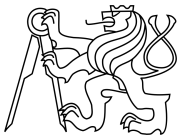
CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF INFORMATION TECHNOLOGY



ASSIGNMENT OF MASTER'S THESIS

Title:	Utilization of Threat Intelligence in Information Security
Student:	Bc. Marek Bertovič
Supervisor:	Ing. Filip Štěpánek
Study Programme:	Informatics
Study Branch:	Computer Security
Department:	Department of Computer Systems
Validity:	Until the end of winter semester 2018/19

Instructions

Threat intelligence (TI) is evidence-based knowledge about an existing or emerging menace or hazard to IT infrastructure that can be used to inform decisions regarding the subject's response to that hazard. This knowledge can be processed by existing platforms for collecting security information and event management. The goal of the thesis is to design and implement correlation algorithms for TI platform of student's choice that would detect or visualize threats aiming at the managed IT infrastructure.
Tasks:
- Analyze the problem of TI and according to the analysis choose a suitable platform for integration and describe the integration process.
- Design TI correlation algorithms for the platform.
- Integrate the correlation algorithms into the platform.
- Validate the integration in laboratory environment.
- Describe how the integration enhanced the detection of incoming threats capabilities of the selected platform.

References

Will be provided by the supervisor.

prof. Ing. Róbert Lórencz, CSc.
Head of Department

prof. Ing. Pavel Tvrdík, CSc.
Dean

Prague February 19, 2017

Czech Technical University in Prague
Faculty of Information Technology
Department of Computer Systems

Master's thesis

Utilization of Threat Intelligence in Information Security

Bc. Marek Bertovič

Supervisor: Ing. Filip Štěpánek

9th May 2017

Acknowledgements

I would like to thank to Ing. Filip Štěpánek for his supervising, consultations and his unconditional willingness to help.

I would also like to thank Mgr. Tobiáš Smolka for sharing his admirable knowledge and expertise. This thesis would not have the desired quality without Filip and Tobiáš.

This thesis would not happen if my family haven't supported me during my studies for which I would like to thank them.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on 9th May 2017 .....

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Bertovič, Marek. *Utilization of Threat Intelligence in Information Security.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2017.

Abstrakt

Práce se zabývá problémem threat intelligence. Je zde diskutován výběr vhodné SIEM platformy a návrh korelačních algoritmů. Tyto algoritmy jsou zaměřeny na detekci TOR, phishingu a ransomware. Součastí je popis návrhu korelačních algoritmů a jejich integrace do zvolené platformy. Výsledné řešení je testováno v laboratorním prostředí a výsledky jsou vyhodnoceny.

Klíčová slova cyber threat intelligence, splunk, SIEM, threat intelligence management platform, TOR, ransomware, phishing

Abstract

This thesis focuses on the problem of "threat intelligence". Selection of suitable SIEM platform and design of the correlation algorithms is discussed. These algorithms aim at TOR communication, phishing and ransomware detection. It includes design and description of the correlation algorithms and their integration in the selected platform. The final results are validated in the laboratory environment and are further evaluated and discussed.

Keywords cyber threat intelligence, splunk, SIEM, threat intelligence management platform, TOR, ransomware, phishing

Contents

List of Figures

Introduction

Threat Intelligence is a knowledge that helps to identify security threats in the IT infrastructure and to make informed decisions. That means decisions that are based on facts or information. It is used by various security teams, including chiefs information security officers(CISOs), security managers, security incident response teams and members of security operation center team. This group of people is responsible for the security of the organization and its clients. Their role is to protect the organizational IT infrastructure from internal and external threats to ensure confidentiality, integrity and availability of the information assets.

Currently, there is not out of the box free or open-source solution that would utilize the threat intelligence in a way that makes it actionable. Actionable means that information value gained from threat intelligence can be used to reduce or eliminate the risk of successful attack. Such solution should be able to automatically collect, process, store, normalize, enrich and utilize threat intelligence data in order to improve and enhance the detection capabilities of the organization. Utilization also means that this data will be automatically correlated with various data sources that are present in the organization as network communication, proxy data or even endpoint data. This data includes logs and events produced by various devices like firewalls, audit logs or proxy events.

This thesis wants to bring a concept that is based on tools that are open-source or free for certain period of time and show how they can be integrated. Moreover, it wants to utilize a specific set of correlation algorithms to make the threat intelligence actionable. Described set of correlation algorithms is focused on detecting risks that are trending in cyber security. Three topics have been selected (TOR, ransomware, phishing). Early detection of these threats can significantly reduce the damage. They are further discussed in the thesis. This would help people who want to get into the threat intelligence community to have something to start with. It would also help organizations that do not have threat intelligence program to have a solid foundation

where they can start off. Organizations may take this concept and enhance it with correlation algorithms that are critical for their operations and business. Academia can use it for doing further research in the threat intelligence field by having an open solution that can utilize and make threat intelligence actionable.

This thesis describes the problem of threat intelligence. It is a relatively new field, therefore the basic terminology, closely related terms and problem statement is described in the analysis chapter. The analysis also describes the current state of the art in the field of threat intelligence and the common platforms used in the industry. Furthermore, threat intelligence cycle model is analyzed and requirements on the proposed solution/platform are discussed. According to the analysis, a suitable platform is selected. Correlation algorithms that would enhance the threat detection capabilities of the selected platform are discussed as well in this chapter.

Detailed design of the proposed solution/platform is described in the design chapter. Furthermore, use cases that suit as a basis for the design of the correlation algorithms are also described in the design chapter. These use cases are used to enhance the threat detection capabilities.

The process of the integration and implementation of the correlation algorithms into the selected platform is described in the realization chapter. There are also described and detailed various approaches and techniques and their performance is measured and compared.

The results are validated in the laboratory environment. This is described in the testing chapter.

Chapter 1

Analysis

This chapter defines what threat intelligence is, where does it come from, how it can be processed and then used to enhance detection of threats aiming at the organization. Furthermore, it defines threat intelligence cycle, an abstract model that is important in the utilization of threat intelligence. Platforms that play a key role in this cycle are then defined, described and proposed.

1.1 Threat Intelligence

Threat Intelligence is a young field. By measuring a number of academic publications containing keyword *cyber threat intelligence* until 2010 was about 20. Currently, the number of all academic publications containing this keyword has increased to 676 from which 475 has been published in last 2 years according to *scholar.google.com* using the keyword "cyber threat intelligence" (see Figure 1.1).

Threat Intelligence is a consequence of cyber security incident response teams and governments agencies need to share information about cyber threats. The cyber threat is a possibility or malicious attempt to damage or disrupt a computer network or system [5]. Disrupting means any unauthorized change in confidentiality, integrity or availability of the information asset (see CIA principle [6]). Currently, many vendors and security practitioners have different opinions on how to define threat intelligence. For the purpose of this thesis, terms and concepts are defined that are later used in the thesis.

First, let's define the terms *intelligence* and *Threat Intelligence* and how they can be represented and categorized. Federal Bureau Investigation defines intelligence as [7]:

> *Intelligence is information that has been analyzed and refined so that it is useful to policymakers in making decisions—specifically, decisions about potential threats to national security.*

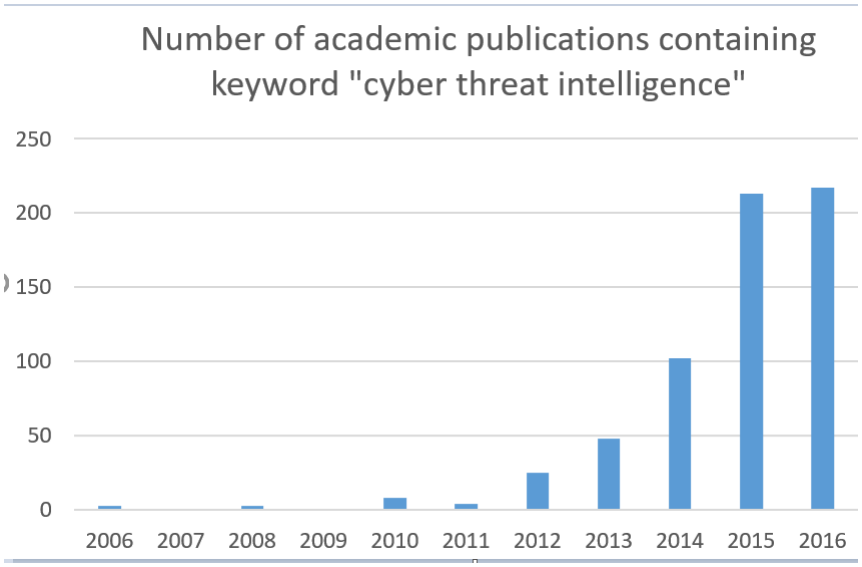


Figure 1.1: Number of academic publications containing keyword "cyber threat intelligence"; Source: scholar.google.com

Putting this definition together, Gartner defines threat intelligence as [8]:

> *Threat intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard.*

Gartner defines threat intelligence in the context of IT infrastructure.

Analogy can be used to better understand *threat intelligence* in the context. Let's define three types of threats:

- Known Knowns
- Known Unknowns
- Unknown Unknowns

Unknown Unknowns describe a situation where the organization is not aware of threats that it is facing. Let's imagine a situation where the shop is going to be robbed by a thief waiting outside. There is no awareness of the threat, neither of the context when and how it is going to happen. Known Unknowns describe a situation where there is threat awareness but it is not known, when and where it happens. Using the same analogy, blackmailing letter was received, knowing the shop will be robbed but it is not known when and how. Preparation is possible but efficiency is limited. Known Knows

describe on the contrary situation where there is threat awareness and context is known. Using the same analogy – information has been received from the police that there will be an attempt on the robbery of the shop in the night by pick-locking the back entrance.

Moving from Unknown Unknowns to Known Knowns is a threat intelligence [1]. This will allow a user or an organization to better prepare for cyber threats and improve their security posture. In cyber security, most of the organizations are aware of threats but they lack contextual knowledge about it. This is due to the fact that currently attackers are using more sophisticated methods and exploiting zero-day vulnerabilities [9, 10]. These types of attacks are usually done by botnet campaigns that are leveraging the same approach for multiple targets, also known as un-targeted attack [11]. If it is known, counter-measures can be prepared to protect the asset against the threats before they happen.

Most of the security experts share the same belief that Threat Intelligence is only and only if that information is actionable [2]. Actionable means that information value gained from threat intelligence can be used to reduce or eliminate the risk of successful attack [12, 13]. In the shop robbery imagined situation, a stronger lock and security cameras will be bought to reduce motivation and efficiency of the threat actor. In the cyber security field, threat actor can be prevented from malicious attempts by blocking him on the perimeter or reduce the detection time in case of insider threats. The insider threat is such a threat that comes from the inside of the organization. That includes accidental, negligent or malicious behavior by the employee [14].

To better understand what this thesis is focusing on, subtypes of threat intelligence need to be defined.

1.1.1 Subtypes

Any information that can inform decision should be considered as threat intelligence [1]. From reading news informing about hackers exploiting new security flaw to a colleague informing about raised number of phishing emails. Based on the way how this information can be handled and who is the aimed consumer, several subtypes are defined:

- Strategic Threat Intelligence
- Operational Threat Intelligence
- Tactical Threat Intelligence

1.1.1.1 Strategic Threat Intelligence

Strategic Threat Intelligence is a high level information consumed by non-technical people like board directors or chiefs information security officers(CISOs)

in the organization. It focuses on long term use. This can include current attack trends, motivations, financial impact and what type of organizations are being targeted from academic institutions to financial organizations like banks. This information might include who is likely to be the attacker including hacker groups, hacker activists or governments and how well resourced they are. This can drive organization decision regarding security countermeasures and areas with heightened focus. Currently, strategic threat intelligence is shared in a form of reports, briefings or conversations. However, strategic threat intelligence can be gained with data-mining sites that are used by threat actors, commonly dark-net or deep-net websites [15, 16].

1.1.1.2 Operational Threat Intelligence

Operational Threat Intelligence is a high level information with focus on short-term usage. It is consumed by security managers or threat hunting leads to identify current risks and threats. It includes information about upcoming attack including vectors like who, how and when it is going to launch an attack. Operational threat intelligence is mostly possessed by national governments as it requires extensive monitoring of hacker groups. However, organization might get such information if an attack is prepared by hacking activist and information is publicly available. Annual report from security vendors, security experts analysis or anti-virus companies statistics are all forms of operational threat intelligence. It can be, however, gained with manual analyst work scanning and infiltrating various hacking forums, including dark-net or deep-net. As well as strategic threat intelligence, operational threat intelligence can be gained using the same mining techniques [15, 16].

1.1.1.3 Tactical Threat Intelligence

Tactical Threat Intelligence is often referred to as tactics, techniques and procedures (TTPs). It has multiple subtypes itself, which is in threat intelligence community often referred to as *pyramid of pain.*(see Figure 1.2) [17]. On the contrary it is a low-level information consumed by incident response teams and security operations center analysts. Top 3 levels of the pyramid informs about tactics, techniques, procedures and tools that attackers have used before in another organization. It is likely that same approach will be used again. In hypothetical scenario, unspecified hacker group has used malicious software tool and exploited vulnerable web server in order to gain access. Privileges have been escalated, malware downloaded from remote command and control server and encryption of the data have been launched. Bottom 3 levels of the pyramid describe data that have been leveraged in the research. It includes IPv4 addresses, fully qualified domain names and uniform resource locators(URLs) that are considered as malicious and have been present in malicious attacks before. This is in computer forensics also known as *indicator of*

compromise [18–20]. Tactical Threat Intelligence is collected in various ways. Honeypot is a device that simulates activity of other systems like web, email or file server [21]. Cybersecurity researchers examine files collected by these sensors and if a malware is found, signature is created. Emails and URLs of webpages are investigated either manually from a honeypot catch or using scanners to look for sites that are compromised by malware or used directly by threat actors as phishing websites [22]. Another form of sharing such data is community driven sharing where users can submit, verify and share cyber threat intelligence data. Descriptions next to the *pyramid of pain* are expressing how difficult it is for the attacker to use evading technique to change her indicator of compromise.

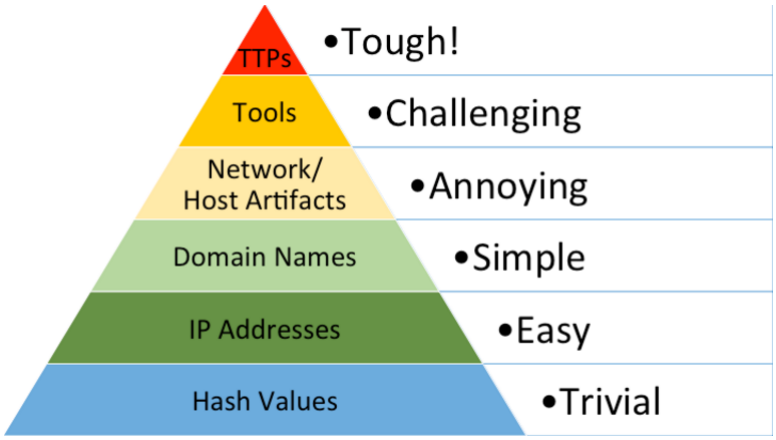


Figure 1.2: Pyramid of pain describes different types of indicators of compromise and the level of effort for the attacker to modify its indicators of compromises

Problems that had to be addressed in Tactical Threat Intelligence are detailed in the following text.

1.1.2 Sources

Where does the threat intelligence come from? Every source that can provide information that informs decision is considered as potential source of threat intelligence [1]. Based on the high-level location of the source, several source types are defined [23, 24]:

- Internal Threat Intelligence
- External Threat Intelligence

1.1.2.1 Internal

Internal threat intelligence is an information that comes from an experience of the organization itself. It combines knowledge of what is protected and what attacks happened in the past to create an actionable intelligence that will help organization to better protect against the same threat in the future. Starting from immature sharing via email up to mature solutions where such an info can be collected and managed centrally.

1.1.2.2 External

External threat intelligence is on the contrary an information coming from external sources. These data are called *threat intelligence feeds*. Providers of these feeds are government agencies, security vendors or community driven platforms. This thesis focuses on leveraging community driven information. These are also known as open source feeds which doesn't require paid subscription like most of the commercial feeds managed by vendor or agency. Most of the open source community feeds offer only data from the bottom 3 levels of the pyramid of pain (see Figure 1.2). Therefore in this thesis it was not possible to utilize threat intelligence from the higher levels.

1.1.3 Data representation

Data representation describes how threat intelligence data are categorized and several subtypes are defined:

- Structured
- Unstructured

These subtypes also define complexity of ingestion and utilization. Natural language or any form of unstructured data is difficult to fetch, parse and therefore utilize in automatized manner. On the other hand, threat intelligence provided in structured format is more suitable for automation as rules for their collection and storage can be built.

1.1.3.1 Structured

Structured threat intelligence is data that is formatted in a structured manner. Threat intelligence providers are using various formats to describe indicators of compromises (IoC) [25]. Most common types are file hashes, also known as signatures and reputation data on domains and IP addresses that have been associated with malicious activity. From an experience, most used formats are JSON, CSV or XML. MITRE organization has tried to standardize a way for such data using formats as CyBOX or STIX [26]. These formats have ability to describe not only indicators of compromise but also more complex

threat intelligence as tactics, techniques, procedures and tools (see Figure 1.2). However, they are not heavily used in community driven feeds as their format complexity makes it difficult to automatically create, share and consume this data.

1.1.3.2 Unstructured

Unstructured threat intelligence is data that does not follow any structure or convention. It includes email communication, twitter posts or community forum posts. It is a complex problem to parse unstructured data as it contains informal language and structure. Most of the strategic and operational threat intelligence is unstructured. It is due to the nature of the strategic and operational threat intelligence and its focus.

This thesis was focused on utilizing *structured* threat intelligence as the primary and only source. Utilization of the unstructured threat intelligence is a complex problem that is outside of scope of this thesis. However, attempts to collect certain type of unstructured data have been made and reader can find them here [15, 16, 27].

1.2 Threat Intelligence cycle

Threat Intelligence cycle is a model that is commonly used when building an effective threat intelligence program.

It is important to note that various books and security experts may have a different naming convention for the threat intelligence cycle [1, 22, 23]. After the research, selected naming convention was chosen as it is the most suitable representation for this thesis(see Figure 1.3) [1].

This model has 5 main areas that help to demonstrate the process necessary for building an effective usage of threat intelligence:

- Requirements
- Collection
- Analysis
- Production
- Evaluation

1.2.1 Requirements

First stage in the cycle is definition of requirements and goals that are going to be achieved by using threat intelligence. Following steps in the threat intelligence cycle are based on the requirements that are defined in this step.

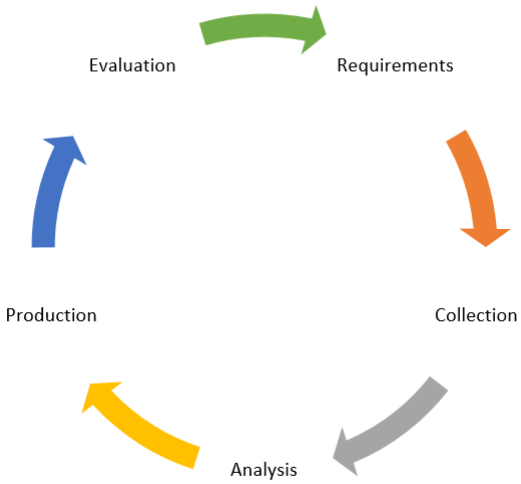


Figure 1.3: Threat Intelligence Cycle and its stages [1]

Planning on how these data will be leveraged also takes place in this stage. This is usually defined by the security team leaders or organizational needs. In this thesis, requirements are to design a correlation algorithms for selected platform to enhance detection capabilities of that platform utilizing threat intelligence data. These requirements transform into a set of use cases that are made of correlation searches. Furthermore, author selected scalability as a requirement to enable scalable development based on this thesis. Architecture and design must be done in a way allowing further enhancements in the future.

