



A Self Organized Threat Intelligence Architecture for Intrusion Detection Systems

by
DGCP Piyasena (168255E)

A thesis submitted to University of Moratuwa in partial fulfilment of the requirements for
the
Master of Computer Science, *Specialized in Security Engineering*

Department of Computer Science & Engineering
University of Moratuwa, Sri Lanka

February 2020



A Self Organized Threat Intelligence Architecture for Intrusion Detection Systems

by
DGCP Piyasena (168255E)

A thesis submitted to University of Moratuwa in partial fulfilment of the requirements for
the
Master of Computer Science, *Specialized in Security Engineering*

Department of Computer Science & Engineering
University of Moratuwa, Sri Lanka

February 2020

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

DGCP Piyasena:

Date

Approved by:

Lt Col Dr Chandana D. Gamage
Department of Computer Science and Engineering
University of Moratuwa

Date

Copyright Statement

I hereby grant the University of Moratuwa the right to archive and to make available my thesis or dissertation in whole or part in the University Libraries in all forms of media, subject to the provisions of the current copyright act of Sri Lanka. I retrain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

DGCP Piyasena

Date

I have supervised and accepted this thesis/dissertation for the award of the degree.

Lt Col Dr Chandana D. Gamage
Department of Computer Science and Engineering
University of Moratuwa

Date

Abstract

An Intrusion Detection System (IDS) is a software application that monitor a corporate network or a computer system and flag activities which it construes to be malicious operations. The rapid and expansive growth of Internet has heightened concerns on how to protect both stored and transmitted digital information in an effective manner.

The reactive IDS will primarily detect intrusions and send out alerts. Defending the system is a secondary task, and its success depends on how early detection can occur when an intrusion is ongoing so that warnings can be sent in time. IPS, which is mainly proactive, will primarily detect vulnerabilities and take preventive measures in addition to providing the second stage functionality for an IDS but with limited knowledge and countermeasure capabilities.

As a solution to this problem, research has been conducted on an area called Automated Defense. The design of Automated Defense systems needs to be radically different from the IDS/IPS schemes as properties such as on-line real-time availability of all participants, use of threat intelligence schemes, availability of high computation power, etc have to be considered. Taking into consideration the context in which Threat Intelligence Architecture operates, where transaction value is very low, IDS/IPS systems need to be designed with a careful trade-off between reliability and cost of implementation.

The research presented in this thesis aims to develop a solution to the problem of providing the functionality of an IDS with an IPS capability that is highly responsive, adaptive and able to leverage the most up-to-date knowledge on dealing with threats. The main objective of the research is to combine an IDS with Threat Intelligence in a manner that can detect file creations and copying anomalies and provide the mechanisms to alert and initiate actions to take defensive measures to decrease the potential for damage from attackers.

The main objective of the research is to combine with Threat Intelligence to provide a mechanism to alert and initiate actions to take defensive measures to decrease the potential for damage.

Acknowledgements

First of all I would like to thank my supervisor Dr. Chandana Gamage whose encouragement, guidance, support, and criticism from start to the very end, allowed me to understand the objectives and challenges of a master degree thesis. I would also like to thank Dr. Shehan Perera (Project Coordinator) for extensive advice, helpful feedback, and constant support.

Furthermore, my special thanks go to Dr. Shantha Fernando who provided an excellent, supporting, innovative, and inspiring environment in which it was a pleasure to create this thesis. Last but not its a pleasure to thank all my CiTes colleagues those who helped to make my thesis in a motion. I would also like to express my sincere gratitude to Mr Tim Crothers from Vice President Security Solutions at Target Minneapolis, Minnesota for the technical expertise provided.

Finally, words alone cannot express the thanks I owe Mr. DG Piyasena my father, Mrs. Pathma Swarnalatha my mother for all the encouragement extended.

Abbreviations

IDS - Intrusion Detection Systems
IPS - Intrusion Prevention Systems
TI - Threat Intelligence
CTI - Cyber Threat Intelligence
SOC - Security Operation Center
SIEM - Security information and event management
openIOC - Open Indicators of Compromise
STIX - Structured Threat Information Expression
CybOX - Cyber Observable
CybOX - Cyber Observable
TAXII - Trusted Automated Exchange of Indicator Information
IODEF - The Incident Object Description and Exchange Format
TIE - Threat Intelligence Exchange
TISP - Threat Intelligence Sharing Platform
HTTP - Hypertext Transfer Protocol
TPM - Trusted Platform Module
VM - Virtual Machine
VME - Virtual Machine Environment
VMM - Virtual Machine Monitor
NIDS - Network Based Intruder Detection Systems
HIDS - Host Based Intruder Detection Systems
PIDS - Physical Intruder Detection Systems
OSSIM - Open Source Security Information Management

Table of Contents

Declaration	v
Copyright Statement	vi
Abstract	vii
Acknowledgements	viii
Abbreviations	ix
List of Tables	xiii
List of Figures	xiv
1 Introduction	1
1.1 Background	1
1.2 Research Problem	3
1.3 Objective	5
1.4 Methodology	6
1.5 Summary	7
2 Literature Review on Intrusion Detection Systems	9
2.1 Introduction	9
2.2 History and Evolution	10
2.3 Architecture of the IDS	11
2.4 Detection Approaches	16
2.4.1 Misuse Detection	17
2.4.2 Pattern Matching	18

2.4.3	Rule-based Techniques	18
2.4.4	State-based Techniques	18
2.4.5	Anomaly Detection	19
2.4.6	Use of Honeypots in Intruder Detection Systems	20
2.5	Intrusion Prevention Systems (IPS)	22
2.5.1	Rate-based IPS	22
2.5.2	Disadvantages of Rate-based IPS	22
2.5.3	Content-based Products	22
2.6	Threat Intelligence	24
2.6.1	Current Threat Intelligence Definition	24
2.6.2	Types of Threat Intelligence	25
2.6.3	Threat Intelligence Platform Capabilities	25
2.6.4	Cyber Threat Intelligence Challenges	27
2.6.5	Attack Vector Reconnaissance	27
2.6.6	Attack Indicator Reconnaissance	27
2.6.7	Cyber Threat Intelligence Opportunities	28
2.7	Threat Sharing Platform	28
2.8	Enabling Automated Responses from Policy	29
2.8.1	Security Policies	29
2.8.2	Engineering and Enforcing Security Policies	30
2.9	Threat Hunting	31
2.9.1	Intel Sources	31
2.9.2	Importance of Detection	32
2.10	Summary	33
3	Methodology for Cyber TI Support to Incident Handling	34
3.1	Introduction	34
3.2	Requirement 1: Sharing Mechanism of the Cyber-Trust Platform	35
3.3	Requirement 2: Expressibility, Flexibility, & Scalability of TI . .	36
3.4	Requirement 3. Information Used to Facilitate Automation . . .	37
3.5	Requirement 4. Streamlining of Hunting and Incident Response Process	38
3.6	Summary	40
4	Solution Framework for Detection and Analysis	41
4.1	Introduction	41

4.2	Tactics Used for Threat Intelligence	41
4.3	Automation of the Threat Intelligence	43
4.4	Building Maturity Model	46
4.5	Summary	49
5	Simulation of TI Automation and Event Sharing	50
5.1	Introduction	50
5.2	Threat Intelligence	50
5.3	Threat Hunting	58
5.4	Summary	63
6	Analysis of Automated Threat Intelligence Architecture	65
6.1	Introduction	65
6.2	Key Findings	66
6.2.1	Key Finding 1: On TI Sharing Platforms	66
6.2.2	Key Finding 2: On Sharing of Indicators of Compromise	67
6.2.3	Key Finding 3 : On TI Platform Availability	67
6.2.4	Key Finding 4 : On TI Platform Availability	67
6.2.5	Key Finding 5 : On Lack of Automation	68
6.3	Discussion of Results	69
6.4	Summary	72
7	Conclusions	73
7.1	Introduction	73
7.2	Observation from CTI	74
7.3	SOAR vs Proposed Framework	75
7.4	Data Feed Providers	76
7.5	Future Work	76
7.6	Summary	78
	References	79

List of Tables

3.1	Threat Intelligence use cases	39
4.1	Confidence matrix for measurable decision making	42
4.2	Confidence-based actions	42
5.1	Security Tools Intel Integration	55

List of Figures

2.1	Structure of Intrusion Detection System	9
2.2	Architecture of IDS	12
2.3	Architecture of IDS with sensor	13
2.4	Console fault tollerance implementation	14
2.5	Hierarchy management of IDS	15
2.6	Three level sensor management scheme	16
2.7	Structure of Misuse Detection	17
2.8	State transition diagrams	19
2.9	Anomaly detection model	19
3.1	David Bianco - Pyramid of Pain	37
4.1	Initializing Intel Feeds on devices	43
4.2	Anecdotal data by using augmented default feeds	44
4.3	Central collection of Intelligence	46
4.4	Centralized Storage and Collection	47
4.5	Centralize intelligence collection with automated action	48
5.1	CRITs observable details	51
5.2	CRITs Analysis Services	52
5.3	Relationships in CRITs	53
5.4	QRadar intel-based rules	53
5.5	Flow of the the monitoring	56
5.6	QRadar Context Menu	58
5.7	MANTIS first run	59
5.8	MANTIS import data	60
5.9	MANTIS User Interface	61
5.10	MANTIS Observables	62

5.11 MANTIS Hashes	63
5.12 MANTIS alert generation	64
5.13 MANTIS with virus total	64

Chapter 1

Introduction

Intruder is defined as an entity, program or person that used to break into an information system or perform an action or threat that is not legally permitted, and may become effective. It refers to intrusion as any collection of acts that seek to compromise the privacy , confidentiality or availability of a computer resource. The act of detecting behavior that threaten the privacy , confidentiality or availability of a computer resource can be referred to as detection of intrusion. A monitoring system for intrusion is a tool or software program that detects network and/or system activities for malicious activities or policy violations and reports.

1.1 Background

Maintaining high-level protection is now necessary to ensure safe and trustworthy communication of information between various organisations [8]. Yet protected, Internet and any other network data communication is still under threat of intrusions and misuse. Given that malicious intrusions into computer systems have become a growing problem, the need to detect these intrusions accurately has increased. This research approach to identify these intrusions using a sophisticated, known method of threat intelligence applied to an Intrusion Detection System.

There are currently various approaches to prevent unauthorized intruders

(e.g., firewalls, password-protected systems) [8]. However, because of their obvious problems is these traditional methods are becoming increasingly vulnerable and inefficient. Today's company needs to develop a robust infrastructure, a complex structure that can provide sufficient protection today but that's versatile enough to defend against tomorrow's unknown threats and new technology. It will create a systematic protection and enforcement plan, including the selection of the security architecture to guide the building method and the maturity model to help to advance the security program with the knowledge of the threat intelligence platform.

Firewall is the most widely implemented solution to protect corporate assets against intrusions, but firewalls are vulnerable to configuration errors in undefined data-driven security policy attacks by approved services and insider attacks. The firewall failure was boon to commercial intrusion detection tools to adequately protect digital assets against computer-based attacks. Misuse detection and identification of anomalies are two common ways of detecting intrusions into computer security in real time [8].

When attackers consistently behave when performing attacks, that particular behavior is considered a signature of the specific combination of attack or attack-attacker. This is called anomaly detection based on signature. Expect users to behave fairly consistently and stably, too. If their behavior varies from this normal practice, consider it to be an anomaly and probably the action of an attacker and not the actual user. This is called identification of intrusion based on anomaly.

However, attackers are constantly trying to hide from intrusion detection systems either by varying their attack behavior significantly (SBID becomes ineffective) or by ensuring that their attack behavior is not significantly different from actual user behavior (so that ABID becomes ineffective).

It needs to counter the attacker in this sense by studying how the attacker tries to disguise himself and hide from the SBID or ABID systems. For SBID systems, the attacker can continue to change the patterns associated with the attack that it executes to ensure there is no corresponding stable signature. The detection systems in such a situation did not mean the attacker action

was exploited.

However, in the ABID scenario, as the attacker continually attempts to adapt to the target user behaviour, the detection systems can manipulate the attacker indirectly by instructing the target user to deliberately make detectable changes to his behaviour.

1.2 Research Problem

The majority of current IDSs suffer at least from the following problems.

The derived information from the IDS is used by getting network's audit trails or packets. Those collected data must follow a longer path from its origin to the IDS and it is potentially disrupted or modified by an attacker during the process. In addition, the system must work out with the system's actions from the data obtained, which may result in misinterpretations or missing events being referred to as the fidelity issue..

Second, the method of intrusion detection constantly requires external resources within the network. It is monitoring even when there are no intrusions, because the component which is belong to the ids work all the time. Since that it may lead to the issue of resource utilization.

