Name or Location	SUNBURST created VBScripts that were named after existing services or folders to blend into legitimate activities

Table 4.3: Continuation of list techniques used by SolarWinds SunBusrt Trojan (Adapted from [54])

Technique name	How it is used?
Modify Registry	SUNBURST had commands that allow an attacker to write or delete registry keys, and was observed stopping services by setting their registry entries to value 4
Obfuscated Files or Information	SUNBURST strings were compressed and encoded in Base64. SUNBURST also obfuscated collected system information using a FNV-1a + XOR algorithm
Indicator Removal from Tools	SUNBURST source code used generic variable names and pre-obfuscated strings, and was likely sanitized of developer comments before being added to SUNSPOT
Process Discovery	SUNBURST collected a list of process names that were hashed using a FNV-1a + XOR algorithm to check against similarly-hashed hard-coded blocklists
Query Registry	SUNBURST collected some registries values from compromised hosts
Signed Binary Proxy Execution: Rundll32	SUNBURST used Rundll32 to execute payloads
Software Discovery: Security Software Discovery	SUNBURST checked for a variety of antivirus/endpoint detection agents prior to execution
Subvert Trust Controls: Code Signing	SUNBURST was digitally signed by SolarWinds from March - May 2020
System Information Discovery	SUNBURST collected hostname, OS version, and device uptime
System Network Configuration Discovery	SUNBURST collected all network interface MAC addresses that are up and not loopback devices, as well as IP address, DHCP configuration, and domain information
System Owner/User Discovery	SUNBURST collected the username from a compromised host
System Service Discovery	SUNBURST collected a list of service names that were hashed using a FNV-1a + XOR algorithm to check against similarly-hashed hard-coded blocklists
Virtualization/Sandbox Evasion: System Checks	SUNBURST checked the domain name of the compromised host to verify it was running in a real environment
Virtualization/Sandbox Evasion: Time Based Evasion	SUNBURST remained dormant after initial access for a period of up to two weeks
Windows Management Instrumentation	SUNBURST used the WMI query <code>Select * From Win32_SystemDriver</code> to retrieve a driver listing