1.2.2 Data collection

Second stage of the cycle is data collection. In this phase threat intelligence sources needs to be selected to fulfill defined requirements. This thesis focuses on threat intelligence that is reusable and publicly available. As *internal* threat intelligence contains confidential data, *external* threat intelligence was leveraged. This consists of multiple phases:

- Threat intelligence feed selection
- Fetching

In the *threat intelligence feed selection* part must be selected what feeds need to be collected. They need to be collected from various open-source feeds because private or commercial threat intelligence feeds required paid subscription. Therefore they were not leveraged in this thesis. It is important to note that private feeds usually offer feed with higher quality as they have a

team of analysts and threat hunters that collect, submit, analyze and validate data. Providing such feed is their primary business. There are many open-source feeds but blind selection would be a mistake. It needs to be known what sort of data is collected in such a feed and how it is maintained. Is the feed removing data that is no longer valid? How often and how many data is added? Is data in the feed overlapping with another feed? These are the questions that need to be answered before it is decided to collect the data from this particular feed. Another important factor is a knowledge of what kind of threat intelligence data is the feed gathering. This is again reviewed against the requirements that were defined. Reader can find further research on open-source feeds in the following publication [28].

In the second phase, a fetching process needs to be setup. This process will regularly download and update content from various feeds and make sure that data in our platform is up to date. It is usually done using simple HTTP methods or via Application programming interface (API).

The output of this phase is raw threat intelligence data collected from selected threat intelligence feeds. This output serves as input to the following stage – analysis.

1.2.3 Analysis

In the analysis stage raw data is being collected from the selected threat intelligence feeds. Collected data feeds from various sources are in different formats as JSON, XML, CSV, STIX or in a custom structure. In order to leverage all of them in one platform, they need to be parsed and modeled into a common format in order to be used later in the production phase. Indicators of Compromise (IoC) must be extracted from the threat feeds and normalized. One goal of the normalization process is to organize newly collected data in a way that only unique indicators are collected and stored in order to avoid duplicates. Indicators of Compromise, like IPv4 addresses must be stripped of values that are irrelevant as private network addresses defined in RFC 1918 [29]. In this phase custom tags are applied to different sources to apply context that will be used in the production phase. There is a contextual difference if the mentioned indicator in the feed represent IP address of command and control server that malware uses to get commands or if the IP address was reported for a phishing campaign that was delivered from this particular address. In the case of MD5 hashes, they could represent hash of the certificate that was used in the malicious campaign as well as hashed signature of the malware. Custom confidence must be added to the processed threat intelligence to represent trustworthiness of the selected data feed provider. Furthermore, analysis stage serves for threat feeds enrichment. That means taking an existing indicator with all it is properties and enrich it with new knowledge or even derive a new indicator based on the original one. For example, an IPv4 address can be enriched with the geo-location. Specific en-

richment methods are applied based on the maturity of the tool or platform used in the analysis stage. It is further detailed in the section 1.6.2. This analysis phase is essential for the functionality as it converts raw data into a normalized set of threat intelligence data that will be used in the production stage.

1.2.4 Production

Production stage is leveraging threat intelligence data that has been defined, collected and normalized in the previous phases of the threat intelligence cycle. The goal of this stage is to fulfill requirements, implement the solution and utilize the data in the production environment.

1.2.5 Evaluation

Last stage serves for evaluation of the requirements defined in the first stage. It points what was done right and what could be done better. This regular feedback allows to continuously develop cyber threat intelligence program. Right after the evaluation stage the cycle is repeated using the new or adjusted requirements. In this thesis one full cycle has been done and evaluation stage is provided in the conclusion.

1.3 Security Information and Event Management

Security Information and Event Management, also known as SIEM is a tool used to collect all security relevant data in a centralized system in order to correlate events from various security devices to enhance detecting and monitoring capabilities. Gartner defines SIEM as [30]:

> *Security information and event management (SIEM) technology supports threat detection and security incident response through the real-time collection and historical analysis of security events from a wide variety of event and contextual data sources. It also supports compliance reporting and incident investigation through analysis of historical data from these sources. The core capabilities of SIEM technology are a broad scope of event collection and the ability to correlate and analyze events across disparate sources.*

Advantages of SIEM tool include log management, data aggregation, ability to correlate data, alerting and data visualizations capabilities. However, this advantages comes with a high price tag.

Goal of this thesis is to enhance threat detection capabilities utilizing threat intelligence, therefore security event and incident management tool needs to be leveraged to correlate threat intelligence data across disparate sources. SIEM tool is acting in *production* and *evaluation* stage.

1.4 Threat Intelligence Management Platform

Threat Intelligence Management Platform is a knowledge base to store threat intelligence information. It has the ability to automatically collect, aggregate and normalize threat feeds. This platform is acting in *data collection* and *analysis* stage of the threat intelligence cycle.

This platform can be represented with a vendor-based solution which then provides external API access to the platform to access threat intelligence data. Commercial platforms often have integration with popular security information and event management tools that are largely used in enterprise security. The advantage is that commercial solutions have large teams of security experts that can do advanced analytics and maintain high quality of the threat intelligence feeds gained internally or via private paid feeds. However, the disadvantage of such solution is a high price and low visibility into the *data collection* and *analysis* stages of the threat intelligence cycle that are outside of the user control. Such vendors include *Recorded Future*[1]

However, in this thesis, open-source solutions were leveraged to suit requirements for Threat Intelligence Management Platform. This allows further research based on an open platform which may lead to the enrichment of the cyber threat intelligence community.

1.5 Utilization

Goal of this section in this thesis is to analyze actionability of the threat intelligence and analyze what use cases will be utilized. They are later implemented in the form of correlation algorithms that will enhance the detection capabilities.

Currently, there are commercial or open-source solutions that solve one or two stages of the threat intelligence cycle. However, for an organization to fully benefit from threat intelligence, a full cycle should be implemented. Having properly collected threat intelligence data without context or proper actionability is limiting the efficiency if any. Actionable means that information value gained from threat intelligence can be used to reduce or eliminate the risk of successful attack [2, 12, 13]. Organizations may tie and integrate these stages together, however, it is a complex problem that requires cooperation of multiple teams and resources, human or financial. Only very mature organizations have fully implemented a threat intelligence program.

In the research and survey done by SANS organization [2], the reader can find more information about problems that are organization facing in the threat intelligence field. On the question, "What is holding your organization back from achieving integrated CTI capabilities?"[2], organizations have

[1] https://www.recordedfuture.com/

[2] CTI means Cyber Threat Intelligence

responded accordingly (see Figure 1.4 [2]). This can be interpreted as that organizations have the interest in having cyber threat intelligence program. Because there is not "one fits all" solution or commonly used standards, it makes the problem of integration complex. Therefore it requires knowledge across different fields supported by quality staff. More information about who and how is using cyber threat intelligence can the reader find here [2].

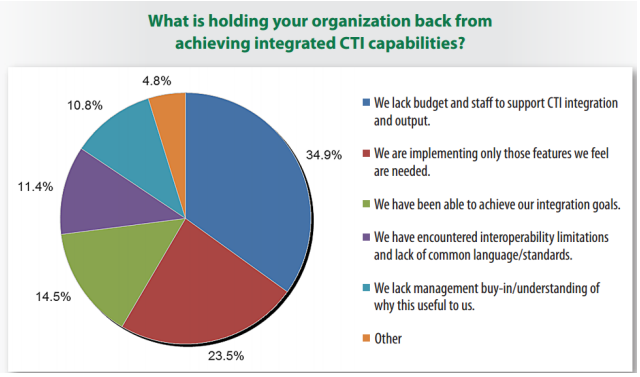


Figure 1.4: SANS - Limitations in the cyber threat intelligence implementation [2]

This thesis wants to bring a concept that is based on tools that are open-source or free for certain period of time and show how they can be integrated and utilize a specific set of use cases to make the threat intelligence actionable. Organizations may take this concept and enhance it with use cases that are critical for their operations and business.

Use case may have different meanings in different IT fields, in this thesis *use case* is understood as a logical, actionable and reportable component of the security information and event management system (SIEM). It can have a form of a rule, alert or visualization.

Use case selections for this thesis were consulted with the global security experts from Splunk during their video-conferences, security researchers presenting on BlackHat 2016 conference and security experts from Accenture Security that are doing security assessments in all areas of IT Security. That includes penetration testings, hardware and software security, incident response, security monitoring, advanced analytics and threat intelligence itself. These use cases address the current issues and trends in the cyber security industry while leveraging threat intelligence data. These use cases are put into three categories:

- The Onion Router traffic
- Ransomware
- Phishing

1.5.1 The Onion Router traffic

The Onion Router, also known as TOR is a project that provides its users anonymity online [31]. It is used in countries where the internet is censored to avoid detection or to access sites that are normally blocked.

However, it is often used by threat actors to hide their tracks. Furthermore, it can be used by malicious software in different stages of the attack. In recent years, botnets have been found to be controlled through TOR network [32,33].

It is not only these external threats as a reason for monitoring and detecting TOR traffic. Usage inside the organization should be forbidden as it can serve for data exfiltration or viewing a malicious content.

TOR itself is not an indicator of a threat, but in the context of the organization, outgoing traffic through the onion router network could indicate a rogue employee or a malicious code trying to communicate with its command and control server. It is not unusual that it is against the company policy to use TOR on their laptops. Therefore such communication should be always detected and investigated.

Incoming traffic from TOR network can be the indication of malicious attempts in a phase of reconnaissance but also in delivery, exploitation, installation or command and control [3, 34, 35](see Figure 1.5).

It is extremely difficult to distinguish TOR traffic from normal traffic as it also uses standard ports used for web communication [31].

1.5.2 Ransomware

Ransomware has become popular among attackers as a method to collect profit [36]. Ransomware is a malicious software that is usually delivered via exploit kits or phishing campaigns and leverage vulnerability in the software to run it is own malicious code and encrypt files on the victim machine [37]. To decrypt files back, the attacker demands a ransom paid in the untraceable currency as for example bitcoin [37, 38]. Some types of ransomware have even used TOR to communicate back to the server of the attacker [37]. In the recent years, serious organizations like hospitals or universities have become victims of some ransomware, which resulted in financial damage [39].

Organizations might be using a various software solution such as anti-virus or next generation firewalls to protect against those threats, however, it seems that it is not enough [10].

1.5.3 Phishing

Phishing is a fraudulent attempt to obtain sensitive information as usernames, passwords or credit card details. It is usually delivered via email or websites and it uses social engineering to trick the victim. Fake content is trying to convince the user to click on the link with malicious content, open the attachment or to enter his private information [21].

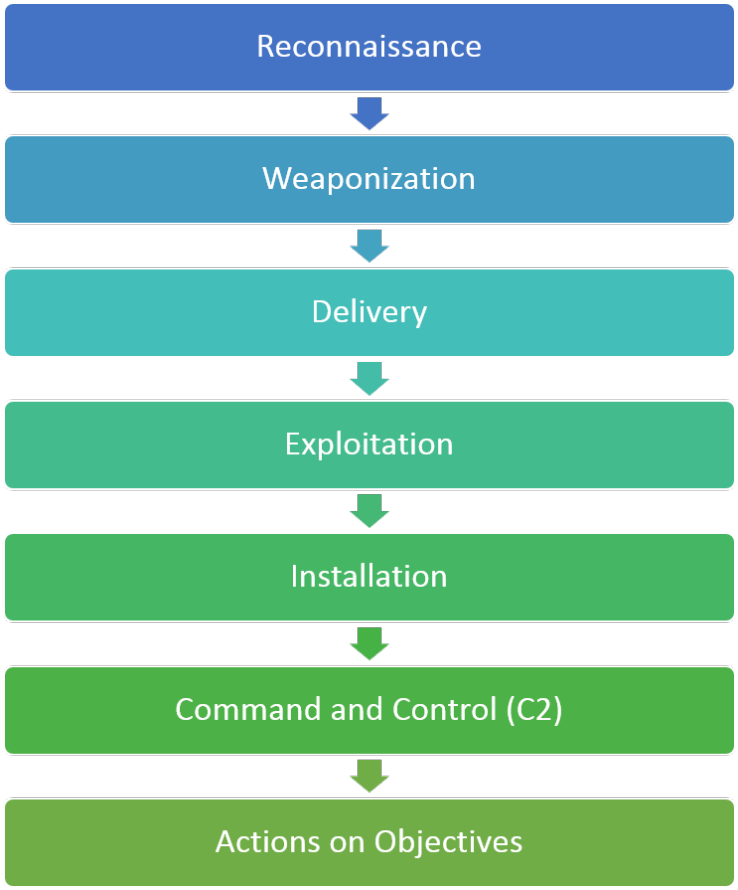


Figure 1.5: Cyber kill chain according to Lockheed Martin [3]

These use cases were selected as phishing campaign is often the point of the infection. It may deliver ransomware or any malicious software.

Some organization use external commercial tools that use their own threat intelligence feeds to inform about potential risky sites or links. In recent years, even company like Google has implemented their anti-phishing filter that use threat intelligence to protect the user against this threat. Early detection of users that have visited such site can reduce further damage and minimize potential risks.

1.6 Platform proposals

In order to utilize and integrate the full threat intelligence cycle, Security Information and Event Management platform and Threat Intelligence Management platform are selected. They play key roles in this thesis, as they offer capabilities that are described in the threat intelligence cycle. Threat Intelli-

gence Management Platform has the capability for *data collection and analysis* stage of the cycle whereas SIEM tool can serve for the *production and evaluation* stage (see Figure 1.6). Principles used in the threat intelligence cycle are, however, not platform specific. Platform selection is present to enable design and implementation of those principles in the full threat intelligence cycle and to provide proof of concept that can be evaluated.

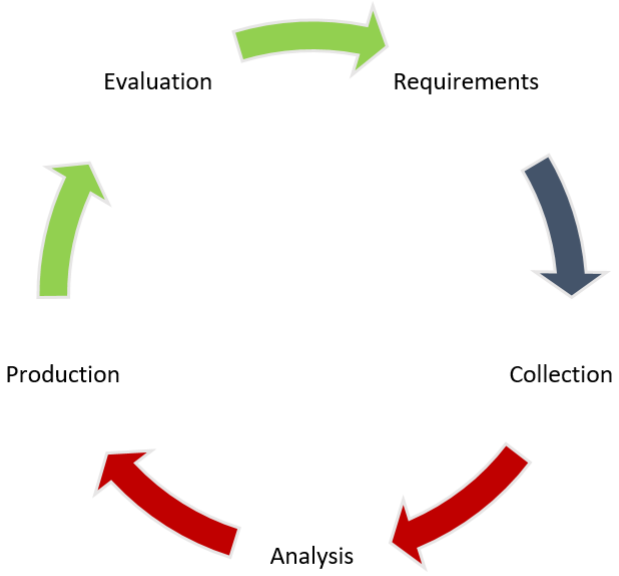


Figure 1.6: Threat Intelligence Cycle – stages highlighted in red are implemented in *Threat Intelligence Management platform* whereas stages highlighted in green are implemented in *SIEM platform*

1.6.1 Security information and event management

To accomplish what was defined in the *requirements* stage, a platform where correlation algorithms take place must be chosen. The selected platform must have data correlation capabilities. Data correlation capability allows to efficiently search data patterns across various data sources. It also enables to filter and transform this data and enable the creation of new correlation searches to extend its capability. Security Information and Event Management tools offer these capabilities and are used in organizations for monitoring and detection of security incidents.

As Threat Intelligence Management platform is used for data collection, normalization and enrichment, the selected platform must be able to integrate with such platform. Selected platform must handle big-data in real-time as

early detection is a key to reducing the damage [40]. The selected platform must be publicly available so further extension or development by the community can happen. Ideally, it would also provide first hands-on experience or documentation for users as security event and management tools are usually very complex platforms. To sum up, requirements for the selected platform are:

- Security Information and Event Management platform with ability to integrate with Threat Intelligence Platform
- Modular platform with ability to handle big-data
- Documented and publicly available

Principles designed in this thesis should be developed and used on a platform that is widely used in organizations to broaden the potential usage. As there are many SIEM tools on the market and it would be a research topic itself to evaluate each one, Gartner research in Security information and Event management platforms is leveraged. Strategy for platform selection was to start with the leading platforms in SIEM according to Gartner [4, 41](see Figure 1.7). Secondly, requirements for the selected platform as well as listed advantages and disadvantages are evaluated.

Main leaders for this industry are *Splunk, IBM, LogRhytm* and next runner-ups are *HPE and Intel Security* [4]. All of these leading SIEM tools are commercial products, however, only *Splunk* offers publicly available free 60-day full feature version of their tool that can be extended for developers [42–44]

Regardless, both IBM QRadar and Splunk offers modular structure, which allows developers building new use cases and integrations. Documentation is available for both platforms, however documentation for Splunk is much richer based on the consultations with security experts that have both experiences with Splunk and IBM QRadar. From the *Splunk, IBM, LogRhytm* tools which are industry leaders, Splunk was the preferred platform. Based on the author experience and the available documentation, it allows him to efficiently design architecture, integration and new correlation searches. Splunk is also able to handle big-data in real-time.

Therefore, *Splunk* platform was selected as *Security Information and Event Management platform* and is further leveraged in this thesis. Splunk user interface for data interaction is mainly composed of the search bar, sidebar, and events (see Figure 1.8). Search bar serves for writing commands that are then translated to REST API calls to retrieve, filter and transform data into statistics or visualizations. Sidebar displays extracted fields with their corresponding values and events show either the resulting events from the search bar or statistics depending on the nature of the used command. Timeline simply puts the event timestamps into visual form.