The component which is belong to the intrusion detection system implemented in separate way. This may lead to the vulnerable to the interference. Attacker or intruder can interfere the program which are currently running. That is the issue of reliability.

Detection of intrusion has the potential to alleviate many of the issues currently facing network security and scaling to large, fast, and complex systems. Most of the existing ID systems are Basically monolithic. More distributed architecture is required to respond effectively to the large-scale attacks. Misuse detection systems rely heavily on getting models of new attacks to counter attacks like the Code Red worm, the delivery of these models needs to out-

strip the outbreak like a worm spread. Subtle recognition. Many existing ID systems are complex to configure and operate, so their use is restricted.

Strong reactive skills. It is obvious from the that it is infeasible to rely on human action to deal with attacks. Most of the existing IDS implementations have restricted reactive capabilities, and an IDS needs to be able to avoid, not just record the attack. While anomaly-based detection systems have the potential to provide successful defense, it is important to address two key problems. First, learning-based systems for detecting anomalies are prone to produce a large number of false positives. Second, unlike misuse-based systems, anomaly detection systems only record an anomaly without any supporting explanation of the detected attack.

IDS, which is reactive, will primarily detect intrusions and send alerts. Defending the system is a secondary task, and its success depends on how early detection can happen when an intrusion is in progress so that timely alerts can be sent. IPS, which is mainly proactive, will primarily detect vulnerabilities and take preventive action in addition to providing the second stage functionality for an IDS but with limited knowledge and capabilities on countermeasures.

The research problem is how to provide the functionality of an IDS with an IPS capability that is highly responsive, adaptive and able to leverage the most up-to-date knowledge on dealing with threats.

1.3 Objective

Once An insider has got out of control, the next phase is to start searching out where they can find data or weaknesses that they can exploit to cause network damage if their goal is to sabotage IT. The research's main objective is to combine these file creations and copying phenomena with Threat Intelligence, which provides the basis for alerting which implementing actions to take protective measures to minimize the potential for harm.

The objective of this research are to

- Enable open-source IDS platforms to enhance their architectures and capabilities to evaluate suitability for adoption as a platform for the proposed solution.
- Create script to take action on alerts and improve script to take response automatically

Internet threats continue to increase, and this causes harm to our security network. For that this threat, a security infrastructure must be in place which has the flexibility for discover and block zero-day attacks. Honeypot is the proactive defense technology in which resources are placed in an extremely network for the purpose of observing and capturing new attacks.

Learning-based techniques are particularly suitable for detecting web attacks as they can detect attacks against custom code developed for which no known signatures or models of attacks exist. These systems also operate in unsupervised mode, with little or no input from system administrators and developers of applications. Consequently, administrators with little safety training can use them.

1.4 Methodology

As a solution to this problem, research has been conducted on an area called A Self-organized threat intelligence architecture for intrusion detection systems. The design of that kind of system needs to be radically different from the intruder detection schemes as properties such as online real time. That is common for intruder detection systems. Taking into consideration the context in which intruder detection schemes operate, where transaction value is very high, such kind of intruder detection systems need to design with a carefully between reliability and cost of implementation. Our approach is to build Threat Intelligence Architecture For Intruder Detection System for available intruder detection systems as follows:

- A experimentation of open source IDS platforms to review their architectures and operational capabilities to evaluate suitability for adoption as a platform for the proposed solution.
- A experimentation of open source threat intelligence sources to review their architectures and operational capabilities to evaluate suitability for integration in a scalable manner to the proposed solution.
- Experimentation of existing security solutions (software, hardware and hybrid) to see how they can be enabled, adapted or extended to make use of threat intelligence feeds.
- Experimentation of mechanisms for collection and management of threat intelligence in a centralized manner.
- Development of an overall integrated architecture.
- Implementation of a prototype test system to evaluate the proposed solution.

1.5 Summary

The rest of this thesis is structured as follows :

- **Chapter 2: Literature Review on Intruder Detection systems**
This chapter introduces the reader to Intruder Detection Systems. It describes the IDS and IPS models, deployment methods, security challenges of IDS and fundamental security challenges faced by IDS and IPS. To provide a more concrete knowledge, this chapter further describes the IDS reference model also.
- **Chapter 3 : Methodology of Cyber Threat Intelligence Support to Incident Handling**
This section presents the methodology used for providing recommendations on the CTI sharing aspects of the Cyber-Trust project by relying on a set of high-level requirements, as described in the description of action (DoA), and by considering the findings of research on the current situation on CTI sharing and automation.
- **Chapter 4: Solution Framework for Detection and Analysis**
This chapter focuses on tactical threat intelligence implementation. The proposed framework will work according to cases of tactical threat intelligence being used in general. The important aspect is implementing actionable integration of the Threat Intelligence platform with defensive systems.
- **Chapter 5: Simulation of TI Automation and Event Sharing**
This section simulate the proposed framework scheme using open source tools. The proposed framework totally comply with the tool used for the simulation process, it is assumed that the correctness of the threat intelligence guaranteed independently according to the tools used. This chapter is to prove to enable automated defensive systems with threat Intelligence platform integration and chapter describe complex cyber threat intelligence and sharing methods.

- **Chapter 6: Analysis of Automated Threat Intelligence Architecture**

This chapter analyzes operational level of intelligence mainly concerned with helping certain individuals and teams on the 'front lines' of defending the network which is proposed by the research.

- **Chapter 7: Conclusions**

This chapter contains a summary and discussion of the limitations and shortcomings of this approach. At the end of this chapter, conclude the thesis by summarizing the results and introducing possible future works.

Chapter 2

Literature Review on Intrusion Detection Systems

2.1 Introduction

This chapter introduces the Intrusion Detection Systems. It describes the IDS and IPS models, deployment methods, security challenges of IDS and fundamental security challenges faced by IDS and IPS provide more concrete knowledge, and this chapter further describes the IDS reference model also. Figure 2.1 elaborates on the basic structure of the intrusion detection system.

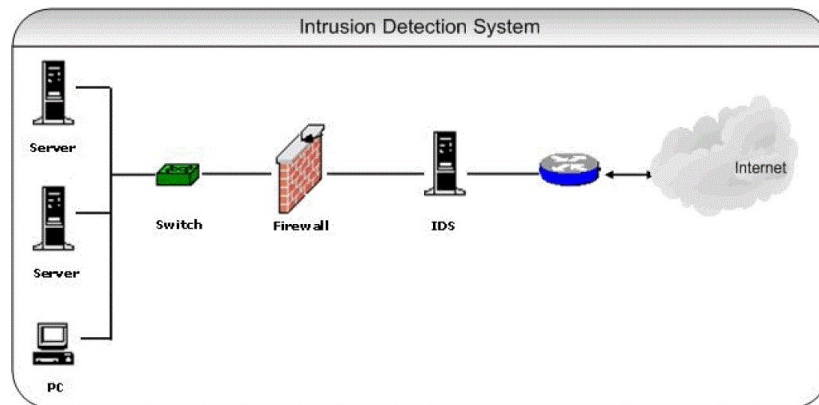


Figure 2.1: Structure of Intrusion Detection System

Detection of intrusion can be defined as the detection of acts that attempt to compromise the confidentiality, integrity or availability of a resource. More precisely, the purpose of intrusion detection is to recognize entities that attempt to subvert existing security controls. IDS can differentiate between where it is detected (network or host) and the detection method used [11].

1. Network Based (Network IDS)

NIDS is basically focused on network traffic that is identify the unauthorized, illegal, and anomalous behavior of that traffic. A NIDS inspect the packet on a network which is traverse through on any network device. The installed IDS processes and it flag on the suspicious data. Unlike an intrusion prevention system, network traffic is not deliberately blocked by an intrusion detection system. The function of an IDS network is passive and only collects, recognizes, logs and alerts.

Examples of Network IDS: SNORT

2. Host Based (HIDS)

HIDS are in attempts to recognize activities which include unwanted, illegal, and anomalous on a single computer are also referred to as HIDS. HIDS generally installed the agent onthe particular system and it will monitoring local OS and application activity and alerting them. To detect illegal behavior the mounted agent uses a combination of signatures, rules, and heuristics. A host IDS has a passive role, collecting, identifying, logging, and alerting only.

Examples of HIDS:OSSEC

2.2 History and Evolution

Before development of modern IDS, intrusion detection consisted of a manual scan for anomalies. Log files analyze incidents that may or should not occur during normal machine and networking action. Manual implementation of that system is difficult but also time consuming and need more manpower to execute. Therefor there was a need of developing automated log file readers to overcome this issue. This process need to automate whole process for a more comprehensive investigation [5]. Out of these anomalies it was possible to

derive patterns of attack with further study. So the first automatic, pattern matching readers of log files were created [5].

The concept was built with development of the cyber security industry. New network based intrusion detection which follow rules to match the pattern as a concept developed as host based detection. That is not look at the log file of particular host, but that is track the network traffic which is looking for the pattern of attackers by looking at the TCP/IP packets. Detection of result of the log file analysis is post factor. That will allowing forensic analysis with possible adaptations of infrastructure relatively long after the actual event. Since adequate processing speed is available, it has now become possible not only to check for attack patterns after the incident occurred, but also to monitor and trigger alerts in real time if intrusions were detected. Because of customer demand, the IT security industry has now begun turning former prototype software into actual intrusion detection systems, consisting of user-friendly interfaces, methods to update attack patterns, various alert methods and even some automatically triggered reactions or actual prevention methods, which can avoid attacks in progress [5].

2.3 Architecture of the IDS

Both systems for intrusion detection can be categorized as members of one of two categories: standalone or client server systems. Resources associated with the first category collect, evaluate and respond to events on a single host. Second-category systems built according to a different principle. The intrusion detection modules (sensors) are mounted at the company network's most sensitive points. These modules detect and react upon attacks. All management tasks performed from the centralized console to which all alerts are forwarded. This intrusion detection system 's architecture is relatively basic. It shows up in Fig. 2.2. It includes seven modules which are each responsible for a particular mission. The data-source processing module is responsible for collecting data in connection with the log file, network adapter, or OS kernel, on the basis of which the system defines an attack presence. The second module handles all the intrusion detection components and organizes their interaction [9].

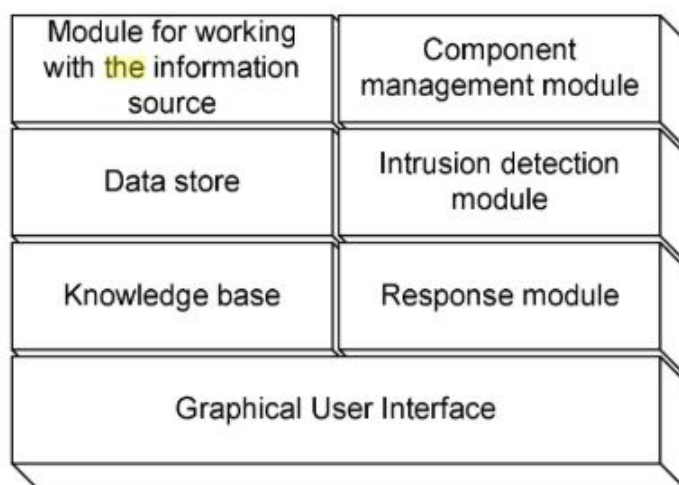


Figure 2.2: Architecture of IDS

The data is contained in a standard log file, all information about recorded attacks and suspicious incidents is contained there. This log file may have a normal text file format (such as the Snort system for example) or may be stored in the database. The database can be local (such as the MS Access database in the IDS trust system) or client server (such as Cisco IDS 4200 Oracle databases, or Real Protected Network Sensor MS SQL databases). The knowledge base includes information on the basis of which the program determines whether a particular data source can report an attack based on the information. This database can store attack signatures, user profiles, etc., depending on the analytics methods implemented.

The intrusion detection module conducts a comparison of the rules stored in the knowledge base to records from a particular data source, which can issue commands to the reaction module based on the results of the comparison. A graphical user interface makes execution of administrative tasks intuitive and convenient. Using the GUI, can collect information from all components of the intrusion detection system and perform management functions. The graphical user interface is lacking in some intrusion detection systems (especially those implemented for Unix) (for example, in the Snort system). If the

device for detecting intrusion is installed as a stand-alone agent, then all the modules described above reside on the same computer. If the system has been designed with the client-server architecture in mind, the sensor, also known as the agent or tracking module (engine), and the console have two basic levels [9]. The sensor detects and reacts to attacks and then transfers the data on the detected unauthorized activity to the management console shown in Figure. 2.3.