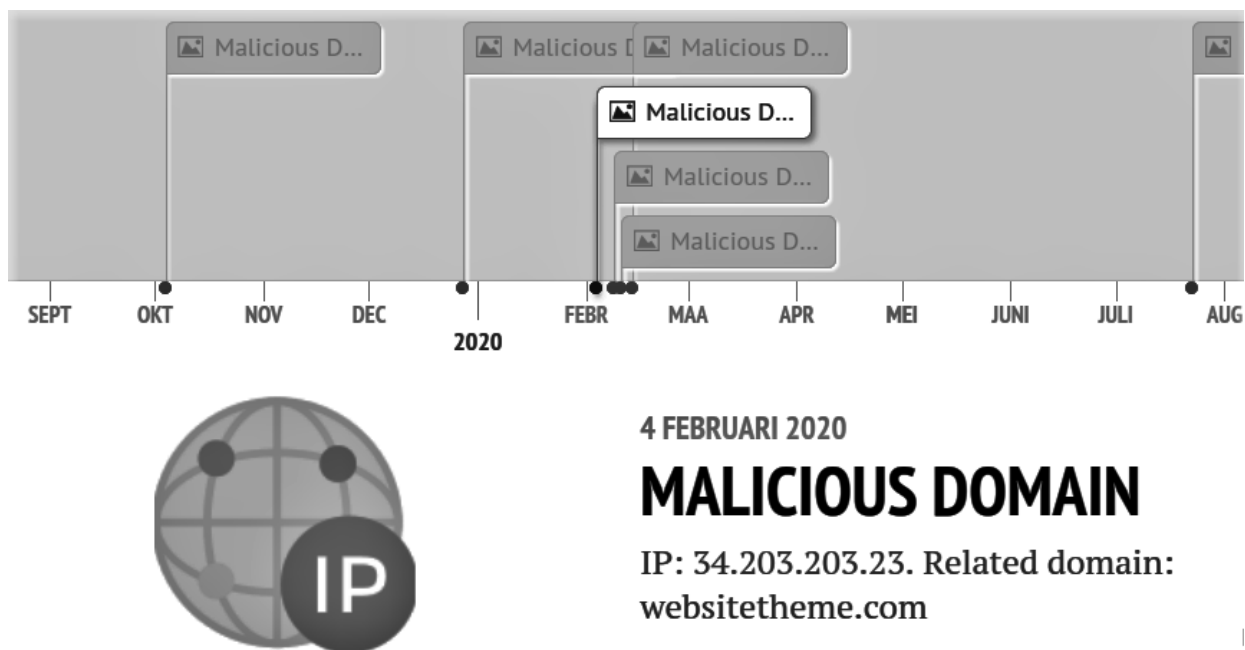


Figure 4.6: SolarWinds SunBurst trojan timeline

Another important advantage is to see how the attackers are behaving, i.e., if they are only mapping their targets' infrastructure/systems or if they have already started their attack campaign against their target, for example. Thereby, both Pegasus and Amnesty analysis examples provide cyber analysts with a very valuable point of view since it brings an human-friendly approach to visualize threat history and behavior.

In the same reasoning of Pegasus, this analysis using our 8-step model with MITRE ATT&CK List of Techniques enriches threat data and foment the sharing of all understood data of that or any specific threat, since MITRE ATT&CK is a very common standard around the world and cyber analysts will be thrilling to correlate all this information to mitigate a threat.

4.4 CHAPTER SUMMARY

In this chapter, it was shown how to extract, store and normalize feeds in order to use them together with threat analysis. All necessary Python code were described as well as two real case examples that were analyzed to exemplify how a Cybersecurity Analyst could create his analysis of a specific threat equipped with more useful threat data like more IOCs and any other information that can improve threat identification and mitigation.

5 CONCLUSIONS AND FUTURE WORK

This work proposes a methodological framework to improve security engineers work life related to Cyber Threat Intelligence. Nowadays, there are no tools that allow to collect threat data from Feeds and other sources and insert, filter and visualize them in customizable ways. Hence, with the proposed eight-step CTI model, developers can build an application to address CTI needs and improve the previous discussed collection, storage, filtering, analysis, visualization and sharing of cyber threat data when gathered from multiple sources. Additionally, with the main idea of exploring data to address a particular case construction, CTI data from multiples sources is still a challenging problem that demonstrates the need for a methodological framework capable of allowing a consistent and reliable case construction.

Pegasus Spyware example explored an attack that plagued politicians, celebrities and even ordinary people. A lot of information was available on the Internet in an unstructured and non-standardized form, making it difficult to detect and standardize the attack. The same reality is almost equally valid for the SolarWinds SunBurst attack. The difference in this case is that the attack was targeted directly on a cybersecurity company, which detected and recorded all the main indicators of the threat.

Thereby, these two examples illustrated how threat data is diffused and non-standardized, reinforcing the latent need to standardize the registry of threats and vulnerabilities as much as possible. This encourages the sharing of information between all players interested in protecting themselves, thus allowing a quick mitigation of these threats, using any existent tool to collect, process, explore, analyze and generate valuable insights in the context of Cyber Threat Intelligence and threat mitigation.

For this matter and to acknowledge the main objective of this work, we aimed to build a user-friendly interface to facilitate user usage of a CTI tool and to give focus on what it is necessary: threat discovery, countermeasures and information sharing along stakeholders. We achieved our goal with the usage of a tool as demonstration of our 8-step model and showed how our novel steps foments an increase in CTI data quality trough visualizing and analyzing threat data as well as sharing it using common industry standards like MITRE ATT&CK Techniques.

5.1 FUTURE WORK

For the Visualization step, future work will explore the improvement of the timeline, allowing the user to attach documents like PDFs or URLs with hyperlinks so it can be easier to use the tool as the main CTI tool of the company's Security Operation Center.