Figure 1.7: Magic Quadrant for Security Information and Event Management displaying where SIEM platforms stands according to their ability to execute and the completeness of vision [4]

1.6.2 Threat Intelligence Management Platform

Security Information and Event Management tool was selected for its detection and monitoring capabilities that can be extended using threat intelligence. Selection of platform that can handle collection, storage, normalization and enrichment of the threat intelligence feeds is essential to fulfill the threat intelligence cycle. There are several open-source platforms with these abilities [45]. After the consultations with the experts in the threat intelligence industry and further research [46], *Collective Intelligence Framework*, also known as CIF was selected. It is an open-source framework developed by REN-ISAC

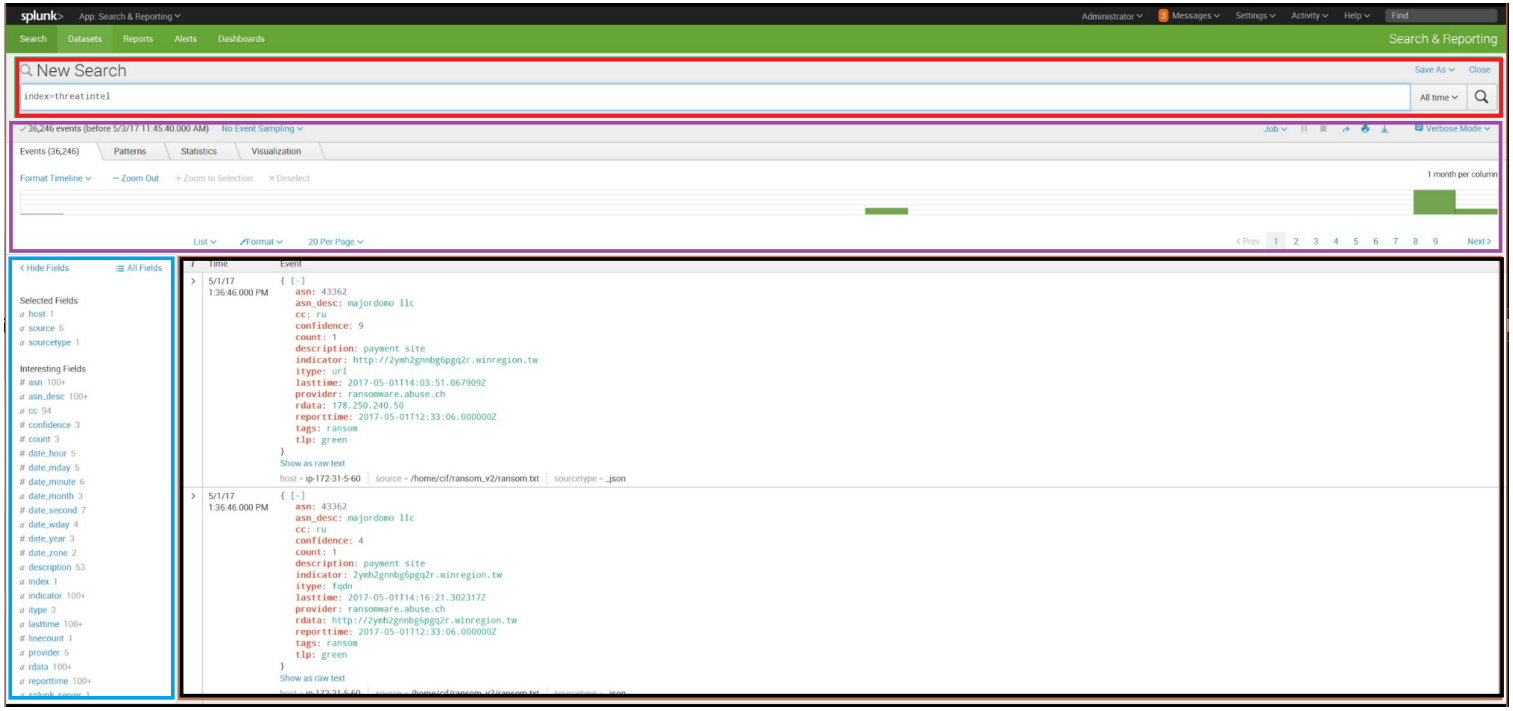


Figure 1.8: Splunk user interface composed of the search bar (red box), sidebar (blue box), events (black box) and timeline (purple box)

and is used to collect, store, normalize and enrich threat intelligence data. Other open-source solutions include *Malware Information Sharing Platform*[3] and *CRITs: Collaborative Research Into Threats*[4]

Collective Intelligence Framework is out-of-box shipped with enrichment capabilities. To understand what type of enrichment is taking place, source code have been reviewed and analyzed.

For the threat intelligence feeds and requirements defined in the threat intelligence cycle, four types of enrichment took place:

1. Enrichment of the URL
2. Enrichment of the FQDN
3. Enrichment of the Subdomains
4. Enrichment of the IP

Enrichment is a process where a certain indicator is taken and new indicators are derived based on the existing one. This is best explained on a practical example:

This is a common URL *http://fit.cvut.cz/student/magistr/DP-a-SZZ* that can be used as an indicator if a phishing or malicious activity was reported on this URL. How can be this URL enriched ? If assumption are made, it is possible to say that if this URL is malicious, then the whole FQDN[5], which is *fit.cvut.cz* is malicious. The same assumption is made for the subdomain. If fit.cvut.cz FQDN was malicious, is *cvut.cz* malicious as well ? At this point two FQDN are available. They can be further enriched by doing a *nslookup*[6] to find IP addresses that are hosting those FQDN. Another assumption is made that if the fit.cvut.cz was malicious, then the whole server with particular IP address hosting this domain is malicious.

However, for every level of assumption, confidence must be lowered because it is still only assumption which might or might not be correct. Therefore for every level of assumption that derive the original indicator, confidence is lowered.

1.7 Test environment

Test environment serves as a validation tool for correlation algorithms. Considering the fact that solution should improve security posture and threat detection capability, usage of data that is as close as possible to the data that

[3] https://github.com/MISP/MISP
[4] https://github.com/crits/crits
[5] fully qualified domain name
[6] https://technet.microsoft.com/en-us/library/bb490950.aspx

will be used in production environment is the desired approach. Therefore two disjunct approaches are proposed.

First approach will be integration of test data that is captured in production environment. This data will require approvals to use and must be anonymized to prevent disclosure of sensitive information. This will have a positive impact on evaluation and validation of the solution as captured data are from the production environment.

On the other hand this can have negative impact on certain use cases as they might not be present in the captured data. Therefore second approach will leverage creation of scenarios that mimics behavior of the attacker. These scenarios are described in the Testing chapter. Data is simulated and used in conjunction with the first approach.

With this test environment this thesis can be validated as well as evaluated the impact in the real-world scenarios. For these scenarios, laboratory environment is designed (see Figure 1.9).

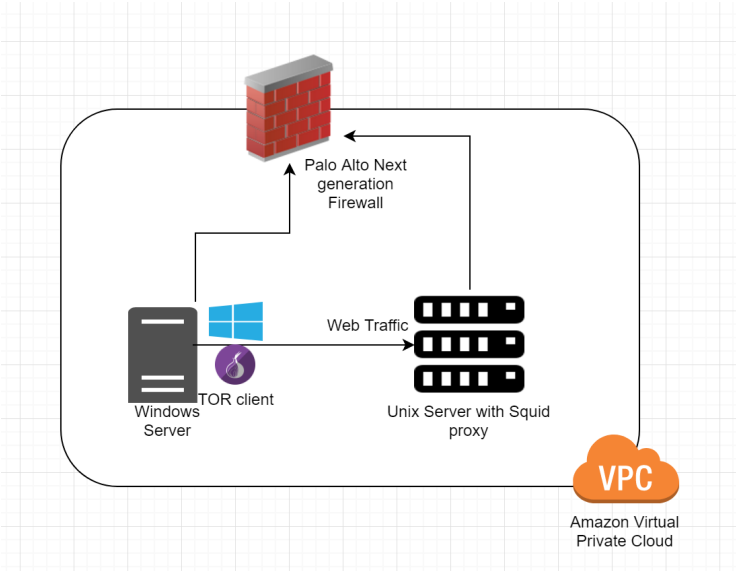


Figure 1.9: Architecture diagram of the lab environment

Chapter 2

Design

In the analysis chapter *threat intelligence* and its types were defined. Utilization of threat intelligence, meaning making threat intelligence actionable, includes integrating each stage of *threat intelligence cycle* together. This chapter serves for designing every stage of that threat intelligence cycle to solve the complex problem of utilization threat intelligence (see Figure 1.3). To achieve that, platforms that play a key role in the cycle have been selected in the analysis chapter. Use cases are part of the production stage of the threat intelligence cycle. Therefore they are also designed here in this chapter. Architecture and design from this chapter will be then leveraged in the realization chapter.

2.1 Integration

In order to fulfill requirements and utilize threat intelligence, Threat Intelligence Management Platform and Security Information and Event Management must be integrated. This integration is designed in this chapter. According to Gartner, an integration is defined as [47]:

> *Integration services are detailed design and implementation services that link application functionality (custom software or package software) and/or data with each other or with the established or planned IT infrastructure.*

Threat Intelligence Management Platform suits for *data collection and analysis* stage of the cycle whereas SIEM tool serves for the *production and evaluation* stage. Requirement stage is outside of any platform. Integration of those two platforms ties together the threat intelligence cycle (see Figure 1.6). That means the goal of the integration is to tie stages highlighted in red (Data collection and Analysis) that are done in the Collective Intelligence Framework with the stages highlighted in green that are one in the Splunk

platform (see Figure 1.6). An indicator of successful integration is that threat intelligence data is being automatically ingested in the Splunk platform and can be leveraged in the *production* stage.

2.1.1 Architecture

In order to design good architecture of the integration, goals and requirements must be defined. In this case, the goal of the integration is to get collected, stored, processed, normalized and enriched threat intelligence feeds into the SIEM tool. Main requirements are automation, scalability, adjustability, extensibility, performance and reliability. Automation means that newly collected threat intelligence feeds are automatically updated and pushed to the SIEM platform. Scalability means that this design is valid even if the number of threat feeds or data from various sources will significantly raise in the future. Adjustability takes into consideration that different organization may have a different infrastructure where strictly limited architecture may not be suitable. Adjustability ensures that the architecture can be slightly adjusted to fulfill needs of the organization. Extensibility takes into consideration that the organization might try to extend platforms by new threat intelligence cycles or other development on both platforms. That way, it is ensured that proposed architecture will not be an obstacle or bottleneck that needs to be redesigned in the future. Performance is important from the production perspective. Using best practices defined by the vendors or developers ensure maximum possible performance. Reliability ensures that for example in the case of for example network failure, automated integration process will continue where it stopped without having to manually intervene. To sum up, requirements for the proposed design are defined as:

- Automated
- Scalable
- Adjustable
- Extensible
- Performance
- Reliable

2.1.2 Proposed Design

Creating a good architecture requires a good knowledge of the both involved platforms to evaluate possible advantages and disadvantages. Therefore it is an important to note that proposed design in this thesis have been changed and altered after the consultations with Splunk engineers and Splunk architects

on their video-led conferences. Main reasons for that were to not make any sacrifices on the architecture requirements. It resulted in the following design (see Figure 2.1).

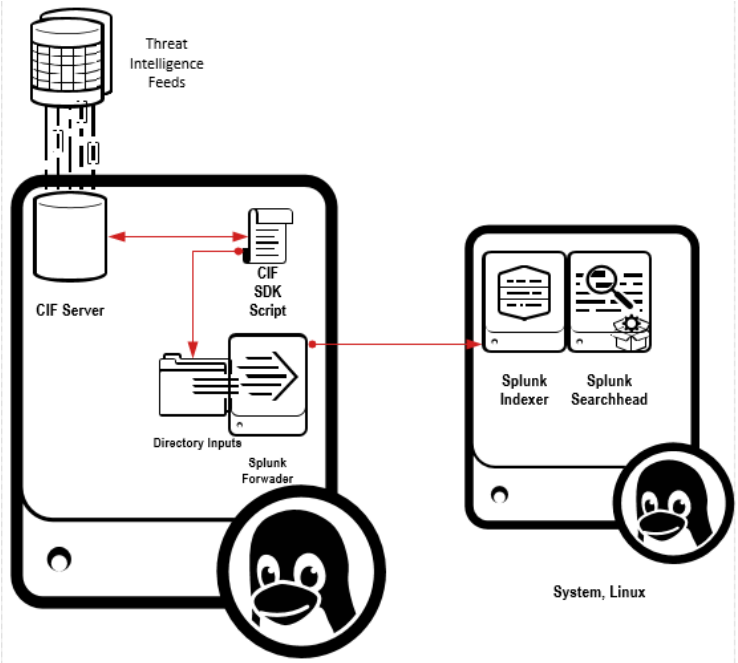


Figure 2.1: Architecture diagram of the proposed design

In the proposed design, two machines with the UNIX operating system are present. However, this is not a design requirement. Reasons for using two distinct machines were to better illustrate that these platforms can reside on different machines or different networks which support adjustability.

The machine on the left (see Figure 2.1) is running Collective Intelligence Framework (CIF Server) and collect, store and parse data from the threat intelligence feeds (see Figure 2.2).

On the same machine (see Figure 2.1) runs software development kit for collective intelligence platform (CIF Client). CIF Client pulls data from the CIF Server using its API (Application Programming Interface). This allows having multiple CIF Clients in multiple locations that can pull data from the same CIF Server. At this point, data is being collected and processed and can be pulled via API. Splunk platform is running on the second machine. Both machines reside in the Amazon Web Services cloud (see Figure 2.3) which requires extra knowledge to configure that is outside of the scope of this thesis. It is, however, not an architecture requirements and these machines can reside on any infrastructure.

Next step is to design how data that can be pulled from CIF Server are going to be sent to Splunk platform in an automated manner. Splunk has

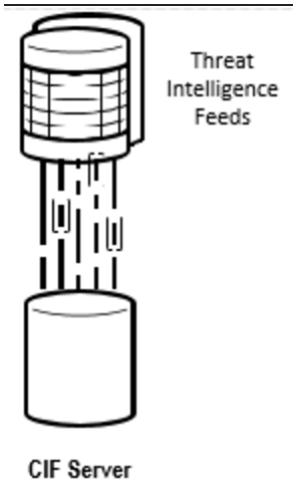


Figure 2.2: Part of the architecture diagram of the proposed design

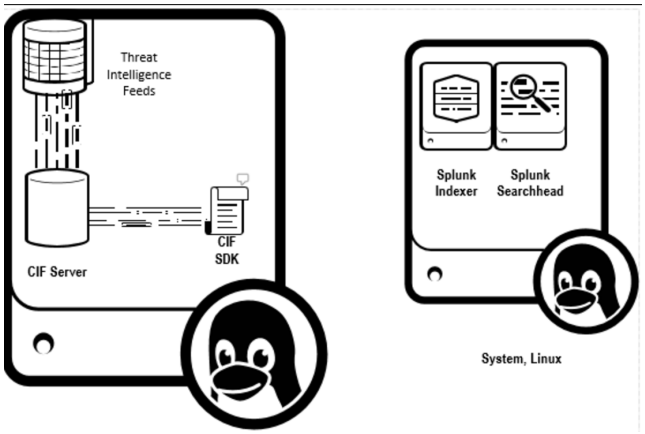


Figure 2.3: Part of the architecture diagram of the proposed design

many ways how to ingest data, one of it is listening on the certain port (see Figure 2.3). Eventually, data could be automatically pulled using script via API and sent directly to Splunk platform on this port. However, there is a better way to do this and it is recommended as a best practice from Splunk vendor itself. This includes a lightweight agent, called *Splunk Universal Forwarder* that collects the data by monitoring files or folders. This data are compressed and sent to Splunk platform. In the case of the larger environment which consists of multiple indexers, it can load-balance the data across all indexers. It has also the ability to cache, monitor and acknowledge if data has been correctly received in the case of the network failure.

In order to use *Splunk Universal Forwarder*, the directory containing pulled data is created. The script, running as a cronjob every hour is created and it

pulls data from CIF Server using API and appends new results to a file (see Figure 2.4). This file is then being monitored by the universal forwarder and forwards data to Splunk platform.

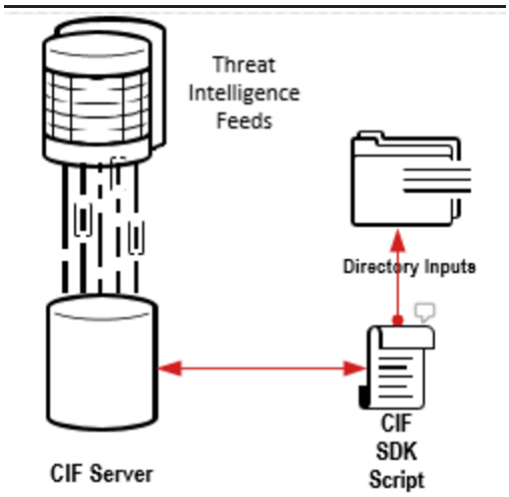


Figure 2.4: Part of the architecture diagram of the proposed design

This concludes the proposed design which implementation is described later in the realization chapter. Final design is shown above (see Figure 2.1). It fulfills all requirements defined in the architecture subsection.

2.2 Usecases

At this point integration took place and the threat intelligence cycle is fully tied. That means that the red stages (see Figure 1.6) are inter-connected with the stages highlighted in green. Therefore selected use cases addressing problems analyzed in the sections 1.5.1 - 1.5.3 can be designed. These use cases will define and represent requirements and drive the following stages –*Data collection, analysis and production.* These three stages are designed for every use case.

2.2.1 TOR Traffic

TOR Traffic has been analyzed in the section 1.5.1. In this section, full threat intelligence cycle with specific use cases that leverage threat intelligence are designed. Their goals are to enhance the organization capability to detect TOR traffic, in both outgoing and incoming traffic. Detection allows and leads to the investigation and elimination of the risk.

2.2.1.1 Requirements

Leveraging open source threat feeds that collect TOR nodes, such communication can be detected and alerted.

Two use cases for TOR traffic are designed:

- Alert - Detection of TOR traffic originated in the organization (see Figure 2.5)
- Alert - Detection of TOR traffic with destination in the organization (see Figure 2.6)

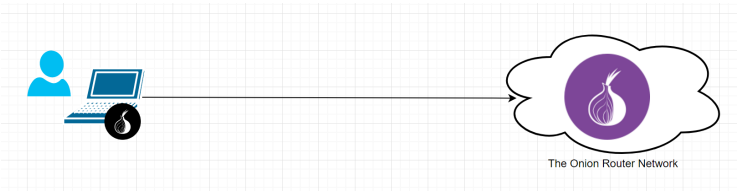


Figure 2.5: TOR traffic originated in the organization

Figure 2.6: TOR traffic with destination in the organization

For the detection of such traffic, logs from the perimeter firewalls must be present in the SIEM tool. The onion router is using TCP protocol. Therefore if the source IP in the network communication is coming from inside the organization, destination IP is on a list of known TOR entry/exit points that indicate outgoing traffic. Because outgoing TOR traffic originated in the organization, an alert should be raised. To allow the security operation center analyst to investigate this more efficiently, statistics on transferred bytes were leveraged. A high number of outgoing bytes originated at one source IP inside the organization to the multiple TOR entry/exit nodes can indicate data exfiltration. Incident response team might contact the user or temporarily cut off the asset from the network before the incident is properly investigated. In the second case, to detect traffic that destinate to our network, various scenarios can happen. It can be legitimate traffic but also an attacker in various stages of attack according to Lockheed Martin Cyber Kill Chain [3]. It would be misleading and trigger lots of false positives if an alert is raised for any

incoming traffic from TOR. However, such behavior should be detected, reported and visualized so the security operation center analyst can investigate the context of the traffic and do a proper incident response.

2.2.1.2 Data collection

Data are collected from open source threat intelligence feeds. For this use case feed with TOR nodes is essential. To select the best feeds, some statistical analysis must be done. These listed questions should be asked for every source:

- How often is selected threat feed updated?
- What is the context behind this feed?
- What is the overlap with other feed?

Fortunately, there is a community portal that is doing regular statistics on some open source feeds [48]. Five different feeds for TOR nodes can be found.

1. tor_exits by TorProject.org
2. iblocklist_onion_router by iBlocklist.com
3. et_tor by Emerging Threats
4. dm_tor by dan.me.uk
5. bm_tor by torstatus.blutmagie.de

However, importing all feeds would not be a right approach. Before accepting a new threat intelligence feed, these questions must be answered. The overlapping table that displays how many percent data of one feed can be found in another feed has been created (see Figure 2.7).

Threat Intelligence Feed	tor_exits_30d	iblocklist_onion_router	bm_tor	dm_tor	et_tor
tor_exits_30d		44.77%	45.38%	45.19%	45.57%
iblocklist_onion_router	13.54%		92.49%	92.39%	93.04%
bm_tor	13.55%	91.27%		98.34%	90.21%
dm_tor	13.47%	91%	98%		89%
et_tor	13.75%	92.34%	91%	90.40%	

Figure 2.7: Selected Threat Intelligence Feeds and their percentage overlap across each other

From the 2.7 it is obvious that four of five feeds have around 90% overlap. How are we going to determine which feed is the most suitable for our needs? Another set of statistics helped to understand and compare how these feed differ (see Figure 2.8).