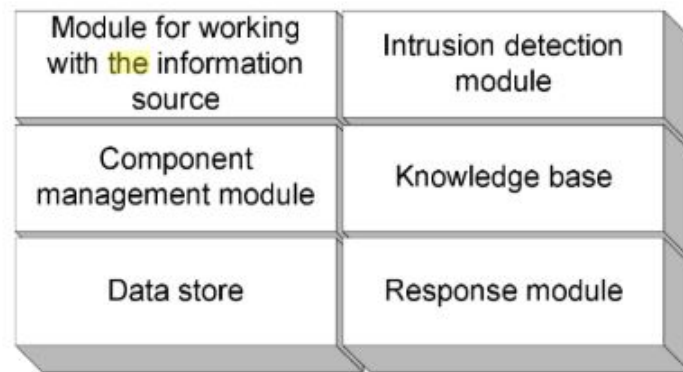


Figure 2.3: Architecture of IDS with sensor

An unlimited number of sensors can be coordinated through a single console. Likewise, any sensor will send information to multiple consoles at the same time, rendering the console fail proof [9]. This theory constructs many intrusion detection devices, such as Real Secure Network Sensor or Cisco IDS 4200.

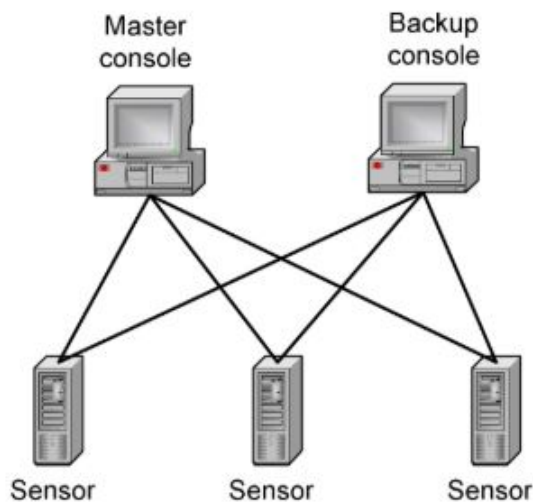


Figure 2.4: Console fault tolerance implementation

To prevent two consoles from simultaneously changing the remote sensor settings, one of these consoles must have a special status. Only the console assigned to this role is permitted to alter a remote sensor configuration and perform other management operations. This method is close to the concepts introduced in the Firewall-1 checkpoint scheme. Only one administrator at a time can link the firewall with the read / write capabilities according to these principles. Many managers only deal with Read Functions. [9]

To avoid two consoles changing the remote sensor settings simultaneously, one of these consoles needs to have a special status. Only the console assigned to this function is allowed to alter the configuration of a remote sensor and to perform other management operations. This method is close to the concepts introduced in the control point scheme of Firewall-1. In accordance with these principles, only one administrator at a time can link the firewall with the read / write capabilities. Most executives deal only with Read Functions. This scheme is particularly useful in the information security departments of companies with hierarchical structures. Such companies typically have a centralized department by design, which establishes a single information security strategy and oversees all security departments located in remote affiliates [9].

Figure.2.4 displays the configuration of the Cisco-developed hierarchical intrusion detection system.

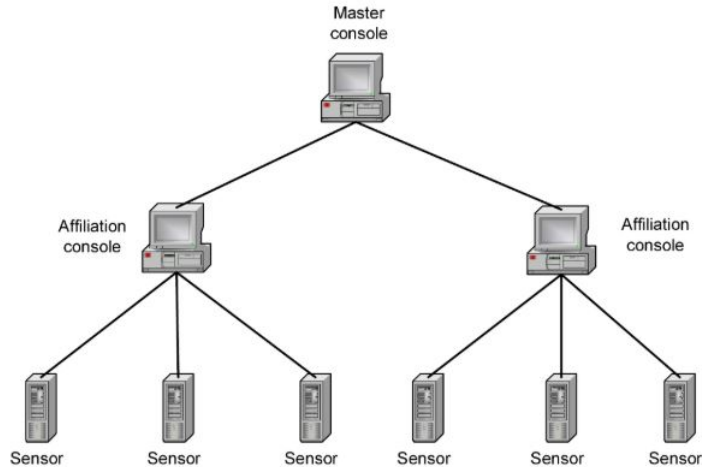


Figure 2.5: Hierarchy management of IDS

When the sensor senses an attack it sends information directly to the affiliate management console about this attack. The scheme suggested by Cisco does not differ significantly in this respect from the classical two-level scheme. However, if the sensor detects an attack that it considers especially dangerous (or some other attack identified by the administrator), it sends an warning to the affiliate console, as well as to the centralized console located at the information security department headquarters.

In addition to the above mentioned architectures, there is yet another scheme, also known as the architecture of the "sensor / control server / administrator console" shown in Figure. 2.6. Right now. In this case, sensors send information on the detected attacks to the controller server instead of to the admin console. The advantage of this scheme is that all data on the sensor loaded security policies, as well as events recorded by the sensors and other information, are stored on the controller server rather than on the administrative console [10]. The admin console function can be assigned to any device. At the same time , the server is usually a powerful, fault-resistant system, on

which the likelihood of failure in any other server or workstation is significantly lower. Under this scheme the NetProwler intrusion detection system is built [9].

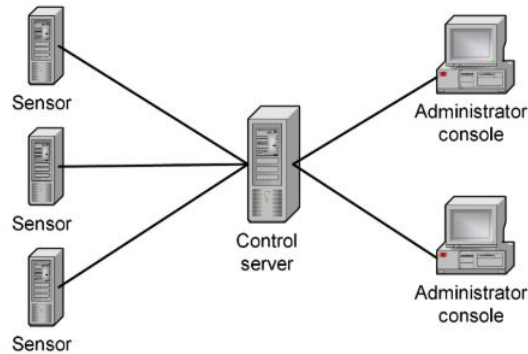


Figure 2.6: Three level sensor management scheme

2.4 Detection Approaches

The theory of detecting unauthorized action is depend on the idea, that is disruptive behaviors are significantly different than usual ones and, thus, are observable. Those methods are divided historically into three groups [6]:

1. Misuse detection
2. Anomaly detection
3. Specification-based detection

Approaches in anomaly detection are committed to developing a flow of data structure that is tracked simple conditions with-out any disruptive procedures being present. Misuse detection approaches, on the other hand, the primary goal to publish those pattern knowledge in to encoded way to get the specific signatures. Security experts have predefined the permitted system behaviors in specification-based detection approaches and are therefore labeled as attacks which is events that do not match with the specifications.

2.4.1 Misuse Detection

Intrusions with known suspicious patterns can be identified by matching real activity reported in audit trails. Although detection of misuse is completely competent to uncover established attacks, it is useless in the face of unknown or new types of attacks for which the signatures are not yet available.

Defining signatures for the attacks which are known is very difficult. This will lead to the increase the false positives and reduce the rate of detection. It is effect on the effective rate of the detection.

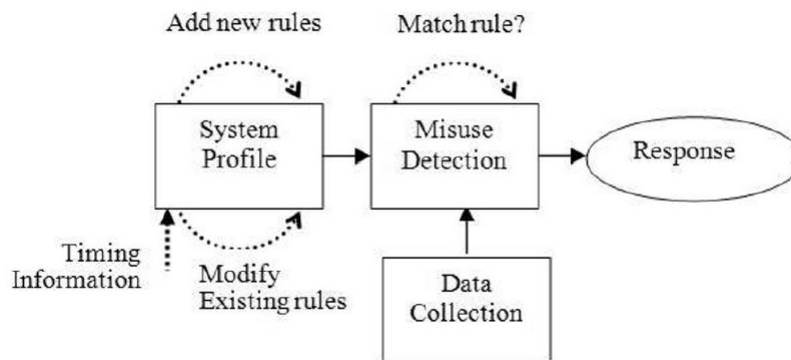


Figure 2.7: Structure of Misuse Detection

Misuse detection is consist with the four models. Feeds are obtained from several data sources, including all the log entries of the resources. The gathered feed is translated into a another format that the other components of the device understand. The device profile is used to describe both natural and anomalous behaviour. The profiles describe what would be the subject 's daily behavior, and what regular operations the subjects conduct on the objects. In the event of anomalies, the profiles are matched to actual device operations and recorded as intrusions [7]. following methods are commonly used in misuse detection

1. Pattern matching

2. Rule-based techniques
3. State-based techniques
4. Data mining

2.4.2 Pattern Matching

Sample compatibility penetration detection, network-based intrusion detection systems methods are commonly used in that model. Attack forms can be created in HIDS by combining terms representing device calls in the audit path. With the various type of attacks, number of attacks signatures are increased. That effect on the computational cost.

2.4.3 Rule-based Techniques

This methods of preventing abuse. Expert systems encrypt intrusive displays into a set of rules that are related to network traffic feeds.

2.4.4 State-based Techniques

State-based techniques use system status expressions and state changes to detect known intrusions. Activities that contribute to intrusion scenarios are defined in state-based techniques defined in the figure 2.7.

The system state is user or process function. The state transition diagram defines intrusion scenarios that include three types of states, namely initial state, transition state and compromised state. In the first initial state corresponds start of the attacking the compromised system, while the compromising, it reflects the attack completion. Transition state is the status where the successive states that exist between an initial state and a compromised state.

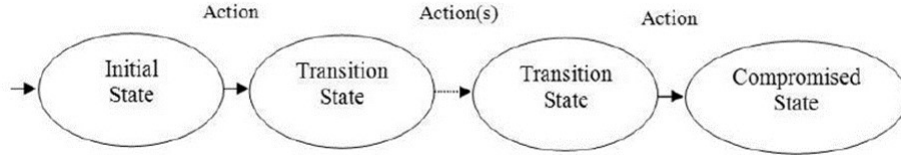


Figure 2.8: State transition diagrams

2.4.5 Anomaly Detection

Unlike detection of misuse, detection of abnormalities is dedicated to creating regular profiles of the system's operation. It assumes that invasive behaviors are actually anomalous.

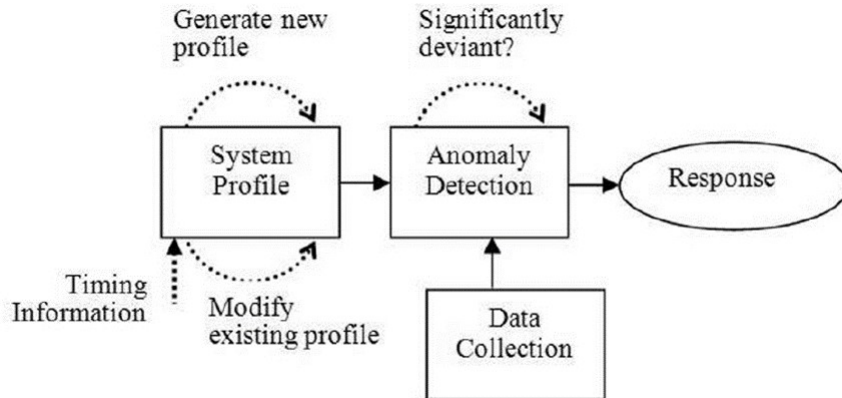


Figure 2.9: Anomaly detection model

A standard model for the identification of anomalies is shown on Figure 2.9. It include four components, namely data gathering, normal system profile, anomalies detection and response. The data collected data are normal user activities or traffic data, and it need to saves them. To build standard device profiles, different modelling techniques are used. The portion of anomaly detec-

tion determines how much the current activities change from the usual profiles and percentage amount of those activities should be identified as anomalous. At final those responses are reported as intrusion.

The advantage of the anomaly detection is ability to identify new attacks; as such it solves the greatest weakness in the detection of misuse.

Anomaly detection false alarm rate is in very high level. The key explanations for this restriction explicitly include the following:

1. The typical behavior model for users is based on data collected during daily operations;.
2. Stealthy attacks are very hard to detect by using anomaly detection techniques. Additionally , security experts typically agree on the types of parameters used as inputs in standard models. Any error that occurs during the process of determining those inputs will be a reason for the high rate of false positives.It reduce the anomaly detection effectiveness.

2.4.6 Use of Honeypots in Intruder Detection Systems

Additionally, threats of internet attacks came alongside it, with the advantage of communicating through the internet. Specific technologies are commonly employed to advance network security. To discover the society of black hats, you need to stay up to date with the innovations of the hackers. In recent times, specific variants of security scenario activities, black-hats and white hats have been determined.

Black hats exploit the network, while white hats protect the system. Honeypots have been used for attacks in combat. Honeypots will be outlined as an attractive defense tool placed in an excessive network that attracts, detects, and observes attackers with the specific intention of understanding them. Honeypots will be used for different purposes, such as prevention , identification and collection of information about network threats.