Other important improvement will be on the Analysis step, where it can be created more kind of analytics visualizations so that the cyber security analyst has more information at their disposal

in a modular and customizable way, i.e., according to his choice.

Finally, another important future work that will greatly contribute to this theme is the development of methodologies to collect unstructured and structured data, as well as sharing protocols of IOCs data that will be useful for the correct implementation and execution of CTI best practices.

BIBLIOGRAPHY

- 1 IBM. X-force threat intelligence index. v. 1, 2020. Available at: <<https://www.ibm.com/downloads/cas/M1X3B7QG>>.
- 2 FBI. 2020 internet crime report. 2020. Available at: <https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf>.
- 3 SHACKLEFORD, D. Who's using cyberthreat intelligence and how? – a sans survey. 2017.
- 4 SUN, T.; YANG, P.; LI, M.; LIAO, S. An automatic generation approach of the cyber threat intelligence records based on multi-source information fusion. 2021. Available at: <<https://www.mdpi.com/1999-5903/13/2/40>>.
- 5 KOLOVEAS, P.; CHANTZIOS, T.; ALEVIZOPOULOU, S.; SKIADOPOULOS, S.; TRYFONOPOULOS, C. intime: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence. *Electronics*, 2021.
- 6 PREUVENEERS, D.; JOOSEN, W. Sharing machine learning models as indicators of compromise for cyber threat intelligence. *Journal Of Cybersecurity And Privacy*, p. 140–163, 2021.
- 7 HAASTRECHT, M.; GOLPUR, G.; TZISMADIA, G.; KAB, R. A shared cyber threat intelligence solution for smes. *Electronics*, 2021. Available at: <<https://www.mdpi.com/2079-9292/10/23/2913>>.
- 8 ARIGANELLO, J. *The Benefits of Sharing Threat Intelligence Inside and Outside Your Organization*. 2022. Available at: <<https://www.anomali.com/blog/the-need-to-share>>.
- 9 AZEVEDO, B.; GIOZZA, W.; MENDONÇA, F. L.; FILHO, D. D. S.; JUNIOR, R. de S.; ALBUQUERQUE, R. Proposta de modelo de referencia de inteligencia de ameacas. 2020. Available at: <http://dx.doi.org/10.33965/ciawi2019_201914L006>.
- 10 AMARO, L. J. B.; AZEVEDO, B. W. P.; MENDONÇA, F. L. Lopes de; GIOZZA, W. F.; ALBUQUERQUE, R. d. O.; VILLALBA, L. J. G. Methodological framework to collect, process, analyze and visualize cyber threat intelligence data. *Applied Sciences*, v. 12, n. 3, 2022. ISSN 2076-3417. Available at: <<https://www.mdpi.com/2076-3417/12/3/1205>>.
- 11 OLTSIK, J. The life and times of cybersecurity professionals. *ESG and ISSA: Research Report.*, 2017.
- 12 CONTI, M.; DARGAHI, T.; DEGHANTANHA, A. Cyber threat intelligence: Challenges and opportunities.. *Advances in Information Security, vol 70. Springer, Cham.*, 2018.
- 13 TD, W.; K, M.; E, P.; AE., A. Cyber threat intelligence sharing: Survey and research directions. *Comput. Security.*, 2019.
- 14 STAFF, U. J. C. of. *Joint Publication 2-0 Joint Intelligence*. 2013. Available at: <https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp2_0.pdf>.