Feeds 2.- 5. overlap each other in around 90%. That means they share approximate the same amount of unique TOR nodes (IPs). However, there is

Name	Number of unique IP addresses	Approximate number of added and removed Ips per 24h	Average update time in minutes	% of IPs added in last 30 day	% of IPs removed in the last 30 days
tor_exits_30d	2131	72	67	48%	64%
iblocklist_onion_router	7045	1000	1620	31%	93.20%
bm_tor	7139	3200	92	43.40%	97%
dm_tor	7147	2400	59	92%	98%
et_tor	7000	1200	1700	29.65%	91.68%

Figure 2.8: Comparison of Threat Intelligence Feeds properties

a significant difference in how often is data in the feed updated and rotated. In the *dm_tor* feed data is updated in average every 59 minutes and 92% of all IPs in the list have been added in the last 30 days. This shows that this feed is regularly managed and taken care of. This significantly raises confidence of the data that is in this feed. This is the reason why this feed was selected. As they overlap in around 90% data with other three feeds, these other feeds were not used as it is preferred to have quality over quantity.

The second selected feed (highlighted in yellow) is managed by the *TorProject.org* itself and it does have only 45% overlap with the first selected feed. In average it is updated every 67 minutes. This indicates good quality of this feed.

To sum up, the following two threat intelligence feeds were collected for the purpose of this use case:

1. tor_exits by TorProject.org

2. idm_tor by dan.me.uk

2.2.1.3 Analysis

Selected threat intelligence feeds were collected using Threat Intelligence Platform, namely – Collective Intelligence Framework that has the capability to collect, store, parse and normalize the threat intelligence feeds and then enrich them. Indicators that are extracted from these feeds are IPv4 addresses. Collective Intelligence Framework allows enablement of custom confidence for each feed as well as custom tags. For these feeds relatively high confidence was assigned due to the fact that both feeds are regularly managed and updated. One of those feeds come from the creators of the TorProject itself. Confidence nine out of ten was therefore assigned for both feeds. This is the best effort based and can differ based on the user experience. Recommendation was taken from the Collective Intelligence Framework e-book hosted on the git-hub wiki [49]. Indicators are sent to Splunk platform where they will be correlated with data from various sources. IPv4 addresses are enriched with the geo-location and the name of the provider, that is managing those autonomous system number.

Proposed time frame before updating indicators in the used platforms is 1 hour to keep performance benefit as well as the benefit of having the most recent data.

To fulfill use case requirements and correlate data with extracted indicators, data sources that are monitoring network traffic must be ingested into Splunk platform. This includes mostly all perimeter firewalls. In this thesis, Palo Alto firewall [50] has been leveraged as a source of network traffic logs. It is important to note, that the firewall should be configured to monitor start of the sessions.

2.2.1.4 Production

In this stage, correlation searches were designed. Natural language is used to describe the logical flow of the use case, whereas in the realization chapter technical details are described.

For the first use case, *Alert - Detection of TOR traffic originated in the organization*, matching alert rule has two logical conditions that need to be satisfied for use cases to trigger:

- Source IP belongs to the internal network and this includes data from all network devices across various vendors
- Destination IP is on the list of extracted indicators

For the second use case, *Report - Detection of TOR traffic with the destination in the organization*, matching rule is composed of two logical conditions:

- Source IP is on the list of extracted indicators
- Destination IP belongs to the organizational network

2.2.1.5 Evaluation

Evaluation of the results serves as feedback for further improvements. Proposed testing includes connections from the lab environment to the internet using TOR browser for outbound and connection attempts to the internal organizational network using TOR browser for the inbound use case.

2.2.2 Ransomware

Ransomware has been analyzed in the section 1.5.2. In this section, full threat intelligence cycle with specific use cases that leverage threat intelligence are designed. Their goals are to enhance the organization capability to detect ransomware activity during different stages of the attack according to the Lockheed Martin Cyber Kill Chain (see Figure 1.5).

2.2.2.1 Requirements

There are three vectors that can be observed with most of the ransomware. This is based on how the selected feed has it defined.

- Distribution of the malicious code
- Command&Control
- Payment

Most common type of ransomware is delivered through phishing email with a malicious attachment that exploits some vulnerability [37]. Inserted malicious code then tries to download the ransomware itself from the *distribution* sites [38]. Let's imagine a situation that the phishing email surpassed the spam and phishing filter and the user opened the malicious attachment which exploited some local vulnerability, downloaded and ran a malicious code. If the signature of the malicious code was unknown for the local anti-virus, it is the first vector where threat intelligence can step in and detect the connection to or from the distribution site. Therefore it can detect a ransomware delivery from the distribution site to the target machine.

The second vector is the discovery of communication to the command and control server. It is common that the ransomware will try to hide its communication among the regular traffic to avoid detection. Detection avoidance can be done through TOR network or by using encrypted protocols like HTTPS to communicate with the remote command and control server [35, 51]. Early detection in this phase (see Figure 1.5) [3] may lead to quick incident response and significantly reduce damage. Furthermore, more aggressive ransomware can not only encrypt the local machine but also attempt to propagate itself across the network [37]. Command and control servers are used for ransomware to "call home" and report infected machine and exchange encryption keys. It is not unusual that the attacker is waiting until a reasonable number of machines inside the organization is infected before delivering encryption keys to the target machines [37].

The third vector is a payment itself. Scared and feared users might try to silently pay the ransom. It is necessary for the cyber security team to know about such behavior to launch an investigation or possibly reduce the damage on other infected machines. Incident response team extracts indicators of compromise of the certain ransomware to find similar machines that are infected but the encryption process hasn't started yet.

To enhance the detection capabilities, network data must be correlated with the threat intelligence feeds. Ideally, the organization will use and ingest data from tools like Proxy or Web Gateway, DNS, Windows AppLocker, Windows Enhanced Mitigation Experience toolkit or anti-virus. In this thesis, proxy data have been leveraged.

Three use cases have been selected and implemented (see Figure 2.9):

- Detection of traffic from/to the malicious distribution site
- Detection of traffic from/to the malicious command and control server
- Detection of traffic to the malicious payment site

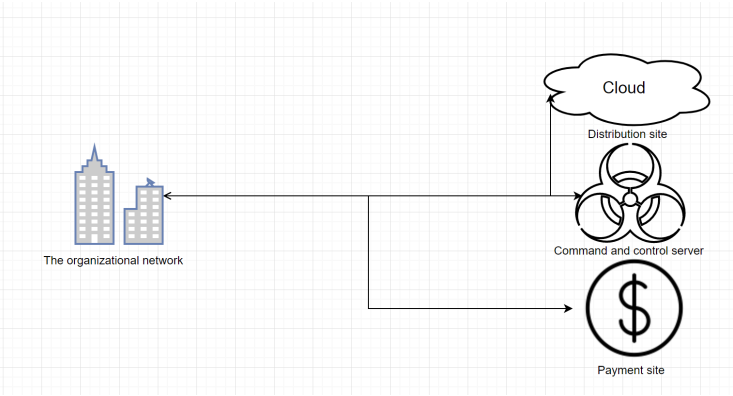


Figure 2.9: Illustration of different types of communication between the target and the attacker according to the threat intelligence feed *ransomware.abuse.ch*

2.2.2.2 Data collection

Data is collected from open source threat intelligence feeds. For this use case, feed of distribution sites, command and control servers and malicious payment site is leveraged. Currently there was only one open-source provider that is keeping track of ransomware indicators and that is:

- ransomwaretracker.abuse.ch

This ransomware tracker is updated every 5 minutes and it provides indicators for following "family" of ransomware: TeslaCrypt, CryptoWall, TorrentLocker, PadCrypt, Locky, CTB-Locker, FAKBEN, and PayCrypt. *Abuse.ch* is in threat intelligence community known as a trustworthy source with a good reputation with minimum false positives.

2.2.2.3 Analysis

Selected threat intelligence feeds were collected using Threat Intelligence Platform, in this case, Collective Intelligence Framework. Indicators that are extracted from these feeds are URLs that are further enriched as described in the analysis chapter in the section 1.6.2. Collective Intelligence Framework allows us to enable custom confidence for each feed as well as custom tags. This feed has a high confidence as it is regularly updated and *abuse.ch* is an open-source provider that has a good reputation in the threat intelligence community. This was confirmed and reviewed by the security researchers from Accenture Security Incident Response team that operates a security operation center for its clients. Therefore, these indicators have confidence nine out of ten. They are sent to Splunk platform where they will be correlated with data from various sources. Proposed time period before updating indicators in the used platforms is 1 hour to keep performance benefit as well as the benefit

of having the most recent data. URLs are enriched to fully qualified domain names with lowered confidence which are further enriched to IPv4 addresses in case of multiple sub-domains hosted on multiple servers.

In the case of ransomware indicators, there is no limited lifespan. Even though the command and control server might be already located somewhere else or being shut down by law enforcement agencies, it would still indicate a behavior that should be properly investigated to find out why that machine is contacting a server that was known for malicious activities. There is an option that the ransomware is just trying to "phone home" from a predefined or generated list of his command and control servers until an active server is found. This way, early detection might help us identify infected machines before the damage is done.

To fulfill use case requirements and correlate data with extracted indicators, data sources monitoring network traffic and web traffic must have been ingested into Splunk. This includes all firewalls monitoring network traffic, Domain Name Server (DNS) data for the domain names indicators and Web Gateway, Proxy or Endpoint data for the URL indicators. In this thesis, Palo Alto firewall [50] has been leveraged as a source of network traffic logs and Squid [52] for proxy logs. In the production environment, it is recommended to ingest also anti-virus and Windows audit logs to provide the full picture for the cyber security teams.

2.2.2.4 Production

In this stage, correlation searches were designed. Natural language is used to describe the logical flow of the use case, whereas in the realization chapter technical details are described.

First use case, *Alert - Detection of traffic from/to the malicious distribution site*, its matching alert rule has two sub-rules and logical condition that needs to be satisfied for use cases to trigger:

- Source IP belongs to the internal network and this includes data from all network devices across various vendors AND Destination IP/domain/URL is on the indicator list as malicious distribution site
- Destination IP belongs to the internal network and this includes data from all network devices across various vendors AND Source IP/domain/URL is on the indicator list as malicious distribution site

The second and third use case have similar logical flow, but focus on command and control server or malicious payment site:

Detection of traffic from/to the malicious command and control server:

- Source IP belongs to the internal network and this includes data from all network devices across various vendors AND Destination IP/domain/URL is on the indicator list as command and control server

- Destination IP belongs to the internal network and this includes data from all network devices across various vendors AND Source IP/domain/URL is on the indicator list as command and control server

Detection of traffic to the malicious payment site:

- Source IP belongs to the internal network and this includes data from all network devices across various vendors AND Destination IP/domain/URL is on the indicator list as malicious payment site

- Destination IP belongs to the internal network and this includes data from all network devices across various vendors AND Source IP/domain/URL is on the indicator list as malicious payment site

This can be, however, matched by one rule and the contextual data about the type of malicious indicator can be provided in the alert. It depends on the organizational needs and their preferences and based on their reaction plan for each scenario.

2.2.2.5 Evaluation

Evaluation of the results serves as feedback for further improvements. Proposed testing includes simulated connections to the known malicious URLs, domains or IPs that has served for malicious activities, including distribution site, command and control server and payment sites.

2.2.3 Phishing

Phishing has been analyzed in the section 1.5.3. In this section, full threat intelligence cycle with specific use cases that leverage threat intelligence are designed. Their goals are to enhance the organization capability to detect users accessing sites known for their malicious content.

2.2.3.1 Requirements

To enhance detection capabilities using publicly available threat intelligence feeds, the use case for accessing phishing sites is implemented:

- Detection of access to sites associated with phishing activities (see Figure 2.10)

To enhance the detection capabilities, network data must be correlated with the threat intelligence feeds. Ideally, the organization will use and ingest data from tools like Proxy or Web Gateway, DNS and Email server.

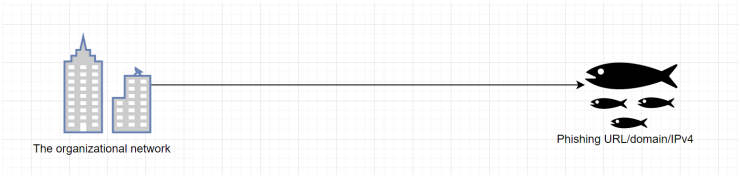


Figure 2.10: User in the organizational network accessing site known for malicious or phishing activities

2.2.3.2 Data Collection

By searching for the keyword "phishing feeds" it is possible to see that many commercial companies are offering phishing threat intelligence feeds that can be purchased.

It was found that communities like *OpenPhish* and *PhishTank* offer phishing feeds that are open for the community usage. OpenPhish launched in 2014 as a result of three years research project on phishing detection. They are using their algorithms to automatically detect phishing sites as well as getting reported submission from their partners. Community, academic and commercial feeds are available. They differ on the amount of enrichment that is shipped with the indicator. PhishTank is operated by OpenDNS which is now a Cisco company. People can submit, verify and share phishing data. Both sources have high reputation in the threat intelligence community.

2.2.3.3 Analysis

The main indicator that is extracted from OpenPhish and PhishTank is URL. This extracted indicator have the highest confidence, nine out of ten as it is primary source of phishing. URLs are enriched to fully qualified domain names with lowered confidence which are further enriched to IPv4 addresses in case of multiple sub-domains hosted on multiple servers. In certain cases, one IP address can host multiple websites where only one is abused for phishing activities. Therefore, the confidence of such indicator is lowered. However, such activity should be detected and reported.

The lifespan of such indicator is not limited as it is important to detect attempts to access sites that have been associated with phishing activities even though the server is already shut down.

Phishing email campaigns can be launched from the same domains that are then used for hosting malicious code or phishing site.

To fulfill use case requirements and correlate data with extracted indicators, data sources monitoring network traffic and web traffic must have been ingested into Splunk. This includes all firewalls monitoring network traffic, Domain Name Server (DNS) data for the domain names indicators and Web Gateway, Proxy or Endpoint data for the URL indicators. In this thesis, Palo

Alto firewall has been leveraged as a source of network traffic logs and Squid for proxy logs.

2.2.3.4 Production

In this stage, correlation searches were designed. Natural language is used to describe logical flow of the use case, whereas in the implementation chapter technical details are described.

First use case, *Detection of access to sites associated with phishing activities*, its matching alert rule has a logical condition that needs to be satisfied for use cases to trigger:

- Alert - Source address belong to the internal organizational network and destination URL is on the indicator list
- Alert - Source address belong to the internal organizational network and destination domain is on the indicator list
- Alert - Source address belong to the internal organizational network and destination IP address is on the indicator list

2.2.3.5 Evaluation

Evaluation of the results serves as feedback for further improvements. Proposed testing includes connections made from the lab environment to the known malicious URLs their domains and IPs that has served for malicious and phishing activities.

Chapter 3

Realization

This chapter describes implementation steps done to fulfill the desired architecture and requirements defined in the threat intelligence cycle. It includes installation and configuration steps that are pre-requirements for the integration.

3.1 Installation

This section describes implementation steps done which are pre-requirements for the integration of Threat Intelligence Platform (Collective Intelligence Platform) with Security information and event management tool (Splunk platform). It includes installation and base configuration.

3.1.1 Splunk

Splunk is a multi-platform tool that can be installed on Windows, Unix or other operating systems. In this thesis Unix machine based on Red Hat Enterprise Linux on Amazon Web Services have been leveraged. Amazon Web Services is an infrastructure as a service (IaaS) provider. In this thesis, it has provided instances sufficient performance that is required by Splunk for a reasonable price.

As one of the requirements for the thesis is to have a solution that is ready for production use, the installation script has been created to make it easier for the reader to install and run Splunk platform. However, this script serves as proof of concept or for small environments and should be adjusted for larger environments that ingest more than 10 GB of data per day according to Splunk Architecture guidance that was presented on their video-conferences.

3.1.2 Collective Intelligence Framework

Collective Intelligence Framework was installed on the Amazon Web Services as well. The newest stable release of Collective Intelligence Framework was downloaded and installed according to the documentation[7]. In the following text this platform will be referred as *CIF Server*

3.1.2.1 Software Development Kit

Collective Intelligence Framework Software Development Kit is shipped and installed together with the CIF platform. However, it is a separate software that serves for the communication with the *CIF Server*. It can be used locally on the same machine or remotely based on the architecture needs. According to the proposed design, in this thesis CIF Software Development Kit resides on the same machine as the *CIF Server* and it is used to locally pull data from the *CIF Server*

3.2 Configuration

This section is describing the configuration that had to be done prior to the integration process. It includes both Collective Intelligence Platform and Splunk platform.

3.2.1 Collective Intelligence Framework

The first step in the threat intelligence cycle is *data collection*. Threat intelligence feeds have been analyzed and designed in the previous chapters. They are implemented in this chapter.*CIF Server* is using a rule-based framework to define parameters to collect selected data feed and to define properties of the feed such as confidence, tag or parsing method. These rules are written in YAML Language[8].

A couple of default rules are shipped with the CIF and can be used as a template for building new rules. Important fields that need to be noticed are *parser* which describes format of the data feed, *tags* that describe context of the data, *confidence* which informs about trustworthiness of the data and *pattern* to define regex to parse data in case of usage of the format that is structured but not in the known formats as CSV, XML or JSON.

These configuration files have been reviewed and written to collect selected data feeds. In this thesis, only selected fields have been listed and full configuration can be found on the attached CD.

Dan.me.uk:

[7] https://github.com/csirtgadgets/bearded-avenger-deploymentkit/wiki
[8] YAML stands for *YAML Ain't Markup Language*

```
   provider: dan.me.uk
   tags:
   - tor
   parser: pattern
   remote: https://www.dan.me.uk/torlist/
   confidence: 9
   pattern: ^(\S+)$
   values:
      - indicator
```

Torproject.org:

```
   provider: torproject.org
   confidence: 9
   tags:
      - tor
   parser: pattern
   values:
      - indicator
      - reporttime

   remote: https://check.torproject.org/exit-addresses
   pattern: ExitAddress (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})
       ([^\r\n]*)
```

Ransomware.abuse.ch:

```
parser: csv
   provider: ransomware.abuse.ch
   confidence: 9
   tags: ransomware
   values:
   - reporttime
   - threat
   - description
   - null
   - indicator
   remote: http://ransomwaretracker.abuse.ch/feeds/csv
```

Openphish.com:

```
   tags: phish
   provider: openphish.com
   confidence: 9
   values: indicator
   itype: url
   pattern: ^(.+)$
   remote: https://openphish.com/feed.txt
```

Phishtank.com:

```
parser: json
remote: http://data.phishtank.com/data/online-valid.json
defaults:
provider: phishtank.com
application:
- http
- https
confidence: 9
tags: phishing
feeds:
urls:
itype: url
map:
- submission_time
- url
- target
- phish_detail_url
- details
values:
- lasttime
- indicator
- description
- altid
- additional_data
```

3.2.2 Splunk

Splunk environment needs to be configured in order to receive, index and process incoming data correctly. This configuration includes index creation and inputs definition. Splunk is using a modular structure to define configurations on different layers. Splunk App is a hierarchical structure of configuration files that allow developers to create a re-usable app with all it is necessary configurations.

In this thesis, such app will be created to utilize threat intelligence. Splunk defines an app directory structure that needs to be followed (see Figure 3.1). The reader can find more information about Splunk app architecture in the developer portal[9]. Default folder serves for the configurations that are app specific and will be overwritten in the next upgrade. Local folder serves for local changes that have higher priority in case of conflict with the default folder but configuration changes will not get overwritten if the app is upgraded. The reader can find more information about configuration precedence in Splunk on the following documentation page [53].