To review in social networks about hackers and the way they transmit to each other. It is necessary to provide the offender with true software so that

the offender gains root system privileges, and knowledge about the attack will be known. The amount of activity the offender performs is called level of interaction. Honeypots are divided into specific categories, namely low-interaction Honeypots and high-interaction Honeypots.

Low-interaction honeypots give the attackers the minimum interaction, and therefore the system and capture only a small amount of knowledge about the attacks. It will simulate various operating systems and provide them with diverse TCP / IP services. A large topology network that can be simulated with completely different routers to work with numerous topology types.

High-interaction honeypots also communicate with attackers. In addition, helps the intruder to play with the critical operating system. High-interaction honeypots do not predict the associated degree attackers can attack and prepare the services to react accordingly. These honeypots explore the attacker with the main operating system and applications.

Honeypots are often categorized into 3 entirely different categories according to the capabilities, namely preventive honeypots, deceptive honeypots and detective honeypots. Preventive honeypots are used for network protection and are often divided into sub-categories such as sticky honeypots and deceptive honeypots.

Sticky honeypots are the low-interaction honeypots that protect the network from attacks powered by machines such as worms. These attacks scans the networks for vulnerable systems and if found, the system is overtaken and slows down the attacker by TCP tricks. On the contrary, deceptive honeypots are the honeypots that may have low honeypot interaction or high honeypot interaction that protects against human attacks. The main aim of these honeypots is to waste the time of the attacker and by the time attacker communicates with the computer all relevant data about the attacker are collected much like the devices, techniques used by the attacker, but they take over the network.

2.5 Intrusion Prevention Systems (IPS)

The IPS, the tool used to detect and blocking on malicious activities in real time. In general, Two categories are placed:

1. rate-based.
2. content-based.

The similar features in firewalls. But firewalls block all traffic except for reason, while IPS is blocking the traffic except that for which they have reason to block it.

2.5.1 Rate-based IPS

The most effective IPS-based rate requires a combination of powerful configuration choices with a variety of answers. Network load based traffic, high rate of packets or misconfigurations lead to the rate based IPS to block thhe content. In rate based IPSs were measure the rate of incident [5].

These kind of system may set a maximum network traffic to the given port. If that port is reached to the threshold, the IPS will take action as block all further source IP traffic only. (source IPs).

2.5.2 Disadvantages of Rate-based IPS

Depending on the rate of deployment, the biggest challenge with IPS goods is determining what constitutes a surcharge. There should be a way to handle normal traffic and owner should know the status of his network in order to operate properly with any IPS dependent rate. Because those kind of IPS requires frequent consistant database tuning of the rate base IPSs. [5].

2.5.3 Content-based Products

These kind of IPSs take action based on the signature and the protocol anomalies of the traffic captured. They are blocking:

1. Worms e (e.g. Blaster and MyDoom) that match a signature can be blocked.
2. Packets that do not comply with TCP/IP RFCs can be dropped.
3. Suspicious behaviour.

The best content-based IPS provides a variety of malicious content detection techniques and several options for managing attacks, such as simply dropping bad packets to drop potential packets from the same attacker, and advanced monitoring and warning strategies. These type of IPSs are looking at the malicious detection on network and enable action for the managing those attacks. It is dropping the packets of malicious activity. Alerts and consistent monitoring were established on those activities. This act with the firewall rules [5].

2.6 Threat Intelligence

Threat intelligence is a very broad area and can mean different things to different people as mentioned above. Multiple structured and unstructured data sources which may include, but are not limited to, IP addresses, geo IP location, pattern matching, malicious binaries, malicious compound documents such as: PDF files, Microsoft word documents, etc. When the Internet continues to shift towards using Transport Layer Security (TLS) and Secured Socket Layer (SSL), relying on IP credibility would be an important correlating point for traffic, but it will also involve the ability to launch artificial virtual smart machines that act as a legitimate user to verify and validate encrypted traffic. This is critical because gathering security intelligence by breaking TLS / SSL and/or VPN encrypted traffic infringes privacy, not to mention compliance with regulations [4].

Cyber security and forensic practitioners need a increasing number of cyber-attacks to identify , analyze and secure real-time cyber threats [4]. In reality, it is difficult to handle such a large number of attacks in a timely manner without reading extensively the characteristics of the attack and adopting sufficient sensible steps of defence. Artificial intelligence, machine learning and data processing technique are the scalable threat intelligence factors.

The most difficult challenge in the digital systems and devices is to ensure that knowledge about individuals and organizations is about security and privacy. During the past few years , cyber attacks were increased rapidly. Since that development of the threat intelligence take prominent factor in cyber security aspect. Data collection and analyzing is vital part of the threat intelligence. There is a requirement of implementing the proper analytic system [12].

2.6.1 Current Threat Intelligence Definition

Cyber Threat Intelligence (CTI), it can be explained by using context, process, indicator, consequences and actionable advice, about an current or emerging danger as evidence based information. This can be decide by the

organization by considering status of the tactical level with the threat level [1].

In [1], Several Cyber Threat Intelligence concepts that are organizational, analytical and domain dependent. Consequently, Cyber Threat Intelligence Operations were defined as acts taken in cyberspace to compromise and protect the information on the threat space. The intelligence analyzes as the study of such defensive actions and the analysis method to protect the resources available.

2.6.2 Types of Threat Intelligence

Even though many claim they compromise security, that can mean a whole host of things. There are four types of threat intelligence [14].

1. The high level analysis of threat intelligence that is used of the decision making process. It is called Strategic Threat intelligence.
2. Collecting observable form the ongoing and the incoming attacks which include the all details of the attackers and traffic data that is help to built threat intelligence. That is called Operational level threat intelligence.
3. Tactical intelligence focused on the on the tactics, techniques, and procedures of threat.
4. Malware research and detection done at the technical threat intelligence. It is heavily based on the Indicators of Compromise. It is catalog malware families to identify the characteristics such as textual or binary patterns.

2.6.3 Threat Intelligence Platform Capabilities

Threat Information systems cover many functional areas that upgrade and represent a security strategy powered by information. Ideally, a fully developed framework that functions help through to automated and streamline the workflows, and it monitor the throughout the session complete. Threat Intelligence Platforms will deliver the following basic items [14]:

1. Collection: Collects and aggregates various sources and data types, including CSV (Comma Separated Value) files, STIX (Structured Risk Information Expression), Custom XML / JSON, CVE (Common Vulnerabilities and Exposures), OpenIOC (Compromise Indicators), system logs, and email. Although SIEM systems can handle multiple feeds related to threat-intelligence, they are improperly instrumented to take in and interpret the unstructured formats of free-form, text-heavy unstructured data that frequently characterize feeds of intelligence. It's not unusual to raise a threat intelligence network with a SIEM, but attempting to substitute it with one isn't successful.
2. Correlation: Facilitates automatic data analysis and correlation so that an attack can be mapped out, linkage can take place, and countermeasures can be implemented. Human health checks and imagination will prevail and be involved in the evaluation of correlated data, but all that is needed is automation and machine learning to fulfill this role at a basic level.
3. Context: Provides enrichment and circumstantial data on incidents in question, in which they remain subjective and without documenting trends and linkages. The data is organized in information. Information organized by the Intelligence, with context. A threat intelligence network should be able to take in additional information obtained from other incidents and investigations in order to make decisions and take action correctly and intelligently.
4. Analysis: Evaluates and draws conclusions concerning indicators of the threat. Research recognizes the intricacies of event relationships to provide concrete knowledge of the hazard from the otherwise irrelevant data in pivoting sets.
5. Integration: Supports organizational workflow and funnels harmful intelligence data for intervention and life cycle maintenance in security software and goods. Platforms can collect and redistribute structured, usable data to other operational instruments, such as SIEM systems and perimeter security technologies, as well as certain processes, like incident management, monitoring and/or ticketing systems.

6. Action: Accelerates and handles the preparation and evaluation of subsequent action and reaction, be it within an entity or outside. Cooperation between the security department and other activities and business lines within an organization, as well as between an organization and other third parties or stakeholders, such as regulatory agencies, law enforcement or (ISACs), with which the organization communicates or reacts.

2.6.4 Cyber Threat Intelligence Challenges

Cyber criminals, in addition, follow many forms of attacking a victim. Those are:

1. unauthorized access to sensitive personal information or
2. perform malicious activities,

These kind of interruption make client machine n to the cyber attack by distributing the malware. The distribution development is initiated on the victims machine or network [3].

2.6.5 Attack Vector Reconnaissance

The identification of the point network weaknesses that could used as abused by cyber criminals. That is an significant factor in protecting against cyber attacks. In addition, by tricking the machine of the victims are carrying the facts used by the victims. Such strategies range from delivering malware in an unusual medium to the compromised resource, leveraging 0-day vulnerabilities and infringing anonymous correspondence to contact threatening actors [3].

2.6.6 Attack Indicator Reconnaissance

Another important factor is cyber criminals used advanced forensic tools for their enhance their capabilities. They perform consistent watch on the vulnerable systems and perform major vulnerability assessments on that particular systems [3].

2.6.7 Cyber Threat Intelligence Opportunities

In the present scenario, the cyber threat intelligence focuses on the major improvement using artificial intelligence and machine learning technologies [3]. There is a growing trend on the anomaly detection that the detection through the ML and artificial intelligence. The efficiency of those systems is proven by the use of those technologies. To mislead the attackers, defenders use honeypots. In that scenario, fake information will show up to the attacker. While his attempt on dealing with the honeypot continues monitoring can be performed. That help to detect the attacker proactively.

2.7 Threat Sharing Platform

Cyber Threat Intelligence has gained substantial media attention in recent years [1], and has been described as a way to combat the growing number of security incidents and the nature of those. Many organisations use open-source threat intelligence or commercial threat intelligence sources to subscribe to the threat feeds on cyber space. Main drawback is the too much data is consume for that process. The problem. This may lead to the overloading information to the threat library. That is the reason behind the implantation of threat sharing platform. That can integrate with the tools available for defending.

Presently, most of the security vendors are provide the threat information over the cyber space. Those TISP solution wiil provide the data feeds to the developed applications. That will enable immediate response on those attacks In [2]. Sloution providers are namely as FS-ISAC, OASIS, IBM X-Force Exchange, Facebook Xchange, HP Threat Central, Checkpoint IntelliStore, Alienvault OTX, and Crowdstrike intelligence exchange more focus on content aggre- gation. While Intelworks, Soltra, Threat stream, ThreatConnect, Vorstack, Threat Quotient and CRITs, are the few TISP avilable in the market.

Sharing platform are powerful instruments that assist defender to access and exchange technological, tactical, organizational and strategic IT more effectively. The Sharing platform also allow them to more automate TI processing. TISPs greatly improve the capacity of an organization to identify risks to

information security .

The integration of sharing platform into the IT infrastructure of an organization as tensively as possible is required. The sharing mechanism support filtering cybersecurity threats, but in organizations that promote sufficient reactions, they need to be complemented by systems. The awareness of the security employees will get the knowledge on their unexpected IT incidents, so that they can respond immediately. Yet they still have to respond manually in most cases.

2.8 Enabling Automated Responses from Policy

This section will go through the policy development for the cyber threat. Through the section, threat sharing platform development and use cases for threat sharing platform will discussed.

2.8.1 Security Policies

TISP priorities and benefits of critical IT infrastructures management. With the use of technological aspect, hardware and software make up these infrastructures. The above enforces application-specific safety targets (e.g., consumer information security, accounting honesty in development, etc.). These systems are increasingly relying on a security policy to protect the security essential resources. Rules generation will implemented on automatically and that will limit access to the resources. This called as the Security Policy-Controlled System (SPCS). Although the rules vary semantically between these classes, they are both aimed at ensuring SPCS security properties.

This motivates the need to apply these policies more efficiently and flexibly during system service. Next step is then to analyze how existing SPCSs are working to achieve such versatility. This leads to an orthogonal policy class to the above-mentioned classification: context aware security policies. The subclass of risk-based security policies deals specifically with the impact of dynamic threats within that class. The operational issue of lacking in-formation,

however, remains. They propose leveraging TIS technology , especially TISPs, to tackle this open problem. This could help to minimize the operational risks that SPCSs face.

2.8.2 Engineering and Enforcing Security Policies

The underlying operating systems need to rigorously enforce specific security standards for the operation of security-critical IT infrastructures. It includes effective network management and an effective system architecture that works on these safety needs. A modern solution for this is focused on encapsulating techniques that are used in security policies to define necessary device properties in terms of access control (AC), authentication, communication protection, etc. The effect is a complex method of software engineering known as model based protection engineering. The process go with the three steps:

1. Policy Engineering refers to the design of a security program Requirements on protection. The consequence is an series of laws defining How security mechanisms in software enforce those requirements.
2. Model Analysis involves systematically evaluating the security strategy to suit safety criteria.
3. Model implementation for the protection.