- 15 BARNUM, S. Standardizing cyber threat intelligence information with the structured threat information expression (stix). 2012. Available at: <<https://www.mitre.org/publications/technical-papers/standardizing-cyber-threat-intelligence-information-with-the>>.
- 16 MITRE. *Introduction to TAXII*. 2021. Available at: <<https://oasis-open.github.io/cti-documentation/taxii/intro.html>>.
- 17 ABU, M.; RAHAYU, S.; (DRAA), D. A. A.; ROBIAH, Y. Cyber threat intelligence—issue and challenges. *Indonesian Journal of Electrical Engineering and Computer Science.*, v. 10, 2018.
- 18 GAO, P.; SHAO, F.; LIU, X.; XIAO, X.; QIN, Z.; XU, F.; MITTAL, P.; KULKARNI, S.; SONG, D. Enabling efficient cyber threat hunting with cyber threat intelligence. *IEEE 37th International Conference on Data Engineering (ICDE)*, 2021.
- 19 WARNER, J.; HINCK, S. *How Threat Hunting Can Evolve Your Detection Capabilities*. 2018. Available at: <<https://blog.gigamon.com/2018/09/27/how-threat-hunting-can-evolve-your-detection-capabilities/>>.
- 20 CORPORATION, L. M. *Putting Intelligence to Work. An intelligent, defensive posture begins with the best inputs and ability to process them*. 2022. Available at: <<https://www.lockheedmartin.com/en-us/capabilities/cyber/intelligence-driven-defense.html>>.
- 21 SERGIO PENDERGAST ANDREW, B. C. C. *The Diamond Model of Intrusion Analysis*. 2013. Available at: <<https://apps.dtic.mil/sti/citations/ADA586960>>.
- 22 STROM, B. Att&ck 101. 2018. Available at: <<https://medium.com/mitre-attack/att-ck-101-17074d3bc62>>.
- 23 IBM. *MITRE ATT&CK mapping and visualization*. 2021. Available at: <<https://www.ibm.com/docs/pl/qradar-common?topic=app-mitre-attck-mapping-visualization>>.
- 24 SPLUNKBASE. *MITRE ATTACK App for Splunk*. 2021. Available at: <<https://splunkbase.splunk.com/app/4617/>>.
- 25 MITRE. *About CybOX (Archive)*. 2020. Available at: <<https://cyboxproject.github.io/about>>.
- 26 MITRE. *CybOX 3.0*. 2022. Available at: <<https://cyboxproject.github.io/cybox3.0/>>.
- 27 OASIS. *Comparing STIX 1.X/CybOX 2.X with STIX 2*. 2021. Available at: <<https://oasis-open.github.io/cti-documentation/stix/compare>>.
- 28 SILVA, A.; GONDIM, J.; ALBUQUERQUE, R.; VILLALBA, L. G. A methodology to evaluate standards and platforms within cyber threat intelligence. *Future Internet.*, 2020.
- 29 TOUNSI, W.; RAIS, H. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & Security*, v. 72, p. 212–233, 2018.
- 30 RAMSDALE, A.; SHIAELES, S.; KOLOKOTRONIS, N. A comparative analysis of cyber-threat intelligence sources, formats and languages. p. 824, 2020.
- 31 SCHLETTE, D.; BÖHM, F.; CASELLI, M.; PERNUL, G. Measuring and visualizing cyber threat intelligence quality. *International Journal of Information Security*, 2020.