[9] http://dev.splunk.com/view/dev-guide/SP-CAAAE29

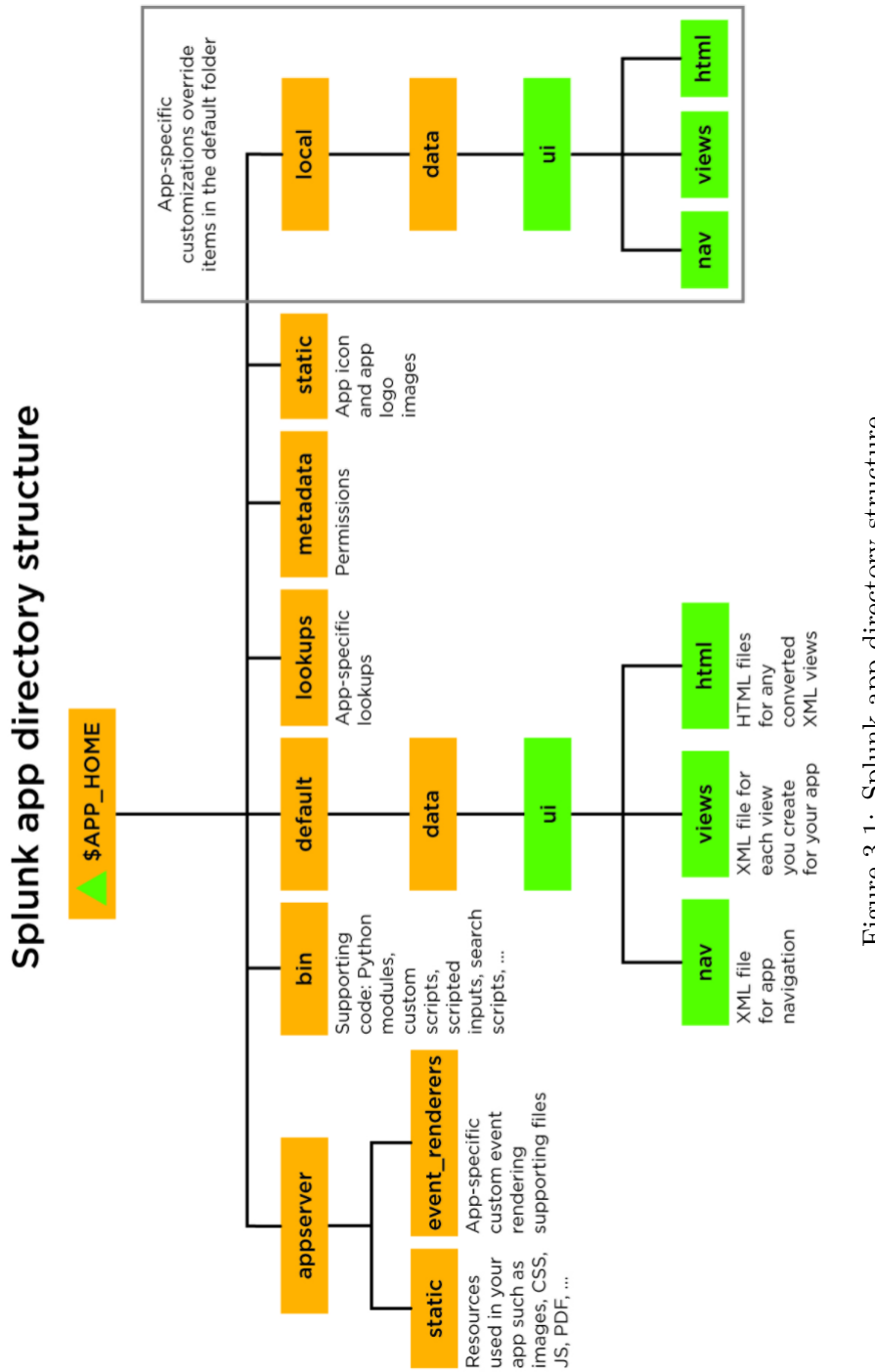


Figure 3.1: Splunk app directory structure

Two configuration files have been created, *inputs.conf* and *indexes.conf* to setup input and index configuration needed for proper data ingestion.

Splunk index is like a jar where data with similar certain characteristics are put. Advanced settings can be then applied to the index like data retention, data rotation or specific searches limited to certain technology, i.e Windows logs. Index can be defined in the following way in the *indexes.conf* configuration file:

```
[threatintel]
coldPath = $SPLUNK_DB/threatintel/colddb
enableDataIntegrityControl = 0
enableTsidxReduction = 0
homePath = $SPLUNK_DB/threatintel/db
maxTotalDataSizeMB = 512000
thawedPath = $SPLUNK_DB/threatintel/thaweddb
```

Inputs are defining a way how Splunk expect to ingest the data. It involves listening on the certain port, scripted inputs or reading a certain directory. According to the proposed design, input for receiving Splunk communication on port 9999 have been created:

```
[splunktcp://9999]
connection_host = ip
disabled = 1
```

More information about configuration options are available for the reader in the Splunk .spec files [10] [11]

It is important to note that this is the result of attending more than 60 hours of video conferences led by Splunk instructor to fully understand concepts and configurations that are necessary to administer, architect the environment and develop new apps.

3.3 Integration

Installation and Configuration section were pre-requirements for the integration of those two platforms. At this point, data are being collected from the selected threat intelligence feeds, normalized to the common format and enriched. Splunk platform is configured and ready to receive this data. The goal of the integration is to implement the proposed design (see Figure 2.1).

Implementation steps for the integration are:

1. Setting up a CIF server and CIF router as a daemon service

[10] indexes.conf - http://docs.splunk.com/Documentation/Splunk/6.5.3/Admin/Indexesconf
[11] inputs.conf - http://docs.splunk.com/Documentation/Splunk/6.5.3/Admin/Inputsconf

2. Install and configure Splunk Universal Forwarder on the *CIF Client* machine
3. Setting up a CIF Client script that pulls data from the *CIF Server*
4. Validate data is being ingested in Splunk

3.3.1 Setting up a CIF server and CIF router as a daemon service

Even thought CIF Server is shipped with the Ansible playbook that can setup a service on the Ubuntu 16 and higher, it was necessary to understand how this service works and adjust it to the needs of this thesis. To run the service in the debug mode, *-d* parameter must be added to the service definition that is in Ubuntu 16 by default in the */etc/systemd/system/csirtg-smrt.service*

For CIF router, in */etc/cif.env* a number of threads for the *hunter* is selected. Hunter is the module that is responsible for the enrichment of the threat intelligence feeds. In this thesis, based on the discussion with the owner and main developer of the *Collective Intelligence Platform*, 6 threads for the hunters have been selected. Hardware that has been used in this thesis is AWS m4.xlarge instance [12]

3.3.2 Install and configure Splunk Universal Forwarder

Splunk Universal Forwarder is a light-weight version of the Splunk Enterprise that has the capability to collect data from files and directories and reliably forward them to the Splunk platform. It handles network failures with queuing and for the large-scale environment can load-balance forwarded data across multiple indexers. This is crucial for the performance in the large-scale production environment as if data are distributed across multiple indexers, Splunk map and reduce search model can run its search and correlation searches in parallel [54]. To install Splunk Universal Forwarder, an installation script has been created and it is included on the attached CD.

To monitor folders where the threat intelligence data are extracted according to the architecture diagram, an app for *inputs* and *outputs* is created. This could be done in the single app, however, it is a best practice and recommendation from the Splunk engineers to create small reusable modules (apps) where each module has it is specific purpose and goal.

Inputs.conf have been configured following this template:

```
[monitor:///<<directory>>]
sourcetype = _json
index = threatintel
disabled = false
```

[12] https://aws.amazon.com/ec2/instance-types/

Monitor stanza defines a folder or file that is monitored. For every folder or file, a separate stanza is created. This is not always true, as wildcards can be used but for simplicity, they haven't been used in this thesis. Index defines to which "jar" is data sent. For this thesis, an index *threatintel* have been created and is used to store threat intelligence data.*_json* is one of the pre-trained sourcetype that Splunk can parse and it expresses the type of data for that specific stanza.

Outputs.conf have been configured as following:

```
[tcpout]
defaultGroup = primary_indexers


[tcpout:primary_indexers]
server = 52.37.0.230:9999

[tcpout-server://52.37.0.230:9999]
```

In outputs.conf a destination for the Universal Forwarder is configured. In this thesis, only one indexer is used, however in larger environments this consist of several indexers where the data are load-balanced across them to increase search performance.

3.3.3 CIF Client script

In order to achieve this design step (see Figure 2.4), a simple script automatically and regularly pulls data from the CIF Server and saves them into the file. This script is leveraging the *CIF Software Development Kit* and extract data from the internal *CIF Server* storage. This script is put into the */etc/cron.hourly/* directory that launches the script every hour.

Script named *extract* have been created and here partially illustrate the logic of the script, with «provider» as different provider, «tags» as various tags and «output_file» as output file:

```
#!/bin/bash
cif --config /home/cif/.cif.yml --provider <<provider>> --tags
    <<tag>> --last-hour --format json >> <<output_file>>
```

This script used *CIF Software Development Kit* to access and extract data from the last hour, from selected providers in the JSON format and append them to the selected files. Splunk is already monitoring these directories and in a case of new data is appended, it will be recognized and automatically send them to Splunk platform.

3.3.4 Validate data is being ingested in Splunk

To validate that data are being received in the Splunk platform, a command that search data across the selected index is issued[13]. Data are being ingested as the number of received events in this index is over 30 000.

3.4 Usecase development

This section describes methods, techniques and steps that have been leveraged in order to implement the use cases that have been previously designed.

3.4.1 Search Processing Language

Search Processing Language or it is abbreviated form, *SPL* is a language that is used in the Splunk platform to search, filter and transform data across the indexes filled with various data sources. In this thesis, this language is also used to define correlation searches and all its sub searches that are required. Splunk defines SPL as:

> *SPL encompasses all the search commands and their functions, arguments, and clauses. Its syntax was originally based on the Unix pipeline and SQL. The scope of SPL includes data searching, filtering, modification, manipulation, insertion, and deletion.*

Search processing language (SPL) is powerful and allows to write efficient correlation searches. In order to meet requirements for the scalability, advanced search techniques in the SPL language needs to be used to perform efficiently in large-scale environments. Even though such environment is not leveraged in this thesis, these techniques have been learned from Splunk documentation and Splunk video-led conferences. More than 50 hours of video conferences have been spent to deeply understand the language and how to use and leverage these advanced techniques. These searches are in this thesis compared with the more basic techniques to prove the concept and usefulness of using such advanced searches.

As this processing language is very complex, the reader can find more information and documentation with examples on the Splunk docs [14].

3.4.2 Datamodels

Splunk concept of data models is very important in this thesis. It enables efficient correlation among various data sources. Let's first note how Splunk defines datamodels [15]:

[13] index=threatintel is typed into the search bar and searched all-time
[14] https://docs.splunk.com/Documentation/Splunk/6.5.3/Search/Aboutthesearchlanguage
[15] http://docs.splunk.com/Documentation/Splunk/6.5.3/Knowledge/Aboutdatamodels

> *A hierarchically structured, search-time mapping of semantic knowledge about one or more datasets that encode the domain knowledge necessary to generate specialized searches of those datasets.*

As data models are used in Splunk for more reasons, this definition might be confusing for the reader to understand it in the context of this thesis.

Let's use an example to explain the role of the data models in this thesis. In the case of network firewalls, multiple vendors exist. As there is no standardized model for the log fields, each vendor uses its own vocabulary for logging. For example, one vendor use *dst_port* to inform about the destination port used in the network communication whereas some other vendor calls it *dport*. It is obvious that the information value is the same even if the field names are different. Data models are trying to bring a standardized abstract model that if being used, normalize these fields and map them to one common field name.

This allows to search for the common field name, for example, dest_port across all data sources that are logging such information. Correlation searches using *SPL* that leverage data models are then more efficient, perform better and are syntactically shorter as only one search is needed to search across all data sources. Correlation searches that leverage this feature are then vendor independent and reusable in multiple environments.

3.4.2.1 Accelerated Datamodel

Data model acceleration is a tool that can significantly speed up and increase performance when searching through large-datasets [55]. This is achieved using high-performance analytics store [55]. Because requirements for this thesis requires scalability and performance even for large environments, data models are accelerated. This is done via Graphical User Interface. It is, however, optional step and because it requires additional storage for the *.tsidx* files, it can be skipped. In this thesis accelerated data models have been used and they have sped up the search performance more than 10 times as detailed later.

3.4.2.2 Technical Addon

However, to use and map logs to the Splunk common information model, configurations – in Splunk terminology called *Technical Addons* needs to be written for every data source to map vendor custom fields to the *Splunk Common Information Model*. One of the great advantages of Splunk platform is the ability to download these technical addons from their community portal where vendors and developers are motivated to share technical add-ons for their device so it is not necessary to "reinvent the wheel" [16]. In this thesis,

[16] https://splunkbase.splunk.com/

technical addons that parse, extract and map logs from Palo Alto firewall and Squid proxy have been leveraged. Documentation of how to install technical add-on is available for the reader on *Splunk docs* for every technical addon through the Splunkbase.

3.4.3 Data correlation

Data correlation is essential for the selected use cases. In this section is described how the correlation between threat intelligence indicators and all the other data sources in the environment is implemented.

In the dictionary, the definition of correlation is:

> *mutual relation of two or more things, parts, etc.*

In our case, it is a relation between data present in the environment and the indicators gained from the threat intelligence feeds. Correlation is defined in the use cases which describes where the mutual relation between these elements (data in the environment and the indicators gained from the threat intelligence feeds) happen. These correlations are then transformed into the Search Processing Language to leverage the platform to quickly perform and correlate data.

Search Processing Language offers many tools that allow to perform and transform these use cases into the Splunk platform. However, as in every language, some methods are more or less effective and therefore affect the resulting performance. It is very important that the implementation steps were optimized to maximize its performance not only in small but also large-scale production environments. To efficiently utilize threat intelligence, all data must be correlated near real-time.

3.4.3.1 Subsearch

One possible implementation was by leveraging Splunk sub-search capability. Sub-search takes the result from one search and uses the results in another search. This can be used to correlate events across different data sources that have at least one common field which in this case would be *indicator of compromise*.

Even though this solution would work perfectly in small environments, there is a significant drawback. Splunk by default limits sub searches to maximum 10 thousand events and the maximum runtime of 60 seconds [56]. In large environments, the results might be incomplete which is not acceptable from the security point of view. It is possible to increase these default values, however, that is not recommended approach because sub searches can't run in parallel in the case of distributed environment which negatively affect the overall performance.

3.4.3.2 Lookup tables

The approach that has been selected in this thesis is by using scheduled searches to generate lookup tables that are further used to correlate against the large set of traffic data. Lookup table is a regular CSV file containing exported fields. This approach would bring significantly better performance in the large distributed environments with a multiple number of indexers as it allows to run in parallel.

This has been confirmed by Splunk Professional Services during their video conferences.

For every combination of use case and type of indicator, lookup generating search will be used. That means if 3 use cases have been selected – *TOR, Ransomware, Phishing* and 3 types of indicators – *IPv4, FQDN, URL*, 9 different lookup generation searches are done for every combination. However, threat intelligence data for the TOR use case only cover IPv4 addresses and can't be further enriched therefore two combinations are subtracted. All seven lookup generating scheduled searches are defined in the *savedsearches.conf* but for simplicity, only the search part of the stanza is shown here.

Below is a simplified example of the lookup generating searches for the *Phishing* use cases. It doesn't contain other parameters like *cron_schedule, schedule_window or search_mode*. Similar searches are generated for *TOR and Ransomware* use case.

```
[URL - Phishing - Lookup generating command enriched with IP]
search = index=threatintel tags=phish* itype=url\
| table indicator,confidence,rdata,provider,tags, description, cc,
    asn_desc | outputlookup createinapp=true malicious_url_phish.csv
[FQDN - Phishing - Lookup generating command with enriched FQDN]
search = index=threatintel tags=phish* itype=fqdn\
| table indicator,confidence,rdata,provider,tags, cc, asn_desc,
    description | outputlookup createinapp=true
    malicious_fqdn_phish.csv
[IPv4 - Phishing - Lookup generating command for IPv4]
search = index=threatintel tags=phish* itype=url rdata!="0.0.0.0" |
    rename rdata as "ip_indicator", indicator as "url_rdata" | table
    ip_indicator, confidence, url_rdata, provider,tags, description |
    rename ip_indicator as "indicator", url_rdata as "rdata" | eval
    confidence=(confidence-3) | outputlookup createinapp=true
    malicious_ipv4_phish.csv
```

This piece of configuration stanza is located in the *savedsearches.conf* and it generates every hour an updated lookup table where it filters threat intelligence data with tags equal phish*, where * serves as wildcard, filters enriched data with invalid IP address, filter type of indicator and in the case of enriched IP address lowers the confidence as the IP address that host site where phishing activities have been reported can be shared and used by sites that

are legitimate.

With the same logic and adjusted *search*, four more lookup tables are created every hour. Their configuration files are available in the attached CD.

- URL - Ransomware - Lookup generating command
- FQDN - Ransomware - Lookup generating command with enriched FQDN
- IPv4 - Ransomware - Lookup generating command for IPv4
- IPv4 - TOR - Lookup generating command

3.4.3.3 Correlated datamodels

At this point of the thesis, platforms are integrated, data are properly ingested from the *Collective Intelligence Platform* to *Splunk platform*, lookup tables are being generated and use cases are defined.

The last step before implementing the defined use case is to select Splunk data models that are going to be used in the use cases. Data sources that are required have been already defined and designed in the Analysis and Design chapter. This is important in order to create correlation searches that are re-usable as they are implemented according to the Splunk Common Information Model. Therefore, data model *Network Traffic* will be used for use cases working with IPv4 addresses as indicators – TOR and enriched IPv4 indicators for Ransomware and Phishing use case. Logs containing communication in the network are being put to this data model. For use cases that use FQDN and URL as indicators, *Web* data model will be leveraged as it logs this information that can be correlated with the FQDN or URL indicators. Fully qualified domain names and URLs are used in the *Ransomware* and *Phishing* use case.

Furthermore, because these data models can contain hundred of thousands of events per certain period of time from various logging devices, suitably advanced search technique must be used to increase the performance to the level that enables almost real-time correlation with the threat intelligence data on a reasonable hardware. To illustrate this performance difference, three different scenarios have been tested and measured. This is an important step as it takes most of the search time in the correlation searches that are implemented later on. Without a proper performance, utilization of the threat intelligence data would not be possible as it would not be possible to correlate this data in reasonable time.

To measure the performance of the search, a unit *Events per second* is defined as the number of scanned events divided by the time of the search. Measurement is done using *Search job inspector*, which is a feature that displays detailed information about the processed search job.

It is important to note that the following three scenarios have the same output (see Figure 3.2). The output is defined as:

- List and summarize all communication (src_ip <-> dst_ip) in the network

Three types of searches that have been tested and measured are:

- Data model search and statistics
- Advanced data model search and statistics
- Advanced accelerated data model search and statistics

For the *Datamodel search and statistics* following search have been used:

```
| datamodel Network_Traffic All_Traffic search | stats
    sum(All_Traffic.bytes_in) as "Bytes in",
    sum(All_Traffic.bytes_out) as "Bytes
    out",values(All_Traffic.action) as "Action taken", count as
    "Number of connections" ,values(sourcetype) as "Datasource type"
    by All_Traffic.src_ip,All_Traffic.dest_ip | sort - "Bytes out"
```

Results of the search are the following (see Figure 3.3).

The search has returned 3,276 results, scanned 74.951 events in 5.017 seconds. Using defined event per second unit, the basic data model search has performed approximately *14939 events per second*

For the *Advanced datamodel search and statistics*, different search technique have been used. This technique is leveraging *tstats* command, which leverage deep knowledge how Splunk platform works internally. Following search have been used:

```
| tstats sum(All_Traffic.bytes_in) as "Bytes in",
    sum(All_Traffic.bytes_out) as "Bytes
    out",values(All_Traffic.action) as "Action taken", count as
    "Number of connections" ,values(sourcetype) as "Datasource type"
    from datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip | sort -"Bytes out"
```

Results of the search are the following (see Figure 3.4).