This methods allow for precise description of security policies in safety models. This method helps formulate the analyzing the security properties and gives strong guarantees of policy implementations being correct. One explanation for this is that computational complexity still plagues model analysis. This will make it difficult o handle the SPCS effectively and it may increase the risk. The existing standards is not help to address the operational risk.

Without adaptive security policies, the dynamics of threats in modern IT communicating systems are increasingly demanding an immediate response. The improvement of the modern technology is focused on the adaptive security policies for the increase the threat detection and the communication method.

Human involvement make this process more complicated .

The main problem faced is the changing requirement of the computer technology. Security policies must update with the new technology concept. adaptive, scalable solution can address the those issues immediately. further it was reconfiguration happen with out developing a new program. Context aware policies is the one of the scalable method definition for the adaptability.

2.9 Threat Hunting

Threat hunting is a feature of well matured soc. Threat hunting is basically discovering the threat information. Vulnerability information of particular system were observed by the threat intelligence platform to proceed to the threat hunting.

2.9.1 Intel Sources

To Maintain a proper threat intelligence, intelligence data come from the various sources. Accuracy of that data sources are mainly affect on the threat intelligence feeds. Data need to extract from the data sources are useful for the intelligence. That will help to make actionable data for the threat platform.

- Internal threats
 - Incident History
 - Data breach attempt history against infrastructure
 - threat on organization
 - Threats to intelligence
- External
 - Commercial intelligence feeds
 - Free Intelligence (OSINT)

2.9.2 Importance of Detection

Defending on the threat is quite important factor when considering threat information. The intruders always gather information on particular organization to update their vulnerability information. Therefore dedicated team need for the improve the detection of those vulnerable point of organization. Prevention is not playing vital role since that is the secondary factor of protecting the resources. Updated knowledge of threat information will increase the detection capability of particular organization. Models for threat hunting are as follows [13] :

- The Lockheed Martin Cyber Kill Chain
- The Mandiant Attack Lifecycle
- The MITRE ATTACK Framework

2.10 Summary

This chapter described the IDS and IPS models, deployment methods, security challenges of IDS and fundamental security challenges faced by IDS and IPS. Further it described the IDS reference model also. This chapter focused on the threat intelligence, concept and origin and how it relate to the IPS/IDS at the level of the defending threats.

Chapter 3

Methodology for Cyber TI Support to Incident Handling

3.1 Introduction

This section presents the methodology used for providing recommendations on the CTI sharing aspects of the Cyber-Trust project by relying on a set of high-level requirements, as described in the description of action (DoA), and by considering the findings of research on the current situation on CTI sharing and automation. This has taken into account:

1. The availability of standards. CTI formats and languages for reporting vulnerabilities, threats and other information gathered from various CTI sources.
2. Their support from open source tool. CTI platforms will developed based on supported open source tools

A distinction should be made between a automating sharing mechanism and a platform. While the former structures the encoding of information (e.g., by providing rules for XML tags to allow for automatic processing and possibly decision-making), the latter provides a tool allowing to efficiently share information. A number of requirements stemming from the above considerations are presented in the following sections.

1. Requirement 1: Sharing mechanism of the Cyber-Trust Platform

2. Requirement 2: Expressibility, Flexibility, & Scalability of TI
3. Requirement 3. Information used to facilitate automation.
4. Requirement 4. Streamlining of hunting and incident response process.

3.2 Requirement 1: Sharing Mechanism of the Cyber-Trust Platform

When considering Threat Intelligence, classification schema like Strategic Intelligence and Tactical Intelligence is a very useful. Several of the intelligence-classification requirements include:

1. Gathering Techniques Threat Intelligence get information over the Internet from a honeynet to tightly secure secrets intercepted by agent.
2. Cost the subscription model is matter with the cot of the sources.
3. Main Usage intelligence as the primary objective
4. Target Audience- place or a system which benefited from the intelligence
5. Specificity threat
6. Lifespan life time of the threat indicators

Classification of sharing is based upon the whim of the analyst. The information also extract the observables. To get the full benefit from the threat intelligence, Structured, machine-readable communication format is needed.

For the facilitate the sharing of intelligence some languages were introduced:

1. Open Compromise Indicators (openIOC)- It was Developed to assist and track the competitors.
2. Structured Threat Knowledge Communication (STIX) developed a language top provide "entirely descriptive, versatile, extensible, automated and as human-readable as possible" (MITRE STIX).

3. Cyber Observable (CybOX)- The scheme created by MITRE to describe observable.
4. Trusted Automated Indicator Exchange Information (TAXII)-is a threat Intelligence sharing protocol to provide information to describe the intelligence.
5. In 2007 Focuses on the reporting of cyber incidents. A few tools appear to implement this standard, including Foundstone and DFLabs.

3.3 Requirement 2: Expressibility, Flexibility, & Scalability of TI

The Threat Intelligence Platform is a for solving the area of collection , storage, and sharing of problems. To enable threat intelligence platform in secure, scalable and easy access threat feed from the OS-INT is required. This help to integrate defensive resources for the threat sharing functionalities. It depend on the capabilities used in the context. That will help to generate threat intelligence for accurate observables.

Following open source tools were provide more organized and promising threat alerts to the contexts.

1. The Collective Intelligence Framework (CIF - <https://code.google.com/p/collective-intelligence-framework/>). REN-ISAC, the educational sharing and analysis of content, established this system. It was created to assist in the ingestion of network component to find out the observable.
2. Collaborative Research into Threats (CRITs - http://crits.github.io/threat_sharing.html). CRITs is mainly act as the dynamic threat analyzer. This information used to update particular threat libray using REST API.
3. Mantis (<http://django-mantis.readthedocs.org/en/latest/>). This is the malware analysis tool, that is capable of importing and storing most existing laguages (IODEF, openIOC, STIX).

4. Malware Information Sharing Platform (MISP <http://www.misp-project.org/>). It was developed by MITRE to analyze the rare malware on the threat space.

3.4 Requirement 3. Information Used to Facilitate Automation

Main focus is to get more advantages from the attacker by interrupting their capabilities using threat intelligence feeds. From the offensive side, the impact of the illustrated pyramid of pain on figure 3.1

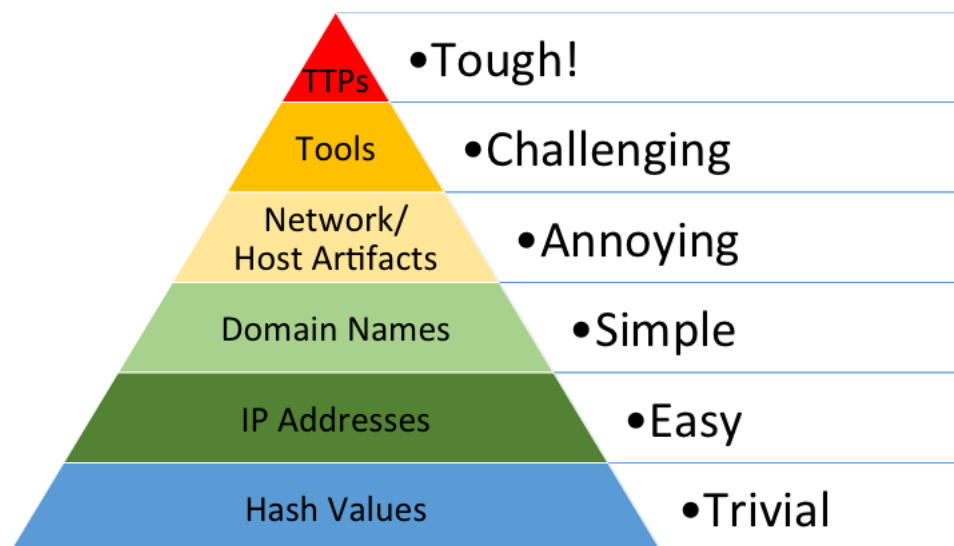


Figure 3.1: David Bianco - Pyramid of Pain

When considering resource utilization of threat intelligence, life cycle of observable is playing vital role. Intruders can use their offline capabilities to compromise the network. It making obsolete lists very quickly. Categorization is mainly based on the trust and verified data. That can used by the consumer to implement the threat information.

Following cases for Threat Intelligence have been given to facilitate

automation :

1. Establish proper plan for Security
2. Proper detection
 - (a) Proper Preventive action
3. Immediate Response on Incident
 - (a) Triggering Alerts
 - (b) Threat Identification
4. Create Threat library:
 - (a) Intelligence of threat discovery
 - (b) Analysis of threat assessment

Both forms of intelligence benefit from ingesting the feeds of information, as well as a number of success metrics in using information in defensive activities. The specific use cases used for the indicators are shown on the table 3.1.

3.5 Requirement 4. Streamlining of Hunting and Incident Response Process

In order to simplify and streamline the hunting and incident response processes so time spend to important task can increase rather than spending on the other operation related to the threat intelligence. Another benefit regarding to the automation is the process of establishment of process continuity and the encouragement.

For achieving the defensive control over the configuration done at the point of integration of threat library and SIEM done at the initial configuration. The first start of the implementation is based on the alert generated by the SIEM rule set. That depend on the those indicators.

The initial observables (IP, FQDN, Hashes) provided by the open source paltform are used to store on threat library. The following reference sets will serve as a basis for various warning acts:

Table 3.1: Threat Intelligence use cases

Use Case	Specificity	Strategic / Tactical	Product	Key Performance Indicators
Security Planning	Low	Strategic	Security Vision, Response Plans, Security Roadmaps	Success in response to a targeted attack
Threat Intelligence Collection and Fusion	Low	Both Threat Intelligence	Reports and Indicators	
Incident Response	Medium	Both	Incident Response	Time to containment, correct identification and scoping of incidents
Enterprise Security	Monitoring	High	Tactical Blocks, Alerts, Context	Time to detection, time to escalation, false-positive rate for alerts

- Emails of the Attacker
- Domains (High and Low)
- Hashes (High and Low)
- Ips (High and Low)
- Targets

For enhancement of the current alert medium set is used. It is based on the signal interpretation. The High set is trigger an alarm automatically. For the advanced attack , email warnings generated and it reported over the attacker.

The source for the threat intelligence is the OS-INT which is combine by default. By default combine collect artifact and by using API it upload to the CRITs. CRITs will populate the indicators with the level of trust. Those observables read from the NYX tool and CRITs and take preventive action. Then it will feed to the system and alert will gernerete through the SIEM.

For response phase,the incident, using nyx tool to assess the degree of trust. By looking at the relevant program, Nyx attempts to assess the confidence associated with the measurable. It will allow some confidence-enhancing methods:

- Status if the indicator as to linked to various sources of intelligence, then the quality measure will be increased on the basis that several third parties have found it to be.
- The confidence of the variable Will be at least extremely effective as the overall level of confidence in the campaign.

NYX, the python library will focused on the five categories of CRIT's Threat Intelligence. They are namely : IP addresses, domains, samples, emails and targets. The particular threat intelligence library will load the developed Bro script.Observable as hashes to analyze malware to take action against detectable.

3.6 Summary

This chapter focused on tactical threat intelligence implementation process. The requirement for the proposed framework was discussed according to cases of tactical threat intelligence.

Chapter 4

Solution Framework for Detection and Analysis

4.1 Introduction

In this chapter focuses on a more comprehensive implementation of tactical threat intelligence, observed automatically monitored by technological controls. The suggested system would work according to cases of tactical threat intelligence being used in general. The key thing is the introduction of actionable incorporation of the Threat Intelligence network into the defense systems. This should keep eye on the observed findings and actionable process help to analyzer to take action against the threat. Quality of the proposed framework will maintain by the analytical process.

4.2 Tactics Used for Threat Intelligence

Most of the time, open source devices not provide the better support in behaviour, but it need time for the engineer to spend on the particular device to setting up the IPS site. Regular tuning and the adjustment is needed since that very useful in the cloud based application and the mail server environment.

Focusing on the threat and confidence, the defender focuses on building a matrix to aid with the measurable decision-making principle shown in Table

4.1. As per the confidence matrix, for the medium impact and for the Medium confidence threat will get the decision as alert. as per the confidence matrix, both of the impact and confidence will be functions of the fusion capabilities of the framework.

Table 4.1: Confidence matrix for measurable decision making

Impact/ Confidence	Unknown	Low	Medium	High
Benign	ignore	context	context	context
Low	context	context	context	context
Medium	context	alert	alert	alert
High	context	alert	block	block
Critical	context	alert	block	block

As per the analyzer view of analyzer, it is better to maintain the distribution of tactical indicator on the for the avoid the false positive as shown in table 4.2 to appropriate systems.