- 32 GAO, Y.; LI, X.; PENG, H.; FANG, B.; YU, P. Hincti: A cyber threat intelligence modeling and identification system based on heterogeneous information network. 2020. Available at: <<https://ieeexplore.ieee.org/abstract/document/9072563>>.
- 33 RIESCO, R.; LARRIVA-NOVO, X.; VILLAGRA, V. Cybersecurity threat intelligence knowledge exchange based on blockchain. *Telecommun Syst*, p. 73, 259–288, 2020.
- 34 SAXENA, R.; GAYATHRI, E. Cyber threat intelligence challenges: Leveraging blockchain intelligence with possible solution. *Materials Today: Proceedings*, v. 51, p. 682–689, 2022. ISSN 2214-7853. CMAE'21. Available at: <<https://www.sciencedirect.com/science/article/pii/S2214785321045752>>.
- 35 ZHANG, W.; BAI, Y.; FENG, J. Tiia: A blockchain-enabled threat intelligence integrity audit scheme for iiot. *Future Generation Computer Systems*, v. 132, p. 254–265, 2022. ISSN 0167-739X. Available at: <<https://www.sciencedirect.com/science/article/pii/S0167739X22000723>>.
- 36 ODEMIS, M.; YUCEL, C.; KOLTUKSUZ, A. Detecting user behavior in cyber threat intelligence: Development of honeypsy system. *Security and Communication Networks*, vol. 2022, 2022.
- 37 LLC., P. I. *The Value of Threat Intelligence: Annual Study of North American & United Kingdom Companies*. 2021. Available at: <https://stratejm.com/wp-content/uploads/2019/08/2019_Ponemon_Institute-Value_of_Threat_Intelligence_Research_Report_from_Anomali.pdf>.
- 38 GROUP, I. *China-linked Group RedEcho Targests the Indian Power Sector Amid Heightened Border Tensions*. 2021. Available at: <<https://www.recordedfuture.com/redecho-targeting-indian-power-sector/>>.
- 39 BLOCKLIST. *Lists of malicious IPs*. 2021. Available at: <<https://lists.blocklist.de/lists/all.txt>>.
- 40 FIREHOL. *Blacklist of malicious IPs*. 2021. Available at: <https://raw.githubusercontent.com/ktsaou/blocklist-ipsets/master/firehol_level1.netset>.
- 41 MIRAI. *List of the last 1000 likely IPs of machinesinfected with Mirai Botnet ransomware*. 2021. Available at: <https://mirai.security.gives/data/ip_list.txt>.
- 42 OPENPHISH. *List of phishing URLs*. 2021. Available at: <<https://openphish.com/feed.txt>>.
- 43 PANUNIT42. *List of malicious masked URLs*. 2021. Available at: <<https://openphish.com/feed.txt>>.
- 44 VXVAULT. *List of malicious and downloadable .dll and .exe files*. 2021. Available at: <http://vxvault.net/URL_List.php>.
- 45 ZERODOT1. *List of blockchain mining bot domains to be blocked by a network admin*. 2021. Available at: <<https://gitlab.com/ZeroDot1/CoinBlockerLists/raw/master/list.txt>>.
- 46 PAGANINI, P. *NSO Group Pegasus spyware leverages new zero-click iPhone exploit in recent attacks*. 2022. Available at: <<https://securityaffairs.co/wordpress/130360/malware/nso-group-pegasus-click-iphone-exploit.html>>.

- 47 AMNESTYTECH. *Indicators from Amnesty International's investigations*. 2021. Available at: <https://raw.githubusercontent.com/AmnestyTech/investigations/master/2021-07-18_nso/pegasus.stix2>.
- 48 ATT&CK, M. *Pegasus for Android*. 2022. Available at: <<https://attack.mitre.org/software/S0316/>>.
- 49 FIREEYE. *FireEye Mandiant SunBurst Countermeasures*. 2021. Available at: <https://github.com/fireeye/sunburst_countermeasures/tree/main/indicator_release>.
- 50 CYBERSCOOP. *How the Russian hacking group Cozy Bear, suspected in the SolarWinds breach, plays the long game*. 2021. Available at: <<https://www.cyberscoop.com/cozy-bear-apt29-solarwinds-russia-persistent>>.
- 51 POST, T. W. *Russian government hackers are behind a broad espionage campaign that has compromised U.S. agencies, including Treasury and Commerce*. 2021. Available at: <https://www.washingtonpost.com/national-security/russian-government-spies-are-behind-a-broad-hacking-campaign-that-has-breached-us-agencies-and-a-top-cyber-firm/2020/12/13/d5a53b88-3d7d-11eb-9453-fc36ba051781_story.html>.
- 52 KHALILI, J. *The inside story of the infamous SolarWinds hack*. 2022. Available at: <<https://www.techradar.com/news/the-inside-story-of-the-infamous-solarwinds-hack>>.
- 53 FIREEYE. *FireEye Mandiant SunBurst Countermeasures*. 2021. Available at: <https://github.com/fireeye/sunburst_countermeasures/tree/main/indicator_release>.
- 54 ATT&K, M. *Solarwinds SUNBURST*. 2022. Available at: <<https://attack.mitre.org/software/S0559/>>.