The search has returned 3,277 results, scanned 74.967 events in 2.727 seconds. Using defined event per second unit, the basic data model search has performed approximately *27490 events per second*. This is a significant increase in the performance.

The last type of search is using the same search technique, however, accelerated data models have been created. It takes a couple of hours to create these accelerated data models but it is definitely worth it as the results have shown. Accelerated data models are only and only possible to leverage using the advanced search technique like in the *Advanced data model search and statistics* case.

Results of the search are the following (see Figure 3.5).

Events | Patterns | Statistics (3,272) | Visualization

20 Per Page | Format | Preview

< Prev 1 2 3 4 5 6 7 8 9 ... Next >

All_Traffic.src_ip	All_Traffic.dest_ip	Bytes in	Bytes out	Action taken	Number of connections	Datasource type
10.99.19.98	172.29.24.239	14646778	264284790	allowed	3516	pan:traffic
10.99.19.98	172.29.24.31	12484162	232993778	allowed	3496	pan:traffic
10.99.20.61	10.99.20.72	24876146	199343351	allowed	1	pan:traffic
10.99.20.72	172.29.24.31	3629842	51693769	allowed	3459	pan:traffic
10.99.20.72	172.29.24.239	3687818	51351475	allowed	3496	pan:traffic

Figure 3.2: Part of the listed and summarized all communication in the network according to the Network_Traffic datamodel

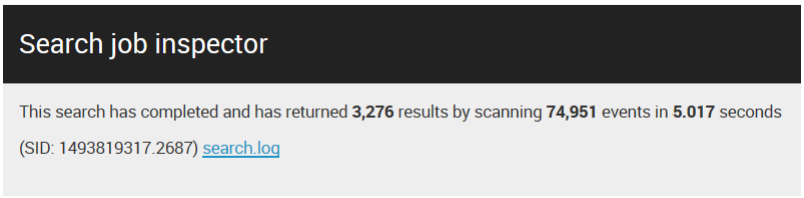


Figure 3.3: Performance of the basic datamodel search and statistics

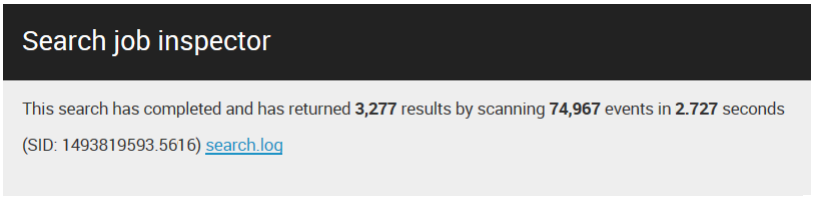


Figure 3.4: Performance of the Advanced datamodel search and statistics

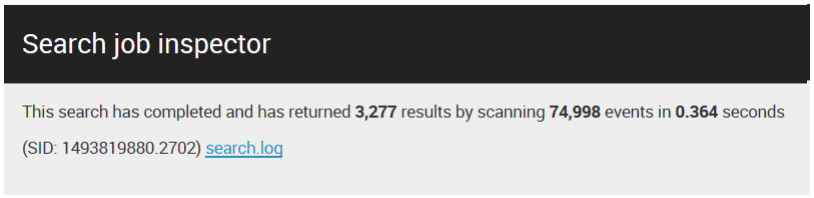


Figure 3.5: Performance of the Advanced accelerated datamodel search and statistics

The search has returned 3,277 results, scanned 74.998 events in 0.364 seconds. Using defined event per second unit, the basic data model search has performed approximately *206038 events per second*.

To sum up, all three types of searches have resulted almost the same number of results (the first one has returned one less because in the meantime one new connection has happened in the network). They have searched a different number of events as event number is increasing with the time. However, results have been normalized to one common unit, processed events per second. The result is summarized as:

- Data model search and statistics - 14939 events per second
- Advanced data model search and statistics - 27490 events per second
- Advanced accelerated data model search and statistics - 206038 events per second

The difference between the best and the worst type and technique is more than 13 times difference in processing power. This has significant consequences

on the cost and performance. By using the best method, hardware performance can be reduced to get the same performance and therefore reduce the cost of the infrastructure or to simply have much better performance for the same cost if the hardware infrastructure is static. Therefore, all use cases will use the best type of search illustrated in this subsubsection.

3.4.4 Correlation search

In this subsection, implementation of the use cases that have been defined in the Analysis and Design chapters is done in the Splunk platform. At this point, a *production* stage of the threat intelligence cycle (see Figure 1.3) [1] is done. To achieve this point, Collective Intelligence Platform and Splunk platform have been selected, installed and configured. These two platforms have been then integrated. The most difficult part of achieving that was to have sufficient knowledge of Splunk platform to be able to properly analyze, design and realize these stages. As it was shown in the previous subsubsection, two searches that produce same result are far from being equally good. Correlation algorithms, also known as correlation searches have been implemented in a way that they are as efficient as they can be, scalable, perform as well as they can and in their simplest form. This allows not only to easier understand them but also to modify them or create a new correlation searches according to the organizational needs and use them as a template. This is a result of extensive research during video conferences and video training led by Splunk engineers and documentation reading.

3.4.4.1 TOR

In the Design chapter, two sub-use cases have been defined and designed:

- Alert - Detection of TOR traffic originated in the organization
- Alert - Detection of TOR traffic with destination in the organization

Goal of this correlation search is to alert if any inbound or outbound connection to the TOR network is detected (see Figure 2.5, 2.6).

Final correlation algorithms (searches) have the following form:

```
[TOR - Traffic coming from the organization]
| tstats sum(All_Traffic.bytes) as "Bytes",values(All_Traffic.action)
    as "Action taken", count,values(sourcetype) as "Datasource type"
    from datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip,\
| lookup malicious_ipv4_tor.csv indicator as All_Traffic.dest_ip
    OUTPUTNEW indicator,confidence,tags,provider,Country, cc,
    asn_desc \
```

```
| where confidence>0 | eval Bytes=round(Bytes/1024/1024,2) | rename
    Bytes as "MegaBytes Transferred", All_Traffic.dest_ip as
    "Destination IP", All_Traffic.src_ip as "Source IP", count as
    "Number of connections", asn_desc as "ASN Description", cc as
    "Country Code", confidence as "Confidence"
```

```
[TOR - Traffic coming to the organization]
| tstats sum(All_Traffic.bytes) as "Bytes",values(All_Traffic.action)
    as "Action taken", count,values(sourcetype) as "Datasource type"
    from datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip,\
| lookup malicious_ipv4_tor.csv indicator as All_Traffic.src_ip
    OUTPUTNEW indicator,confidence,tags,provider,Country, cc,
    asn_desc \
| where confidence>0 | eval Bytes=round(Bytes/1024/1024,2) | rename
    Bytes as "MegaBytes Transferred", All_Traffic.dest_ip as
    "Destination IP", All_Traffic.src_ip as "Source IP", count as
    "Number of connections", asn_desc as "ASN Description", cc as
    "Country Code", confidence as "Confidence"
```

These correlation searches return assets that have been communicating with the TOR network (inbound and outbound separately) and includes metadata as action taken by the device that has logged such communication, bytes transferred, confidence, Country of the indicator, a description based on the autonomous system number. This allows the person that investigate the incident to better understand the context of the alert to conduct further investigation.

To put this correlation searches into a detection mechanism, an *alert search* is created in the *savedsearches.conf*:

```
[TOR - Traffic coming from the organization]
action.email.useNSSubject = 1
alert.digest_mode = False
alert.suppress = 1
alert.suppress.fields = All_Traffic.src_ip
alert.suppress.period = 10m
alert.track = 1
auto_summarize.dispatch.earliest_time = -1d@h
counttype = number of events
cron_schedule = */10 * * * *
dispatch.earliest_time = -10m@m
display.general.type = statistics
display.page.search.mode = verbose
display.page.search.tab = statistics
display.visualizations.show = 0
display.visualizations.type = mapping
enableSched = 1
```

```
quantity = 0
relation = greater than
request.ui_dispatch_app = search
request.ui_dispatch_view = search
search = | tstats sum(All_Traffic.bytes) as
    "Bytes",values(All_Traffic.action) as "Action taken",
    count,values(sourcetype) as "Datasource type" from
    datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip,\
| lookup malicious_ipv4_tor.csv indicator as All_Traffic.dest_ip
    OUTPUTNEW indicator,confidence,tags,provider,Country, cc,
    asn_desc \
| where confidence>0 | eval Bytes=round(Bytes/1024/1024,2) | rename
    Bytes as "MegaBytes Transferred", All_Traffic.dest_ip as
    "Destination IP", All_Traffic.src_ip as "Source IP", count as
    "Number of connections", asn_desc as "ASN Description", cc as
    "Country Code", confidence as "Confidence"
```

```
[TOR - Traffic coming to the organization]
action.email.useNSSubject = 1
alert.digest_mode = False
alert.suppress = 0
alert.track = 1
auto_summarize.dispatch.earliest_time = -1d@h
counttype = number of events
cron_schedule = */10 * * * *
dispatch.earliest_time = -15m@m
display.general.type = statistics
display.page.search.mode = verbose
display.page.search.tab = statistics
enableSched = 1
quantity = 0
relation = greater than
request.ui_dispatch_app = ThreatIntel
request.ui_dispatch_view = search
search = | tstats sum(All_Traffic.bytes) as
    "Bytes",values(All_Traffic.action) as "Action taken",
    count,values(sourcetype) as "Datasource type" from
    datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip,\
| lookup malicious_ipv4_tor.csv indicator as All_Traffic.src_ip
    OUTPUTNEW indicator,confidence,tags,provider,Country, cc,
    asn_desc \
| where confidence>0 | eval Bytes=round(Bytes/1024/1024,2) | rename
    Bytes as "MegaBytes Transferred", All_Traffic.dest_ip as
    "Destination IP", All_Traffic.src_ip as "Source IP", count as
    "Number of connections", asn_desc as "ASN Description", cc as
    "Country Code", confidence as "Confidence"
```

These alert searches include cron schedule to run every 10 minutes in a last 15-minute time window. This is ensured with *dispatch.earliest_time* set to -15m@m. This is due to the fact that Splunk platform internally might delay the search job in case resources are needed for another job. This may cause running the search later and in order to not lose visibility, broader time is used. To avoid overwhelming number of alerts, alerts throttling is set up. It is defined in the TOR traffic from the organization as:

```
alert.suppress.fields = All_Traffic.src_ip
alert.suppress.period = 10m
alert.track = 1
```

This ensures that if certain computer connects to the TOR network, alert with the same source IP will be raised only once. This setting was not applied for the inbound TOR traffic as if the potential attacker is trying to scan our network via TOR, it might use the same source IP and it is important to know it. On the other hand, if the destination IP is the same and the sources are different, it might indicate a coordinated botnet attack coming via TOR. Therefore no throttling was applied on the inbound TOR traffic and it was applied only on outbound traffic. Results of this correlation search are described in the Testing chapter.

3.4.4.2 Ransomware

In the Design chapter, three use cases have been defined and designed:

- Detection of traffic from/to the malicious distribution site
- Detection of traffic from/to the malicious command and control server
- Detection of traffic to the malicious payment site

Goal of this correlation search is to alert if any connection to the site associated with ransomware activities is detected (see Figure 2.9). In the configuration file, *«type»* is either distribution site, payment site or command and control server.

Final correlation algorithms (searches) have the following form:

```
[Ransomware - <<type>> - communication detected to malicious site]
| tstats count, sum(Web.bytes), values(Web.http_content_type),
    values(Web.http_method) from datamodel=Web.Web where
    (Web.http_method=GET OR Web.http_method=POST) by Web.dest, Web.src
| lookup malicious_url_ransom.csv indicator as Web.dest OUTPUTNEW
    indicator,confidence,tags,provider, cc, asn_desc, description,
    rdata,
| search confidence=* AND description="<<type>>"
```

```
[Ransomware - <<type>> - communication detected to enriched IPv4
    address ]
| tstats count,values(All_Traffic.dest_port) as "Destination Ports" ,
    sum(All_Traffic.bytes) as "Transferred Bytes" from
    datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip
| lookup malicious_ipv4_ransom.csv indicator as All_Traffic.dest_ip
    OUTPUTNEW indicator,confidence,tags,provider,rdata, description
| search confidence=* AND description="<<type>>"
```

First correlation searches return assets that have accessed URL that is on the malicious list and includes metadata as methods, content type and the amount of data transferred. This allows the person that investigate the incident to better understand the context of the alert to conduct further investigation.

Second correlation search returns assets that have been communicating with the IP address that was enriched from the URL indicators. This search has lowered confidence as the IP address can be legitimate and it is abused for redirections to phishing sites.

To put this correlation searches into a detection mechanism, an *alert search* is created in the *savedsearches.conf*:

```
[Ransomware - <<type>> - communication detected to malicious site]
action.email.useNSSubject = 1
alert.digest_mode = 0
alert.severity = 5
alert.suppress = 0
alert.track = 1
auto_summarize.dispatch.earliest_time = -1d@h
counttype = number of events
cron_schedule = */10 * * * *
dispatch.earliest_time = -15m@m
enableSched = 1
quantity = 0
relation = greater than
request.ui_dispatch_app = ThreatIntel
request.ui_dispatch_view = search
search = | tstats count, sum(Web.bytes),
    values(Web.http_content_type), values(Web.http_method) from
    datamodel=Web.Web where (Web.http_method=GET OR
    Web.http_method=POST) by Web.dest, Web.src\
| lookup malicious_url_ransom.csv indicator as Web.dest OUTPUTNEW
    indicator,confidence,tags,provider, cc, asn_desc, description,
    rdata, \
| search confidence=* AND description="<<type>>"
```

```
[Ransomware - <<type>> - communication detected to enriched IPv4
    address]
action.email.useNSSubject = 1
alert.digest_mode = 0
alert.suppress = 0
alert.track = 1
auto_summarize.dispatch.earliest_time = -1d@h
counttype = number of events
cron_schedule = */10 * * * *
dispatch.earliest_time = -15m@m
enableSched = 1
quantity = 0
relation = greater than
request.ui_dispatch_app = ThreatIntel
request.ui_dispatch_view = search
search = | tstats count,values(All_Traffic.dest_port) as "Destination
    Ports" , sum(All_Traffic.bytes) as "Transferred Bytes" from
    datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip\
| lookup malicious_ipv4_ransom.csv indicator as All_Traffic.dest_ip
    OUTPUTNEW indicator,confidence,tags,provider,rdata, description \
| search confidence=* AND description="<<type>>"
```

These alert searches include cron schedule to run every 10 minutes in a last 15-minute time window. This is ensured with *dispatch.earliest_time* set to -15m@m. This is due to the fact that Splunk platform internally might delay the search job in case resources are needed for another job. This may cause running the search later and in order to not lose visibility, broader time is used.

Alert throttling was not set as it is important to alert about every distinct connection to the site with phishing or malicious content. Results of this correlation search are described in the Testing chapter.

3.4.4.3 Phishing

In the Design chapter, one use case has been defined and designed:

- Detection of access to sites associated with phishing activities (see Figure 2.10)

The goal of this correlation search is to alert if any connection to the site associated with phishing or malicious activities is detected (see Figure 2.10).

Final correlation algorithms (searches) have the following form:

```
[Phishing - User has accessed phishing site]
| tstats count, sum(Web.bytes), values(Web.http_content_type),
    values(Web.http_method) from datamodel=Web.Web where
    (Web.http_method=GET OR Web.http_method=POST) by Web.dest, Web.src
```

```
| lookup malicious_url_phish.csv indicator as Web.dest OUTPUTNEW
    indicator,confidence,tags,provider, cc, asn_desc, description,
    rdata,
| where confidence>0
```

```
[Phishing - User has accessed IP address that is hosting phishing URL
    ]
| tstats count, values(All_Traffic.dest_port) as "Destination Ports"
    from datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip,
| lookup malicious_ipv4_phish.csv indicator as All_Traffic.dest_ip
    OUTPUTNEW indicator,confidence,tags,provider rdata
| where confidence>0 | rename rdata as "Malicious URL associated with
    this IP"
```

Correlation searches return assets that have accessed URL that is on the malicious list and includes metadata as methods, content type and the amount of data transferred. This allows the person that investigate the incident to better understand the context of the alert to conduct further investigation. The second type of correlation searches return assets that have been communicating with the IP address that was enriched from the URL indicators. This search has lowered confidence as the IP address can be legitimate and it is abused for redirections to phishing sites.

To put this correlation searches into a detection mechanism, an *alert search* is created in the *savedsearches.conf*:

```
[Phishing - User has accessed phishing site]
action.email.useNSSubject = 1
alert.digest_mode = False
alert.suppress = 0
alert.track = 1
auto_summarize.dispatch.earliest_time = -1d@h
counttype = number of events
cron_schedule = */10 * * * *
dispatch.earliest_time = -15m@m
display.general.type = statistics
display.page.search.mode = verbose
display.page.search.tab = statistics
enableSched = 1
quantity = 0
relation = greater than
request.ui_dispatch_app = ThreatIntel
request.ui_dispatch_view = search
search = | tstats count, sum(Web.bytes),
    values(Web.http_content_type), values(Web.http_method) from
    datamodel=Web.Web where (Web.http_method=GET OR
    Web.http_method=POST) by Web.dest, Web.src\
```

```
| lookup malicious_url_phish.csv indicator as Web.dest OUTPUTNEW
    indicator,confidence,tags,provider, cc, asn_desc, description,
    rdata, \
| where confidence>0
```

```
[Phishing - User has accessed IP address that is hosting phishing URL]
action.email.useNSSubject = 1
alert.digest_mode = False
alert.severity = 2
alert.suppress = 0
alert.track = 1
auto_summarize.dispatch.earliest_time = -1d@h
counttype = number of events
cron_schedule = */10 * * * *
dispatch.earliest_time = -15m@m
display.general.type = statistics
display.page.search.mode = verbose
display.page.search.tab = statistics
enableSched = 1
quantity = 0
relation = greater than
request.ui_dispatch_app = search
request.ui_dispatch_view = search
search = | tstats count, values(All_Traffic.dest_port) as
    "Destination Ports" from datamodel=Network_Traffic.All_Traffic by
    All_Traffic.src_ip,All_Traffic.dest_ip,\
| lookup malicious_ipv4_phish.csv indicator as All_Traffic.dest_ip
    OUTPUTNEW indicator,confidence,tags,provider rdata\
| where confidence>0 | rename rdata as "Malicious URL associated with
    this IP"
```

These alert searches include cron schedule to run every 10 minutes in a last 15-minute time window. This is ensured with *dispatch.earliest_time* set to -15m@m. This is due to the fact that Splunk platform internally might delay the search job in case resources are needed for another job. This may cause running the search later and in order to not lose visibility, broader time is used. Alert throttling was not set as it is important to alert about every distinct connection to the site with phishing or malicious content. Results of this correlation search are described in the Testing chapter.

Chapter 4

Testing

Testing chapter evaluates the design and realization done in the previous chapters. As mentioned in the analysis chapter, two distinct approaches have been used to evaluate proof of concept and the realization.

Correlation algorithms have been tested on large production data samples of the organization that operates in finance. They have found assets in the organization that had been communicating with the TOR network and couple of phishing accesses have been detected. These were critical events that must be further investigated. This has confirmed that the solution from this thesis can be used in real production environment. Unfortunately, due to the complexity of the organization, up to this date no further approval have been received and therefore these results can't be further detailed in this thesis.