Table 4.2: Confidence-based actions

Action	Technologies used for Defense
Block	IPS or Next Generation Firewall
Alert	BRO-IDS , SIEM
Context	Integration of Threat Library
Ignore	/dev/null

4.3 Automation of the Threat Intelligence

When it is focused on the automation of the threat-intelligence framework, a system analyzer has different views and it is basically depend on the maturity level, experience and the capabilities. Most intrusion detection system are allow to feed vendor specific threat feeds to the system. It is make easier to absorb those information to the vendor specific Antivirus, web filter and next-generation firewall providers.

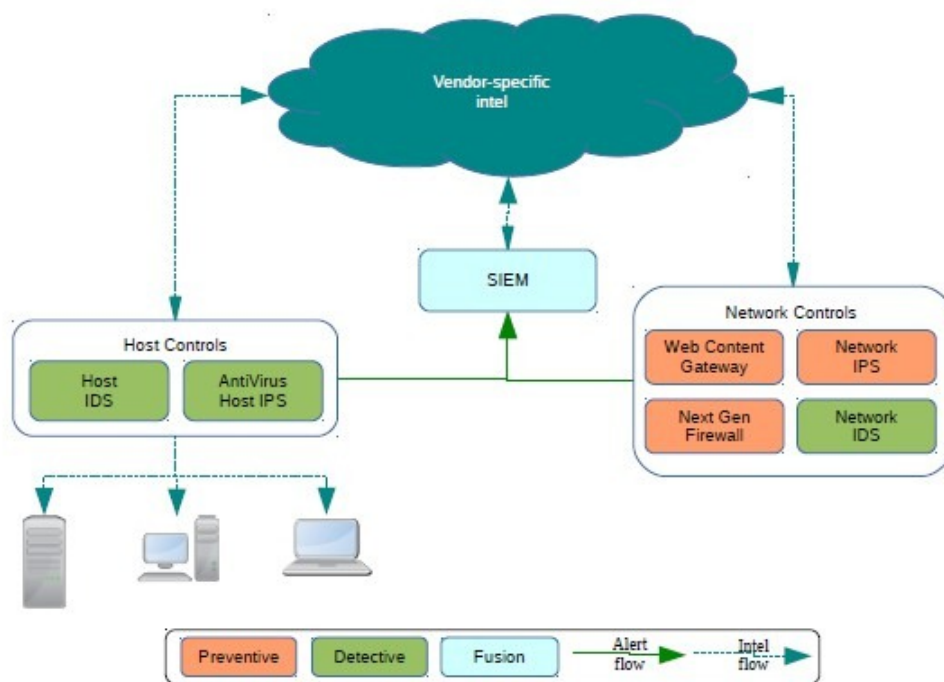


Figure 4.1: Initializing Intel Feeds on devices

The intelligence solution would provide the advantage of the different tools. That is mainly on threat Intelligence capabilities, and be easy to use. When considering threat are not in malicious infrastructure, suggested method is not provided the effective protection from the threat. It is the initial start to the project implementation. Considering those facts, the next target is to integrating defensive capabilities to the threat intelligence, initially it is to manually apply blocking or alerting mechanism to threat intelligence. Threat information will gather information from the OS-INT and from sources of commercial intelligence. Using those information provided by the OS-INT, provided system will be modified accordingly on an ad-hoc basis with signatures and blacklists.

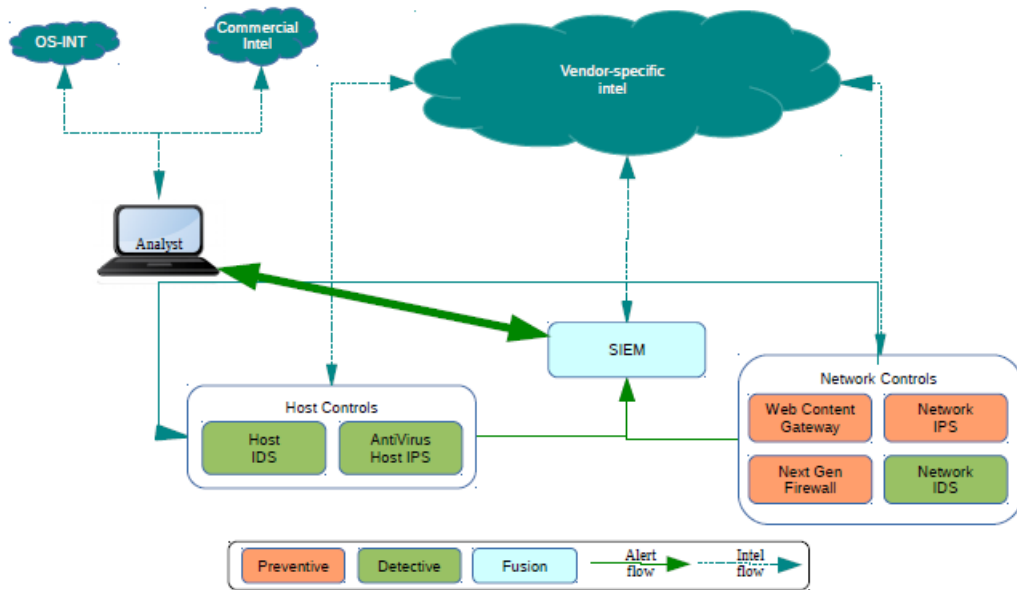


Figure 4.2: Anecdotal data by using augmented default feeds

Combined with anecdotal information, Augmenting Vendor centric is the way to deal with the threat information intelligence feature is shown in Figure 4.2.

This will provide method for the augments, while that is the method for the blocking based on the data gathered from the threat intelligence and it provide the block based on the vendor provided intelligence and this provide the documenting of those attacks and it will update the observable, this make lead to the time scaling issue with the observable found. Scaling according to the false positives are the another issue faced. Setting up the bro ids, bro-signatures to provide legitimate warning to the system provided.

Considering issue on false positives, that is opportunity for the alerts to establish a continuous improvement program. As the number of tracks observed increases, the need for a life cycle and a degree of trust is becoming more evident. Figure 4.3 demonstrates implementation of the System. One option is to fill the gap of observable Threat Intelligence and enhance the SIEM work as the alerting tool for the detected observable.

The maturity delivery model in the threat intelligence is the next step. This proposed framework provide centralized threat collection for the use of automation with the maturity deliver model. Proposed framework, data optimization is play vital role. For the optimization of threat gathered as knowledge, need to develop script to incorporate the various threat defensive technologies. There should be a proper plan to update the list automatically for proper indicators. Figure 4.3 will demonstrate the combine process of the threat intelligence.

There should be away to evaluate the observables in very structured manner with the trust framework. Screening method should adopt to the observables. Then it can rank accordingly with outputting the CSV file to better sources in any black list available for the threat intelligence. Through importing only highly secret sources, intelligence would help defenders spend precious time reviewing warnings for the test.

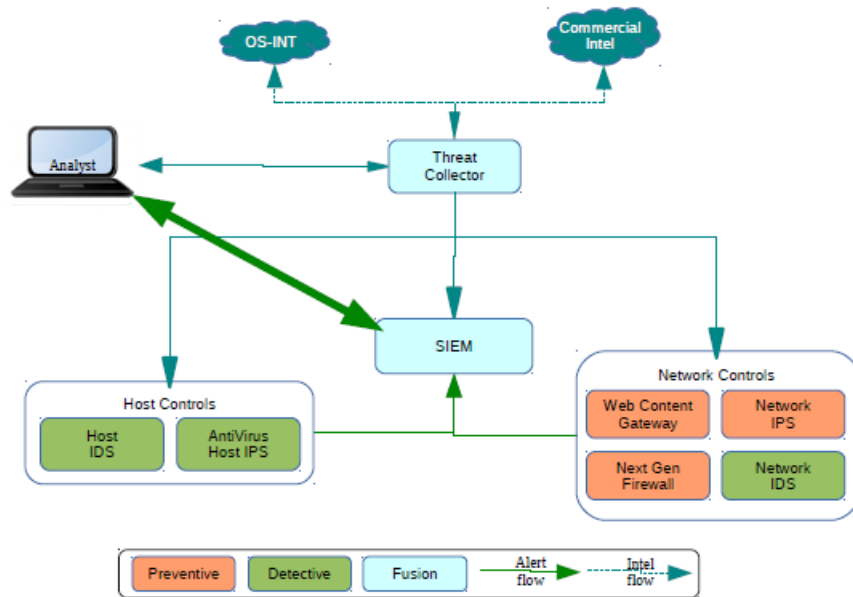


Figure 4.3: Central collection of Intelligence

4.4 Building Maturity Model

The quality warning should coming from the SIEM used. For that, threat intelligence should play on maturity level. warning should generated from the feedback mechanism by using loop as figure 4.4. This concept will enhance the quality response for incident response.

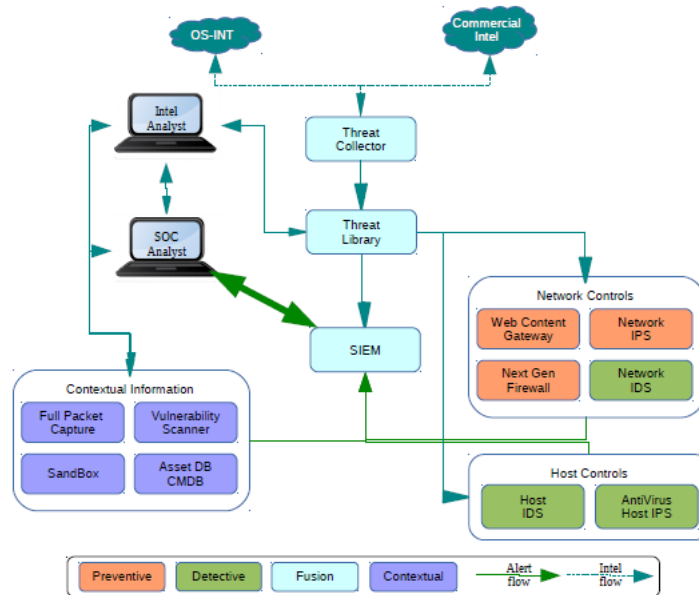


Figure 4.4: Centralized Storage and Collection

A mature model of ingestion and distribution of Threat Information will involve a range of open source and commercial information feeds that a collector would automatically investigate. Since the initial set, which records the observable as well as the contextual details, the observable will be moved to a Threat Library. The selected Hazard Library must update detective and preventive checks according to the observable list that is based on the confidence matrix discussed earlier.

For the better hunting capability, the concept proposed move to the proactive to reactive towards the alert based attitude. When it complete the packet capturing, it use logs from ids, reports of vulnerable assets and it help to context tools to complete the hunting process. The particular defender will filter the finding according to the threat intelligence gathered from the compromised hosts.

The definition would also shift away from the reactive, alert-based approach towards a more optimistic attitude towards hunting. The defenders will

be able to filter some of the findings of the search into potential intelligence or information about compromised hosts.

The integration of the proposed platform and the threat collector at this point on Figure 4.5, could support a concept by moving towards enhanced automation. From the proposed framework will not enable analytic method at once to gain automatic response towards the infrastructure. It will able it for the certain task at the initial step.