The second testing approach consisted of creating a lab environment where such behavior can be simulated and tested. This testing environment was created in Amazon Web Services cloud infrastructure. This included Windows and Unix servers, Palo Alto Next Generation Firewall, Squid proxy and Universal Forwarders (see Figure 1.9). These environments are very expensive and therefore were limited only to above selected instanced to proof the concept and their goal wasn't mimic the real production data. Palo Alto Firewall and Squid must have been configured properly in order to log the required information. Configuration details are outside of the scope of this thesis.

Fully qualified domain name indicators have not been tested as Amazon Web Services logs are not available. However, domain names can be extracted from the URL with using the regular expression like:

```
http[s]*:\\/\\/[www.]*(?P<url\_domain>[\\w\\W\d]*?)[\\/\"]
```

However, as it is extracted from the proxy logs, it won't test any specific cases. For that, data from DNS server should be leveraged.

4.1 TOR

In order to test detection of the TOR traffic from the inside of the organization, TOR browser – the client, has been installed on the Windows Server. Connection to the publicly available URL have been done (see Figure 4.1).

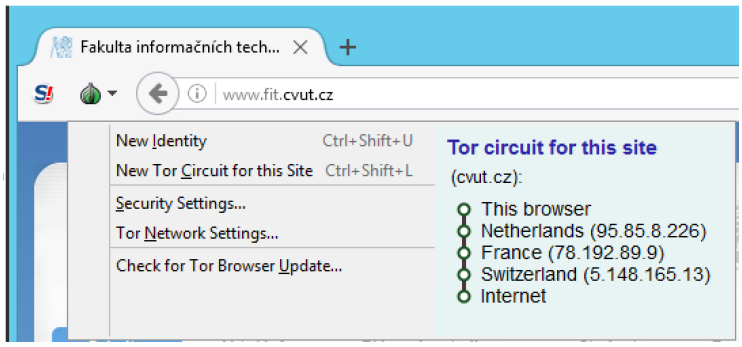


Figure 4.1: TOR circuit created to anonymize connection to the site

One can see that the first node of the TOR circuit is IP address 95.85.8.226 located in Netherland. To confirm detection of such traffic, Splunk alerts are reviewed and alert for this IP have been found (see Figure 4.2).

4.2 Ransomware

In order to test detection of the Ransomware communication from inside of the organization, simulated connections have been made. It was not possible to intentionally install real ransomware into the cloud environment. Connection to the malicious URL have been made (see Figure 2.9) from the lab environments[17,18]

Splunk alerts are reviewed and alert for this IP have been found (see Figure 4.3 4.4).

On the figure 4.3 it is possible to see that the correlation search has found the malicious connection to the *http://uuianqwjmdmn.eu/main.php*. Because this indicator was not derived, it has the full confidence equals nine.

On the figure 4.4 it is possible to see two types of situations. First one is displaying connection to the server that hosts the malicious URL from the figure 4.3. As this indicator is enriched, the confidence is only six. From this dashboard, it is possible to see that the server hosting *http://uuianqwjmdmn.eu/main.php* is also hosting another three sites that are reported as malicious. That helps incident response team to get the broader context to the event.

[17] http://uuianqwjmdmn.eu/main.php
[18] Connection to the server hosting malicious URL: 95.85.8.226

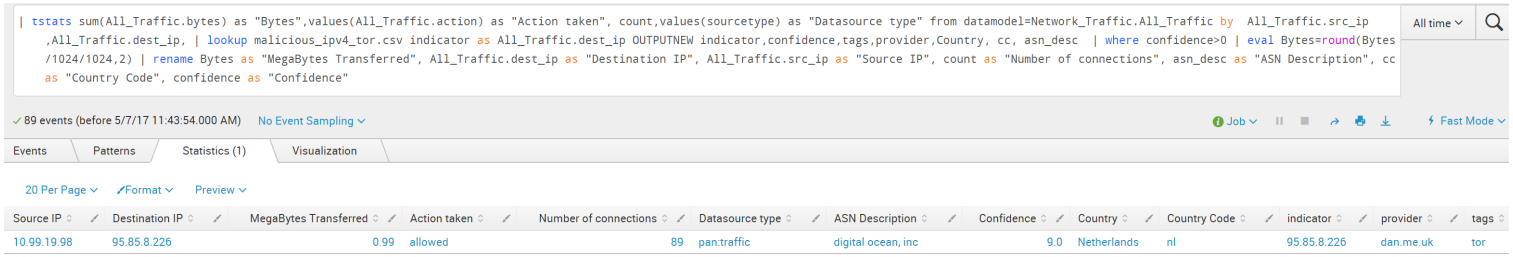
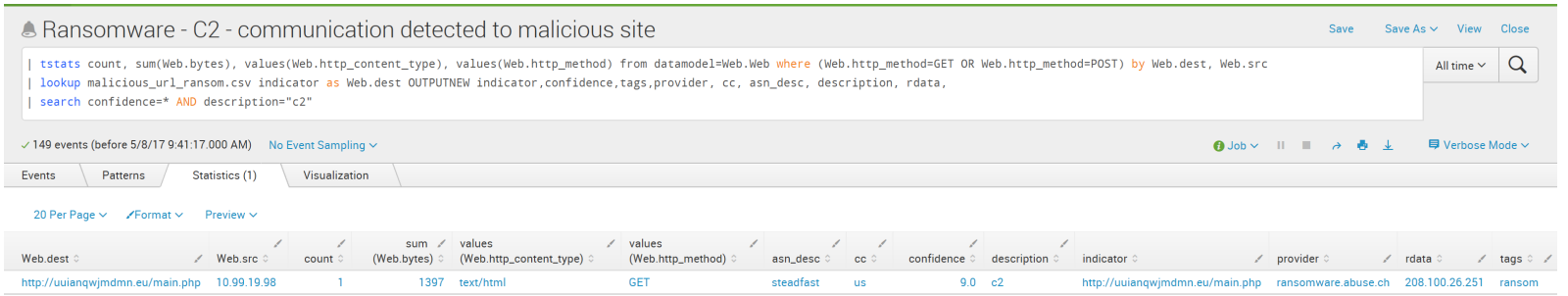


Figure 4.2: Results of the Splunk correlation search that has found outbound TOR traffic that we have simulated



Ransomware - C2 - communication detected to malicious site

Save | Save As | View | Close

```
| tstats count, sum(Web.bytes), values(Web.http_content_type), values(Web.http_method) from datamodel=Web.Web where (Web.http_method=GET OR Web.http_method=POST) by Web.dest, Web.src
| lookup malicious_url_ransom.csv indicator as Web.dest OUTPUTNEW indicator,confidence,tags,provider, cc, asn_desc, description, rdata,
| search confidence=* AND description="c2"
```

All time

✓ 149 events (before 5/8/17 9:41:17.000 AM) No Event Sampling

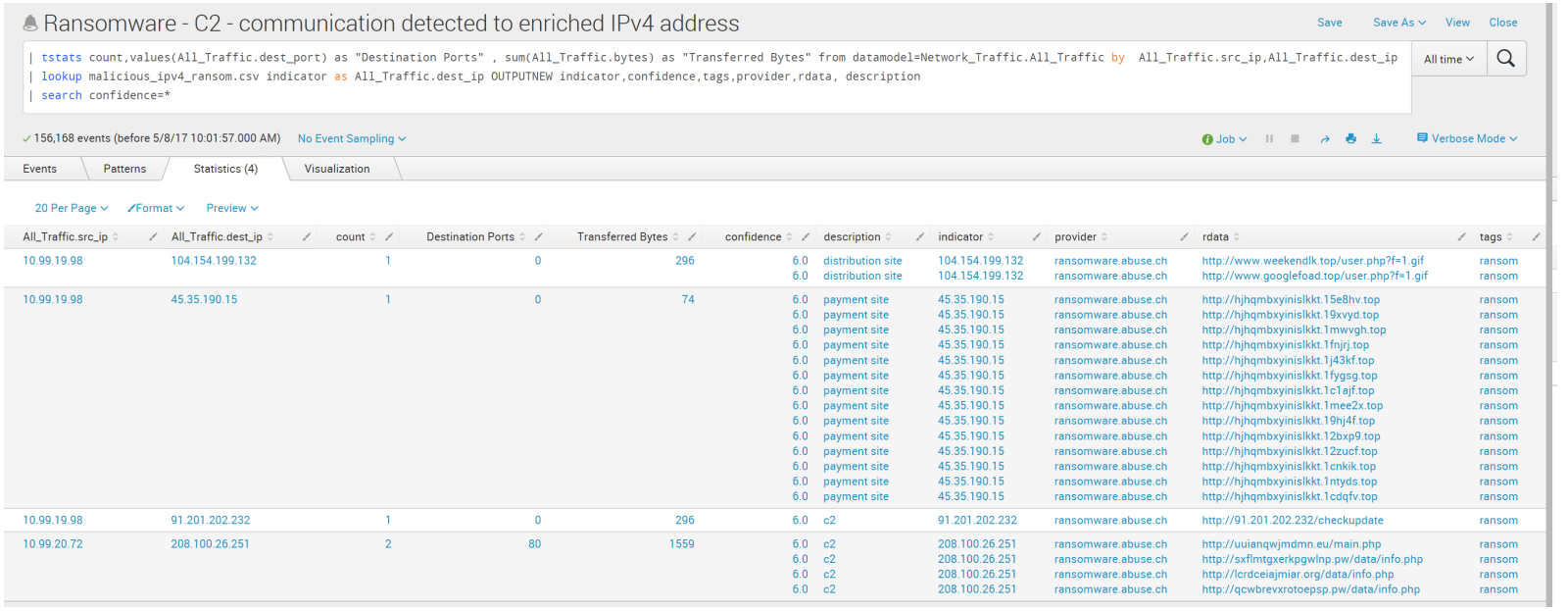
Job | Verbose Mode

Events | Patterns | Statistics (1) | Visualization

20 Per Page | Format | Preview

Web.dest	Web.src	count	sum (Web.bytes)	values (Web.http_content_type)	values (Web.http_method)	asn_desc	cc	confidence	description	indicator	provider	rdata	tags
http://uuianqwjmdmn.eu/main.php	10.99.19.98	1	1397	text/html	GET	steadfast	us	9.0	c2	http://uuianqwjmdmn.eu/main.php	ransomware.abuse.ch	208.100.26.251	ransom

Figure 4.3: Ransomware - C2 - communication detected to malicious site



Ransomware - C2 - communication detected to enriched IPv4 address

Save | Save As | View | Close

```
| tstats count,values(All_Traffic.dest_port) as "Destination Ports" , sum(All_Traffic.bytes) as "Transferred Bytes" from datamodel=Network_Traffic.All_Traffic by  All_Traffic.src_ip,All_Traffic.dest_ip
| lookup malicious_ipv4_ransom.csv indicator as All_Traffic.dest_ip OUTPUTNEW indicator,confidence,tags,provider,rdata, description
| search confidence=*
```

All time

✓ 156,168 events (before 5/8/17 10:01:57.000 AM) No Event Sampling

Job | Verbose Mode

Events | Patterns | Statistics (4) | Visualization

20 Per Page | Format | Preview

All_Traffic.src_ip	All_Traffic.dest_ip	count	Destination Ports	Transferred Bytes	confidence	description	indicator	provider	rdata	tags
10.99.19.98	104.154.199.132	1	0	296	6.0	distribution site	104.154.199.132	ransomware.abuse.ch	http://www.weekendlk.top/user.php?f=1.gif	ransom
					6.0	distribution site	104.154.199.132	ransomware.abuse.ch	http://www.googlefoad.top/user.php?f=1.gif	ransom
10.99.19.98	45.35.190.15	1	0	74	6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.15e8hv.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.19xvyd.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1mwvgh.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1fnjrj.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1j43kf.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1fygsg.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1c1ajf.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1mee2x.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.19hj4f.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.12bxp9.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.12zucf.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1cnkik.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1ntyds.top	ransom
					6.0	payment site	45.35.190.15	ransomware.abuse.ch	http://hjhqmbxyinislkkt.1cdqfv.top	ransom
10.99.19.98	91.201.202.232	1	0	296	6.0	c2	91.201.202.232	ransomware.abuse.ch	http://91.201.202.232/checkupdate	ransom
10.99.20.72	208.100.26.251	2	80	1559	6.0	c2	208.100.26.251	ransomware.abuse.ch	http://uuianqwjmdmn.eu/main.php	ransom
					6.0	c2	208.100.26.251	ransomware.abuse.ch	http://sxflmtgxerkpgwlnp.pw/data/info.php	ransom
					6.0	c2	208.100.26.251	ransomware.abuse.ch	http://lcrdceiajmiar.org/data/info.php	ransom
					6.0	c2	208.100.26.251	ransomware.abuse.ch	http://qcwbrevxrotoepsp.pw/data/info.php	ransom

Figure 4.4: Ransomware - C2 - communication detected to enriched IPv4 address

The second situation is the tested IP address *104.154.199.132* where it is possible to see that it hosts two URLs that are reported as malicious and therefore the communication with this server has been detected and alerted.

4.3 Phishing

In order to detect access to sites that are reported for their phishing or malicious activities, simulated connections have been made. Simulated phishing email was sent and the obfuscated URL hyperlink was opened. Redirection was made to the malicious site (see Figure 2.10). This site[19] has represented user accessing phishing website (see Figure 2.10).

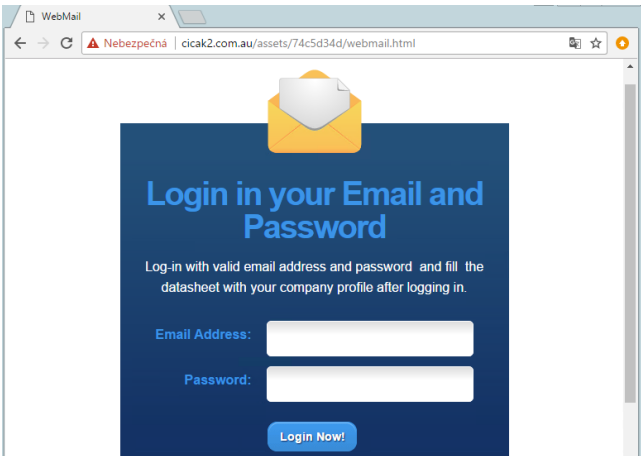


Figure 4.5: Simulated access to the phishing website

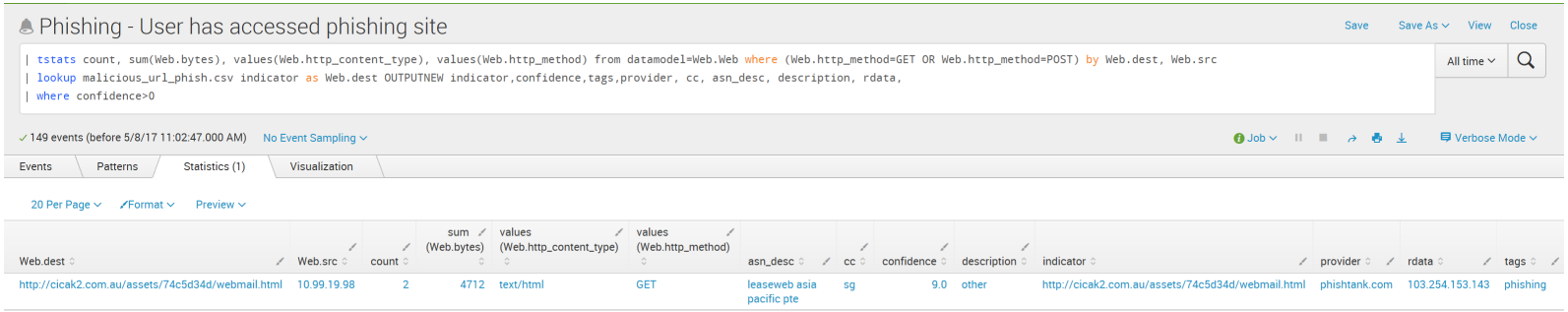
Correlation searches have found this attempt and triggered the alert. Alert is reviewed in the Splunk dashboard.

On the figure 4.6 it is possible to see that the correlation search has found the malicious connection to the [20]. Because this indicator was not derived, it has the full confidence equals nine.

On the figure 4.7 it is possible to see two types of situations. First one is displaying connection to the server that hosts the malicious URL from the figure 4.6. As this indicator is enriched, the confidence is only six.

The second situation displays a false positive situation. It confirms how important is to lower confidence when deriving new indicators from the original indicator. It is possible to see that communication to *216.58.216.174* have been alerted because this server is hosting multiple URLs that has been reported as malicious. However, this IP address belongs to the *Google Inc.* and their services have been abused to host malicious or phishing content.

[19] http://cicak2.com.au/assets/74c5d34d/webmail.html
[20] Refer to footnote 14



Phishing - User has accessed phishing site

Save | Save As | View | Close

```
| tstats count, sum(Web.bytes), values(Web.http_content_type), values(Web.http_method) from datamodel=Web.Web where (Web.http_method=GET OR Web.http_method=POST) by Web.dest, Web.src
| lookup malicious_url_phish.csv indicator as Web.dest OUTPUTNEW indicator,confidence,tags,provider, cc, asn_desc, description, rdata,
| where confidence>0
```

All time

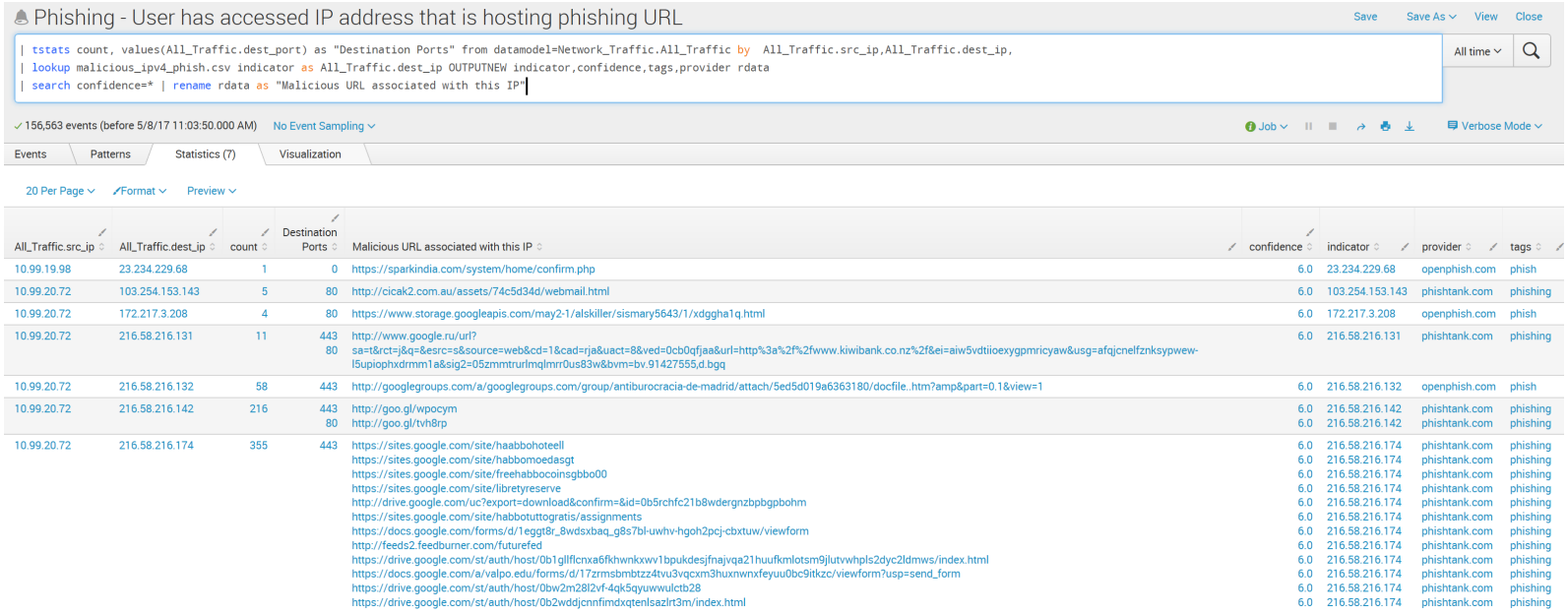
✓ 149 events (before 5/8/17 11:02:47.000 AM) | No Event Sampling | Job | Verbose Mode

Events | Patterns | Statistics (1) | Visualization

20 Per Page | Format | Preview

Web.dest	Web.src	count	sum (Web.bytes)	values (Web.http_content_type)	values (Web.http_method)	asn_desc	cc	confidence	description	indicator	provider	rdata	tags
http://cicak2.com.au/assets/74c5d34d/webmail.html	10.99.19.98	2	4712	text/html	GET	leaseweb asia pacific pte	sg	9.0	other	http://cicak2.com.au/assets/74c5d34d/webmail.html	phishtank.com	103.254.153.143	phishing

Figure 4.6: Phishing - User has accessed phishing site



Phishing - User has accessed IP address that is hosting phishing URL

Save | Save As | View | Close

```
| tstats count, values(All_Traffic.dest_port) as "Destination Ports" from datamodel=Network_Traffic.All_Traffic by  All_Traffic.src_ip,All_Traffic.dest_ip,
| lookup malicious_ipv4_phish.csv indicator as All_Traffic.dest_ip OUTPUTNEW indicator,confidence,tags,provider rdata
| search confidence=* | rename rdata as "Malicious URL associated with this IP"
```

All time

✓ 156,563 events (before 5/8/17 11:03:50.000 AM) | No Event Sampling | Job | Verbose Mode

Events | Patterns | Statistics (7) | Visualization

20 Per Page | Format | Preview

All_Traffic.src_ip	All_Traffic.dest_ip	count	Destination Ports	Malicious URL associated with this IP	confidence	indicator	provider	tags
10.99.19.98	23.234.229.68	1	0	https://sparkindia.com/system/home/confirm.php	6.0	23.234.229.68	openphish.com	phish
10.99.20.72	103.254.153.143	5	80	http://cicak2.com.au/assets/74c5d34d/webmail.html	6.0	103.254.153.143	phishtank.com	phishing
10.99.20.72	172.217.3.208	4	80	https://www.storage.googleapis.com/may2-1/alskiller/sismary5643/1/xdggha1q.html	6.0	172.217.3.208	openphish.com	phish
10.99.20.72	216.58.216.131	11	443 80	http://www.google.ru/url? sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0cb0qfjaa&url=http%3a%2f%2fwww.kiwibank.co.nz%2f&ei=aiw5vdtiioexygpmricyaw&usg=afqjcnelfznksypwew-l5upiophxdrmm1a&sig2=05zmmtrurlmqlmrr0us83w&bvm=bv.91427555,d.bgq	6.0	216.58.216.131	phishtank.com	phishing
10.99.20.72	216.58.216.132	58	443	http://googlegroups.com/a/googlegroups.com/group/antiburocracia-de-madrid/attach/5ed5d019a6363180/docfile..htm?amp&part=0.1&view=1	6.0	216.58.216.132	openphish.com	phish
10.99.20.72	216.58.216.142	216	443 80	http://goo.gl/wpocym http://goo.gl/tvh8rp	6.0 6.0	216.58.216.142 216.58.216.142	phishtank.com phishtank.com	phishing phishing
10.99.20.72	216.58.216.174	355	443	https://sites.google.com/site/haabbohoteell https://sites.google.com/site/habbomoedasgt https://sites.google.com/site/freehabbocoinsgbbo00 https://sites.google.com/site/libretyreserve http://drive.google.com/uc?export=download&confirm=&id=0b5rchfc21b8wdergnzbpbgpbohm https://sites.google.com/site/habbotuttogratis/assignments https://docs.google.com/forms/d/1eggt8r_8wdsxbaq_g8s7bl-uwhv-hgoh2pcj-cbxtuw/viewform http://feeds2.feedburner.com/futurefed https://drive.google.com/st/auth/host/0b1gllflcnxa6fkhwnkxwv1bpukdesjfnajvqa21huufkmlotsm9jlutvwhpls2dyc2ldmws/index.html https://docs.google.com/a/valpo.edu/forms/d/17zrmsbmbtzz4tvu3vqcxm3huxnwnxfeyuu0bc9itkzc/viewform?usp=send_form https://drive.google.com/st/auth/host/0bw2m28l2vf-4qk5qyuwwulctb28 https://drive.google.com/st/auth/host/0b2wddjcnnfimdxqtenlsazlrt3m/index.html	6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0	216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174 216.58.216.174	phishtank.com phishtank.com phishtank.com phishtank.com phishtank.com phishtank.com phishtank.com phishtank.com phishtank.com phishtank.com phishtank.com phishtank.com	phishing phishing phishing phishing phishing phishing phishing phishing phishing phishing phishing phishing

Figure 4.7: Phishing - User has accessed IP address that is hosting phishing URL

Conclusion

The goal of the thesis was to analyze the problems of threat intelligence. This has been done according to the best practices used in the industry and also using the information within academia. For the purpose of this thesis, the threat intelligence and all the related terms were defined using these resources.

According to the analysis, the suitable platform has been selected. The focus was on leveraging the platform that is used in the real-world environments. Furthermore, another focus was to keep the solution as open as possible. This has been achieved by using one of the leading Security Information and Event Management platform (Splunk) that at the same time is publicly available for the limited time or developer purpose. Another issue has arisen during the analysis, specifically how to collect, store, normalize and enrich threat intelligence data. There were two distinct approaches, either to create a solution from scratch or to leverage publicly available solution. The second approach was chosen in the form of Threat Intelligence Management platform that has been further integrated into the solution presented by this thesis. These platforms were integrated to cover the whole Threat Intelligence cycle (i.e., Requirements, Collection, Analysis, Production, Evaluation). The result of the integration provides the complex platform that enables utilization of the threat intelligence. This platform is based on tools that are open-source or free for certain period of time.

Another goal was to propose suitable correlation algorithms (searches) that would enhance the threat detection capabilities of the selected platform (Splunk). After analysis, following use cases have been selected:

- The Onion Router Traffic (TOR)
- Ransomware
- Phishing

The intention was to detect such communication, access or threat in the organization's various network traffic logs. These correlation algorithms were

designed and then integrated into the platform. Integrated correlation algorithms have been validated with the simulated data in the lab environment. The validation was successful – malicious traffic has been detected according to the use cases. On the top of the thesis requirements, correlation algorithms have been validated with data from the production environment. In this environment, TOR traffic and phishing attempts have been successfully detected.

Splunk Enterprise is free for limited use and it does not provide any detection capabilities as the ones mentioned above. This thesis provides the solution that can provide enhancement of platform capabilities. These enhancements are free to extend and opened for usage.

On the top of the thesis, additional goals were set by the author. That has included making the design and realization generic enough so its parts can be further used as proof of concept, template or guidance to further improvements. This was achieved by using a threat intelligence cycle model in all chapters. At the same time, the goal was to bring the solution with regards to the production requirements and trends in the cyber security. That was also achieved by setting proper requirements on the architecture and discussion with experts in various security fields.

Therefore, the solution brought in this thesis can be used as a whole to enhance detection capabilities using threat intelligence. However, if the organization already has parts of the threat intelligence cycle in place, parts of this thesis and concepts used here can be used as a template, guidance or proof of concept. This concept is currently being implemented into the real-world production environment as it was proved on the confidential data samples that can leverage threat intelligence and provide context.

In the future, the solution can be extended by defining and designing new correlation algorithms. It can be also enhanced by integrating another type of threat intelligence.

Bibliography

[1] Chismon, D.; Ruks, M. Threat Intelligence: Collecting, Analysing, Evaluating. 2015, [online] (visited on 08.05.17). Available from: https://www.mwrinfosecurity.com/assets/Whitepapers/Threat-Intelligence-Whitepaper.pdf

[2] Shackleford, D. Who's Using Cyberthreat Intelligence and How? feb 2015, [online] (visited on 08.05.17). Available from: https://www.sans.org/reading-room/whitepapers/analyst/cyberthreat-intelligence-how-35767

[3] Hutchins, E. M.; Cloppert, M. J.; et al. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, volume 1, 2011: p. 80.

[4] Kavanagh, K. M.; Rochford, O.; et al. Magic Quadrant for Security Information and Event Management. 2016, [online] (visited on 08.05.17). Available from: https://www.gartner.com/doc/reprints?id=1-2JNR3RU&ct=150720&st=sb

[5] Oxford University Press. Cyberthreat. [online] (visited on 08.05.17). Available from: https://en.oxforddictionaries.com/definition/cyberthreat

[6] Rouse, M. Confidentiality, integrity, and availability (CIA triad). nov 2014, [online] (visited on 08.05.17). Available from: http://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA

[7] Federal Bureau of Investigation. Intelligence Branch. [online] (visited on 08.05.17). Available from: https://www.fbi.gov/about/leadership-and-structure/intelligence-branch

[8] McMillan, R. Definition: Threat Intelligence. may 2013, [online] (visited on 08.05.17). Available from: https://www.gartner.com/doc/2487216/definition-threat-intelligence

[9] Kliarsky, A. Responding to Zero Day Threats. jun 2011, [online] (visited on 08.05.17). Available from: https://www.sans.org/reading-room/whitepapers/incident/responding-zero-day-threats-33709

[10] Aziz, A. The Evolution of Cyber Attacks and Next Generation Threat Protection. apr 2013, [online] (visited on 08.05.17). Available from: https://www.rsaconference.com/writable/presentations/file_upload/spo1-r31_spo1-r31_1_.pdf

[11] Cert-UK and GCHQ. Common Cyber Attacks: Reducing The Impact. 2015, [online] (visited on 08.05.17). Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/400106/Common_Cyber_Attacks-Reducing_The_Impact.pdf

[12] Sieber, J. Actionable Threat Intelligence: The Key to Efficient and Comprehensive Security. feb 2016, [online] (visited on 08.05.17). Available from: http://researchcenter.paloaltonetworks.com/2016/02/actionable-threat-intelligence-the-key-to-efficient-and-comprehensive-security/

[13] PhishMe. What is Actionable Intelligence? mar 2017, [online] (visited on 08.05.17). Available from: https://phishme.com/what-is-actionable-intelligence/

[14] SentinelOne. Insider Threats in Cyber Security—More Than Just Human Error. dec 2016, [online] (visited on 08.05.17). Available from: http://www.csoonline.com/article/3149754/security/insider-threats-in-cyber-security-more-than-just-human-error.html

[15] Nunes, E.; Diab, A.; et al. Darknet and Deepnet Mining for Proactive Cybersecurity Threat Intelligence. *CoRR*, volume abs/1607.08583, 2016. Available from: http://arxiv.org/abs/1607.08583

[16] Fachkha, C.; Debbabi, M. Darknet as a Source of Cyber Intelligence: Survey, Taxonomy, and Characterization. *IEEE Communications Surveys Tutorials*, volume 18, no. 2, Secondquarter 2016: pp. 1197–1227, ISSN 1553-877X, doi:10.1109/COMST.2015.2497690.

[17] Bianco, D. The Pyramid of Pain. jan 2014, [online] (visited on 08.05.17). Available from: http://detect-respond.blogspot.cz/2013/03/the-pyramid-of-pain.html

[18] Emmett Koen, J. M. Indicators of Compromise and where to find them. feb 2017, [online] (visited on 08.05.17). Available from: https://blogs.cisco.com/security/indicators-of-compromise-and-where-to-find-them

[19] Andress, J. Working with Indicators of Compromise. may 2015, [online] (visited on 08.05.17). Available from: https://c.ymcdn.com/sites/www.issa.org/resource/resmgr/journalpdfs/feature0515.pdf

[20] MANDIANT Corporation. OpenIOC. [online] (visited on 08.05.17). Available from: http://www.openioc.org/

[21] Kissel, R. L. Glossary of Key Information Security Terms. *NIST Interagency/Internal Report (NISTIR)*, [online] (visited on 08.05.17). Available from: https://www.nist.gov/publications/glossary-key-information-security-terms-1

[22] Sanders, C.; Smith, J.; et al. *Applied network security monitoring*. Syngress, an imprint of Elsevier, first edition, 2014.

[23] Liska, A.; Gallo, T. *Building an intelligence-led security program*. Syngress is an imprint of Elsevier, first edition, 2015.

[24] Bromiley, M. Threat Intelligence: What It Is, and How to Use It Effectively. sep 2016, [online] (visited on 08.05.17). Available from: https://www.sans.org/reading-room/whitepapers/analyst/threat-intelligence-is-effectively-37282

[25] Farnham, G. Tools and Standards for Cyber Threat Intelligence Projects. oct 2013, [online] (visited on 08.05.17). Available from: https://www.sans.org/reading-room/whitepapers/warfare/tools-standards-cyber-threat-intelligence-projects-34375

[26] MITRE Corporation. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX). jan 2012, [online] (visited on 08.05.17). Available from: https://www.mitre.org/sites/default/files/publications/stix.pdf

[27] Hovor, E.; Modi, S.; et al. Unstructured Threat Intelligence Processing using NLP. 2014, [online] (visited on 08.05.17). Available from: https://www.blackhat.com/docs/us-15/materials/us-15-Hovor-UTIP-Unstructured-Threat-Intelligence-Processing.pdf

[28] Cunningham, T. *A Cyber-Threat Intelligence Program – How to develop one and why it matters*. Master's thesis, Luleå University of Technology Department of Computer science, Electrical and Space engineering, 2015, [online] (visited on 08.05.17).

[29] Rekhter, Y.; Rekhter, Y.; et al. Address Allocation for Private Internets. feb 1996, [online] (visited on 08.05.17). Available from: https://tools.ietf.org/html/rfc1918

[30] Gartner, Inc. Security Information and Event Management (SIEM). [online] (visited on 08.05.17). Available from: http://www.gartner.com/it-glossary/security-information-and-event-management-siem/

[31] The Tor Project. Tor: Overview. [online] (visited on 08.05.17). Available from: https://www.torproject.org/about/overview.html.en

[32] Brown, D. Resilient Botnet Command and Control with Tor. jul 2010, [online] (visited on 08.05.17). Available from: https://www.defcon.org/images/defcon-18/dc-18-presentations/D.Brown/DEFCON-18-Brown-TorCnC.pdf

[33] Constantin, L. Cybercriminals are using the Tor network to control their botnets. 2013, [online] (visited on 08.05.17). Available from: http://www.pcworld.com/article/2045183/cybercriminals-increasingly-use-the-tor-network-to-control-their-botnets-researchers-say.html

[34] Hutzler, D. Malware Spread Via Tor Exit Node. 2014, [online] (visited on 08.05.17). Available from: https://www.opswat.com/blog/malware-spread-tor-exit-node

[35] Prabhu, V. This malware uses Tor to open a backdoor on Mac OS X. 2016, [online] (visited on 08.05.17). Available from: https://www.techworm.net/2016/07/malware-uses-tor-open-backdoor-mac-os-x.html

[36] Check Point Software Technologies Ltd. Ransomware Doubled in Second Half of 2016. feb 2017, [online] (visited on 08.05.17). Available from: https://www.checkpoint.com/press/2017/ransomware-doubled-second-half-2016-says-check-point/

[37] Mehmood, S. Survival Guide for Ransomware Attacks. apr 2016, [online] (visited on 08.05.17). Available from: https://www.sans.org/reading-room/whitepapers/incident/enterprise-survival-guide-ransomware-attacks-36962

[38] Avast Software s.r.o. A closer look at the Locky ransomware. mar 2016, [online] (visited on 08.05.17). Available from: https://blog.avast.com/a-closer-look-at-the-locky-ransomware

[39] BBC. University pays $20,000 to ransomware hackers. 2016, [online] (visited on 08.05.17). Available from: http://www.bbc.com/news/technology-36478650

[40] Cole, E. Detect, Contain and Control Cyberthreats. jun 2015, [online] (visited on 08.05.17). Available from: https://www.sans.org/reading-room/whitepapers/analyst/detect-control-cyberthreats-36187

[41] Rochford, O.; Kavanagh, K. M.; et al. Critical Capabilities for Security Information and Event Management. 2016, [online] (visited on 08.05.17). Available from: https://www.gartner.com/doc/reprints?id=1-2Q17LAL&ct=151019&st=sb

[42] Splunk Inc. Free vs. Enterprise. 2017, [online] (visited on 08.05.17). Available from: https://www.splunk.com/en_us/products/splunk-enterprise/free-vs-enterprise.html

[43] Splunk Inc. Splunk Licensing for Academic Instruction and Research. 2017, [online] (visited on 08.05.17). Available from: https://www.splunk.com/en_us/solutions/industries/higher-education/academic-licenses.html

[44] Splunk Inc. Splunk Developer License Signup. 2017, [online] (visited on 08.05.17). Available from: http://dev.splunk.com/page/developer_license_sign_up/

[45] Poputa-Clean, P. Automated Defense Using Threat Intelligence to Augment Security. jan 2015, [online] (visited on 08.05.17). Available from: https://www.sans.org/reading-room/whitepapers/threats/automated-defense-threat-intelligence-augment-35692

[46] European Union Agency for Network and Information Security. *Standards and tools for exchange and processing of actionable information.* First edition, 2014. Available from: https://www.enisa.europa.eu/publications/standards-and-tools-for-exchange-and-processing-of-actionable-information/at_download/fullReport

[47] Gartner, Inc. Gartner IT Glossary : integration. 2016, [online] (visited on 08.05.17). Available from: http://www.gartner.com/it-glossary/integration

[48] Tsaousis, C. All Cybercrime IP Feeds. 2016, [online] (visited on 08.05.17). Available from: http://iplists.firehol.org/

[49] Csirtgadgets. The CIF Book. 2016, [online] (visited on 08.05.17). Available from: https://github.com/csirtgadgets/massive-octo-spice/wiki/The-CIF-Book

[50] Palo Alto Networks, Inc. 2016, [online] (visited on 08.05.17). Available from: https://www.paloaltonetworks.com/

[51] Brown, D. Resilient Botnet Command and Control with Tor. 2010, [online] (visited on 08.05.17). Available from: https://www.defcon.org/images/defcon-18/dc-18-presentations/D.Brown/DEFCON-18-Brown-TorCnC.pdf

[52] Squid. 2016, [online] (visited on 08.05.17). Available from: http://www.squid-cache.org/

[53] Splunk Inc. Configuration file precedence. 2017, [online] (visited on 08.05.17). Available from: http://docs.splunk.com/Documentation/Splunk/6.5.3/Admin/Wheretofindtheconfigurationfiles

[54] Sorkin, S. Large-Scale, Unstructured Data Retrieval and Analysis Using Splunk. An Easier, More Productive Way to Leverage the Proven MapReduce Paradigm. 2011, [online] (visited on 08.05.17). Available from: https://www.splunk.com/web_assets/pdfs/secure/Splunk_and_MapReduce.pdf

[55] Splunk Inc. Splunk accelerated datamodels. 2017, [online] (visited on 08.05.17). Available from: http://docs.splunk.com/Documentation/Splunk/6.5.3/Knowledge/Acceleratedatamodels

[56] Splunk Inc. Splunkbase. 2017, [online] (visited on 08.05.17). Available from: http://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutsubsearches#Subsearch_performance_considerations

Appendix A

Contents of CD

```
readme.txt ..... the file with CD contents description
master_thesis.zip .. Zip archive with the LaTex source codes and PDF
threatintel.tar.gz ..... Splunk ThreatIntel app
installSplunk.sh ..... Splunk installation script
installUF.sh ..... Splunk Universal Forwarder installation script
manual.pdf  Splunk platform connection guidance with access credentials
```