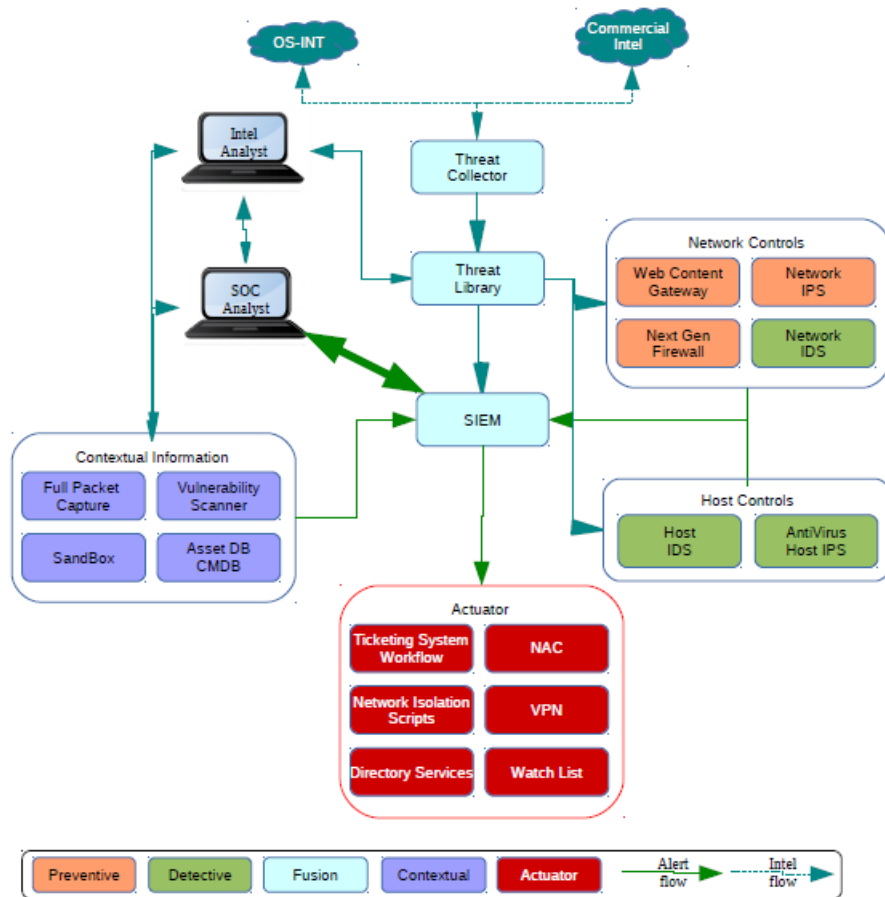


Figure 4.5: Centralize intelligence collection with automated action

When automating the incident response, the biggest issue related is providing false positive, that may lead to the network interruption. To mitigate that, system should matured accordingly with the rule set of SIEM and integrated intelligence source. As per the proposed framework describe, at the initial level the aim of self organization for the process automation processes. That will streamline the threat hunting and incident response. Threat Intelligence is the bridge for the information between the security threat checks.

4.5 Summary

This chapter focused on the issue of tactical threat intelligence implementation. The proposed framework is intended to work according to cases of tactical threat intelligence being used in general. The most critical aspect discussed is the mechanism for implementing actionable integration of the Threat Intelligence platform with defensive systems.

Chapter 5

Simulation of TI Automation and Event Sharing

5.1 Introduction

This section simulate the proposed framework scheme using open source tools. The proposed framework totally comply with the tool used for the simulation process, it is assumed that the correctness of the threat intelligence guaranteed independently according to the tools used. This chapter is to prove to enable automated defensive systems with threat Intelligence platform integration and chapter describe complex cyber threat intelligence and sharing methods.

5.2 Threat Intelligence

To enable threat intelligence feeds on the devices, OS-INT downloaded. By downloading the Open Source Intelligence (OS-INT) feeds from the commonly available on the internet in the easiest way to enable threat intelligence. Combine comes with a few input and outbound files feeds. For the proposed framework initially start with the creating observable by the use of CIF or CRITS and Combine. For the simulaion process CRITS and combine used. Nyx is used as a plug and play tool to automation.

Threat Library should become the observables reference system and it

will focus on the meta data and hashes of observable. CRITs, the analyzer useful when doing some static analysis on the file. All this information should be useful when implementing an automated distribution system to warn and block the malicious activities.

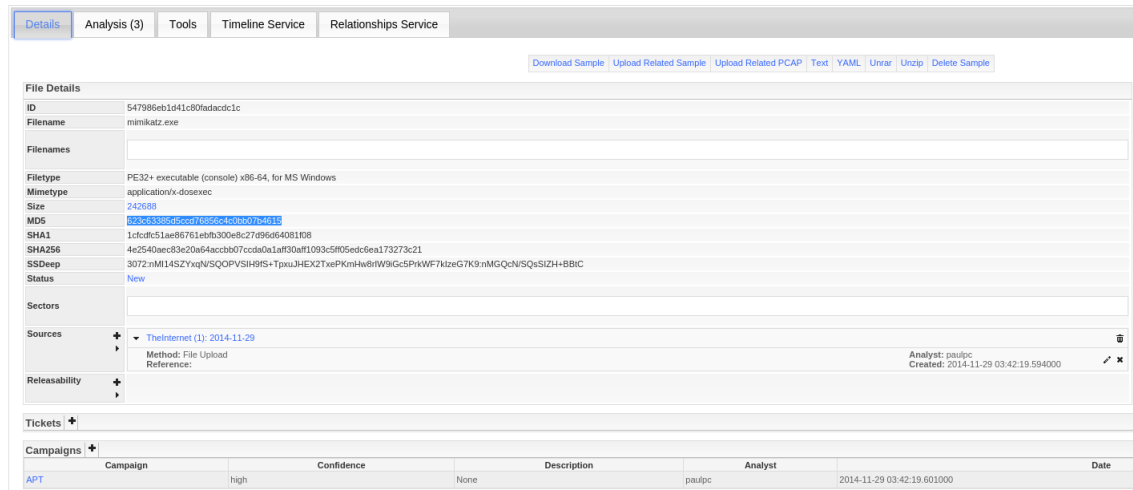


Figure 5.1: CRITs observable details

Threat intelligence libraries could provide some analytical help. CRITs have a broad range of services to promote a cursory static analysis of the observable by hitting services such as Virus Max, facilitating the analysis of strings and XOR, or similar services.

The Relationship Service provides the ability to link observed together to attempt a high-level narrative of the tactics, techniques, and procedures of the opponents, effectively moving to the controls provided by the proposed framework. This also helps in detection component by refining the metrics to obtain higher performance and enabling comparable validity of observable.

Actually, taking advantage of certain right-click context choices makes the context easier to change. For instance, it is rather trivial to search for IP addresses in Virus Total, Trusted Source, Threat Library or the ISC sites.

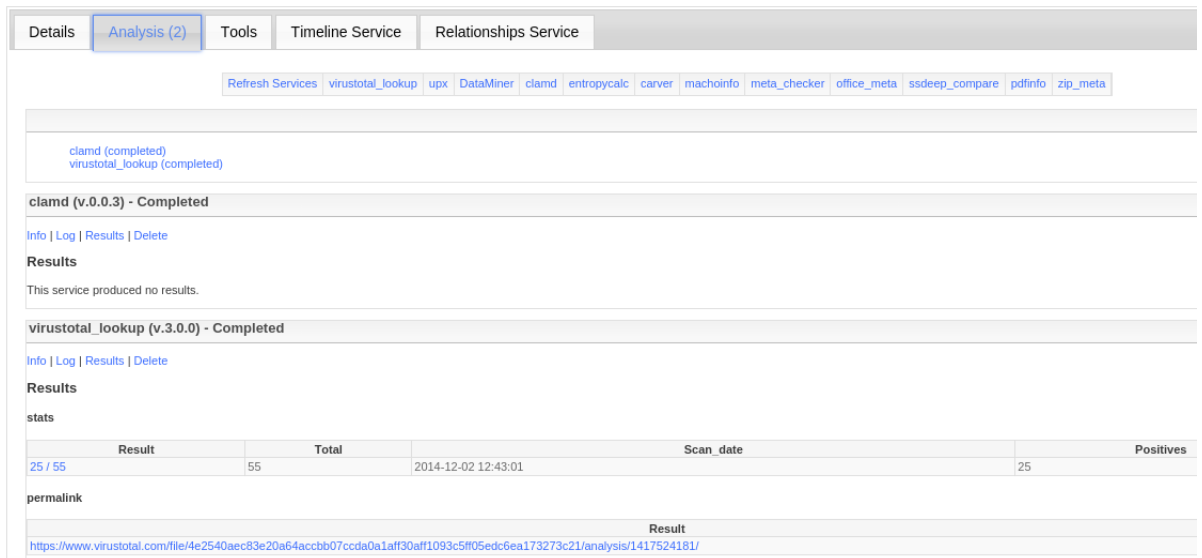


Figure 5.2: CRITs Analysis Services

To develop this, it is most practical to use Reference Set as a local repository for the observable. Most of these observable ones will have relatively low confidence, so due to the high false-positive rate they may make poor alerts, but they may be a great enhancer of alert relevance. The following reference sets should be a basis for various warning acts to this end:

- Emails of the Attacker
- Domains (High and Low)
- Hashes (High and Low)
- Ips (High and Low)
- Targets

The medium sets are used to improve the current warnings based on being part of the signal as observed. The high-sets automatically trigger an alarm. The email alerts focus on both the targets of advanced attacks as well as some of the infrastructure known to the attackers.

SIEM's effectiveness depends on the quality of the data which is provide warning for the traffic observed. For example, a set of Bro sensors monitoring

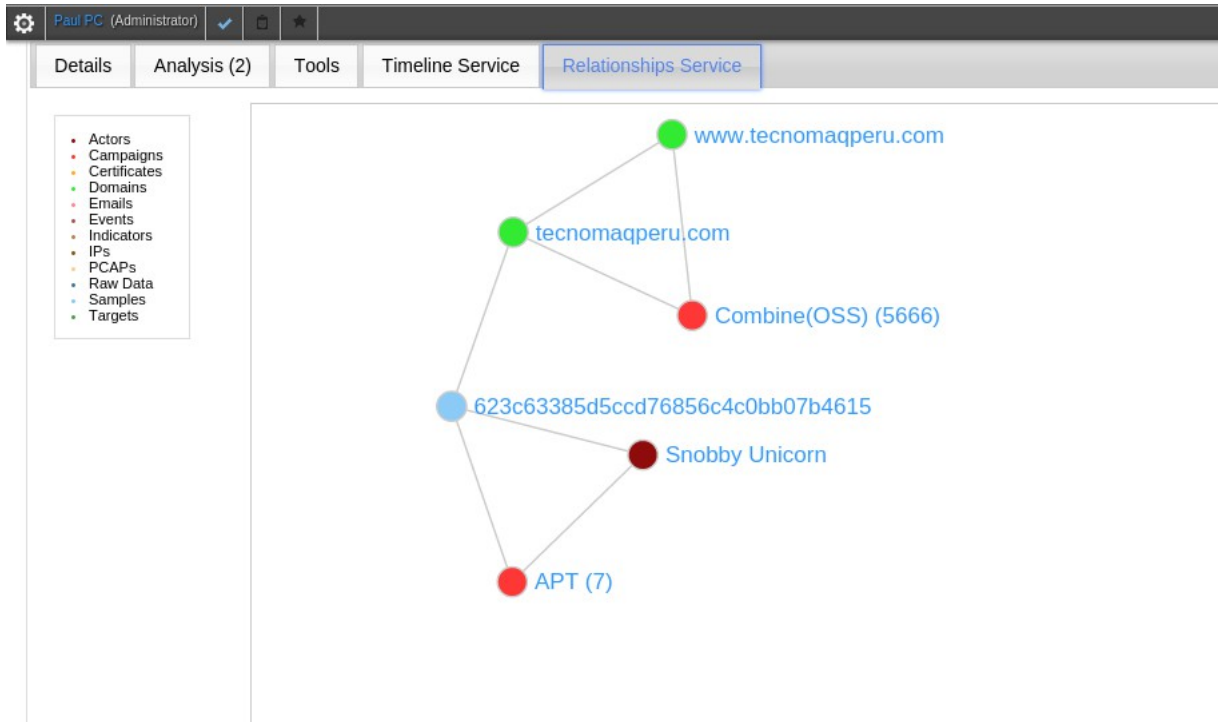


Figure 5.3: Relationships in CRITs

Rule Name ▲	Gr...	Rule Category	Rule Type	Enabled	Response	Event/Flow Count	Offense Count	Origin
Intel.High.Domains	Intel	Custom Rule	Event	True	Email, Notification	0	0	User
Intel.High.Hashtes	Intel	Custom Rule	Event	True	Email, Notification	0	0	User
Intel.High.IPs	Intel	Custom Rule	Common	True	Email, Notification	0	0	User
intel.medium.domains	Intel	Custom Rule	Event	True		0	0	User
Intel.Medium.Hashtes	Intel	Custom Rule	Event	True		0	0	User
intel.medium.ip	Intel	Custom Rule	Common	True		0	0	User

Figure 5.4: QRadar intel-based rules

traffic going in and out of the network may be more effective than having the SIEM check for all the IP addresses in each Firewall, Web Proxy, IDS, DNS, and End Point log entry. Thus, one of this model's premises is a 'control-in-depth' approach, with minimal network control of both outbound and inbound traffic (on the border), Web Proxy, a Next-Generation Firewall, and some visibility of the End Point.

This paper focuses on the incorporation of the following commercial and open source resources to promote the dissemination of threat intelligence with automated decision-making:

Table 5.1: Security Tools Intel Integration

Product	Role	Observable
Combine	Collect OS-INT	IP, FQDN
CRITs	Threat Library store Threat Intelligence observ- able	IP, FQDN, file metadata, targets, emails
QRadar	SIEM log aggregation, alert correlation	IP, FQDN, hashes, file- names, email addresses, userid
Palo Alto	Next-Generation Firewall	FQDN, IP, userid
BRO IDS	Intrusion Detection, Net- work Security Monitoring	IP, FQDN, MD5, User Agent String
Generic Web proxy	Web content gateway	FQDN, IP, userid
Generic Sandbox	Analyze binaries and out- put results to SIEM	MD5, filename
Host IDS	Host intrusion detection record binaries ran on end-points and report to SIEM	MD5, filename, IP, FQDN
Host IDS	Host intrusion detection record binaries ran on end-points and report to SIEM	MD5, filename, IP, FQDN
Generic Email Gate- way	Stop SPAM, send email records to SIEM	Email addresses, subject

Some of the tools are specific since scripting depending on their availability and APIs. This is not actually an appreciation of the standard of the software, but a result of the possibilities of tools availability and used API for the simulation. The distribution of observed ones will look as follows:

For the simulation, the sources of intelligence are sources of OS-INT which are set in Combine by default. After Combine collects the artefacts, it uses the API to upload them to CRITs, populating the source field with the origin of the indicators, the Combine campaign, and a 'medium' confidence level. The measurable will be interpreted and disseminated to detective and

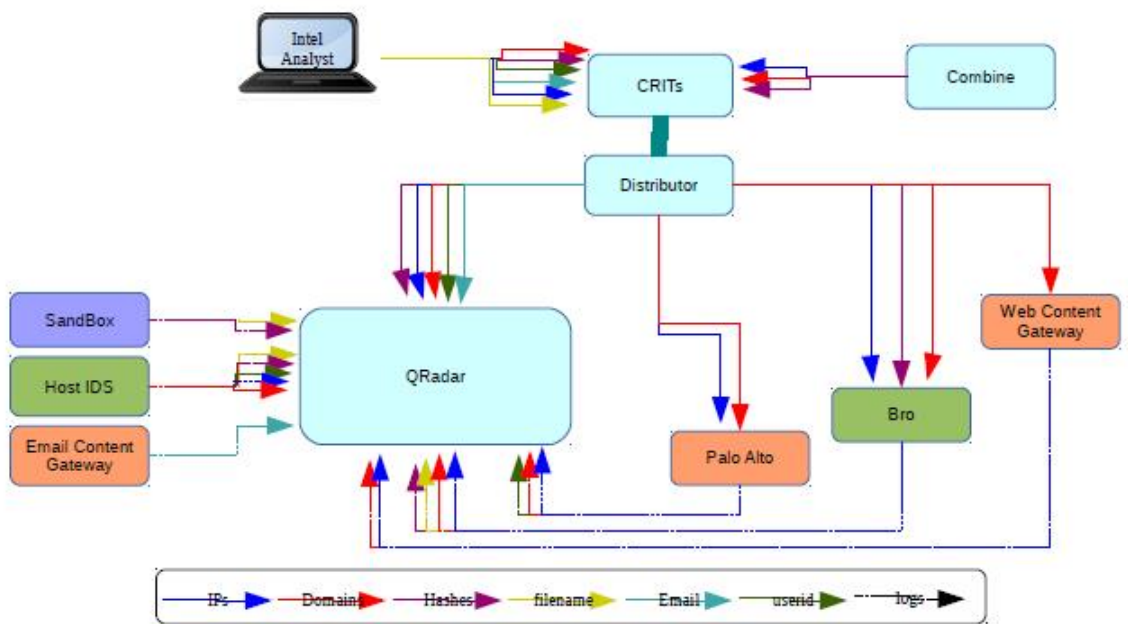


Figure 5.5: Flow of the the monitoring

preventive technologies by the delivery system, Nyx, from CRITs (along with any modifications created by the threat analysts). Such systems would then feed the alerts back into the SIEM, alerting the observed high-fidelity and enabling the defenders to search the enriched logs.

Before disseminating the observable, Nyx attempts to determine by looking at the related campaign the confidence associated with the observable one. It will allow some confidence-enhancing methods:

- If an indicator manages to be connected to different sources, the potential of it being an indicator of performance will be increased on the assumption that various optimization parties also identified it.
- The confidence of the indicator, that should be at least as high as the highest campaign confidence.

Currently Nyx is focused to observe the five categories for the CRIT: IP, domains, samples, emails and targets. Nyx could indeed load and disseminate low-level observables from the following high-level categories: (IP address, domain name, MD5 hash, file name, email address, userID) to the systems concerned:

- Bro IDS: The IP addresses, domains, MD5 hashes, and file names are stored in a text file and made accessible on a Web server. To retrieve the text file, Crontab begins a script in the Bro Manager. A Bro script loads the file through the Threat Intelligence Library.
- Palo Alto: Nyx will send the domains and IPs to the Palo Alto API. Domains would be placed in two custom categories, one focusing on blocking the observable high-fidelity, the other on alerting. Only the most secure IP addresses will be submitted to Palo Alto because it is more used as a blocking tool.
- Web Proxy: The High-Fidelity domains are placed on the webserver in a text file. Most web filters can load a flat-file into a custom blocking category and use the domains.
- QRadar: Nyx must send IP addresses, domains, MD5s, email addresses and user IDs to the QRadar API that are stored in reference sets based on trust in the campaign.

After detective and preventive systems process the intelligence they send warnings to the SIEM. QRadar warns for high-confidence blocking alerts. Based on the learned lower confidence reference sets, QRadar will increase relevance for events and particular threat contain Threat Intelligence pieces, helping them to bubble up to the top to increase the likelihood that they will receive attention during the hunting activities.

5.3 Threat Hunting

Threat Intelligence improves the detection experience of all sources of OSINT, internal and IP address search for the use of CRITs.

Scripts that gather DFIR artefacts can also be deployed using the right-click menu. This should reduce the Identification phase manual tasks, Build a reliable, repeatable process, and make triage simpler.

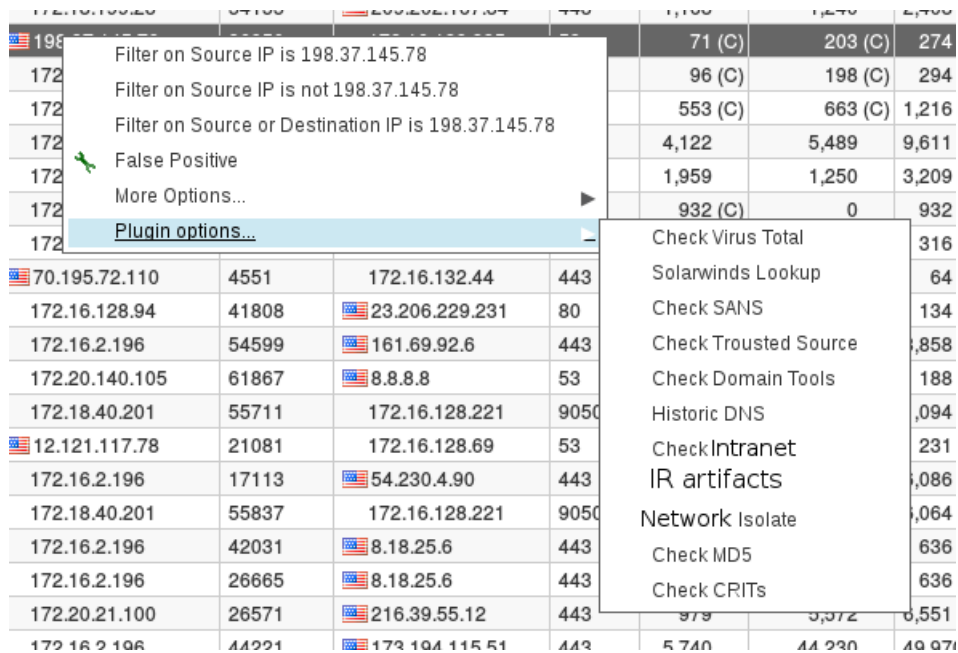
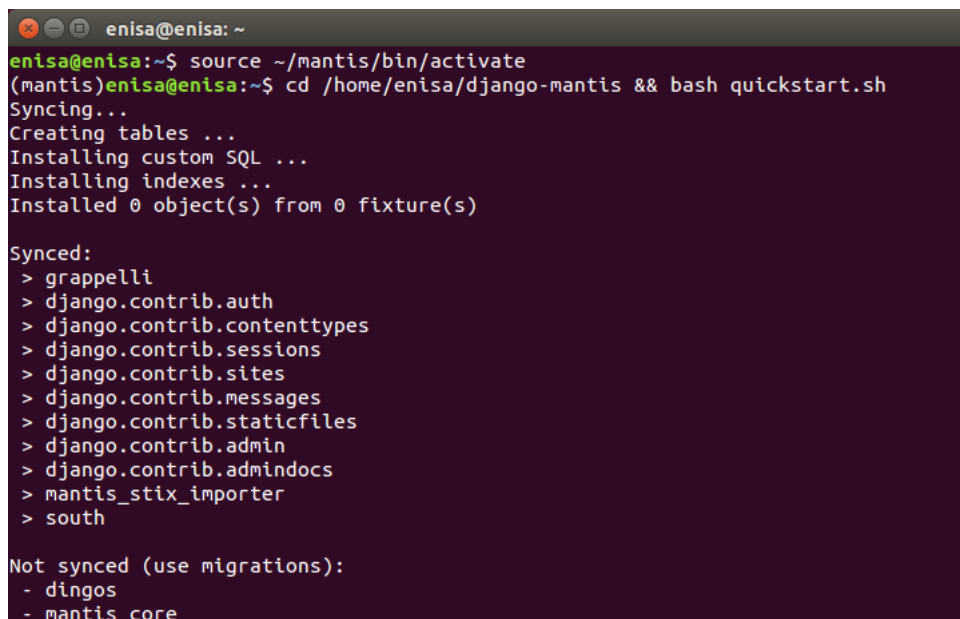


Figure 5.6: QRadar Context Menu

The development of automated actions based on high-fidelity warnings and the automatic launch of the Incident Response scripts was an significant improvement to the existing scripts. A computer displaying signs of Crypto Locker should have restricted access to network drives. (<https://github.com/PaloAltoNetworks-BD/SplunkforPaloAltoNetworks>). Although QRadar provided a good forum for right-click integrations, some vendors of Threat Intelligence provide similar functionality through browser extensions. Some noteworthy examples include CIF, ThreatStream and VirusTotal.

Security systems IDS/IPS or endpoint protection often give the possibility of creating own threat and vulnerability definitions to close or at least narrow the gap between threat/vulnerability detection and vendors response. Again, however, in the implementation process encounter the problem of unstructured security information that slows down the implementation of countermeasures. MANTIS is already installed in virtual machine as shown as figure 5.7. That provide the malware analysis intelligence.

A terminal window with a dark purple background and white text. The window title is "enisa@enisa: ~". The user has entered the command "source ~/mantis/bin/activate" and is now in a virtual environment. They then run "cd /home/enisa/django-mantis && bash quickstart.sh". The terminal shows the following output: "Syncing...", "Creating tables ...", "Installing custom SQL ...", "Installing indexes ...", "Installed 0 object(s) from 0 fixture(s)". Under the heading "Synced:", there is a list of Django apps: grappelli, django.contrib.auth, django.contrib.contenttypes, django.contrib.sessions, django.contrib.sites, django.contrib.messages, django.contrib.staticfiles, django.contrib.admin, django.contrib.admin docs, mantis_stix_importer, and south. Under the heading "Not synced (use migrations):", there is a list: dingos and mantis_core.

```
enisa@enisa:~$ source ~/mantis/bin/activate
(mantis)enisa@enisa:~$ cd /home/enisa/django-mantis && bash quickstart.sh
Syncing...
Creating tables ...
Installing custom SQL ...
Installing indexes ...
Installed 0 object(s) from 0 fixture(s)

Synced:
> grappelli
> django.contrib.auth
> django.contrib.contenttypes
> django.contrib.sessions
> django.contrib.sites
> django.contrib.messages
> django.contrib.staticfiles
> django.contrib.admin
> django.contrib.admin docs
> mantis_stix_importer
> south

Not synced (use migrations):
- dingos
- mantis_core
```

Figure 5.7: MANTIS first run

Since this is first run it is time to import some data to database to search it through. During this exercise we will use some of the samples provided by CybOX Project at <https://github.com/CybOXProject/schemas/tree/master/samples> as samples as shown as figure 5.8.

```

<!-- Create Iran-Oil .exe Trojan file-->
<cybox:Event>
  <cybox:Type xsi:type="cyboxVocabs:EventTypeVocab-1.0.1">File Ops (CRUD)</cybox:Type>
  <cybox:Description>Create Iran-Oil .exe Trojan file.</cybox:Description>
  <cybox:Actions>
    <cybox:Action>
      <cybox:Type xsi:type="cyboxVocabs:ActionTypeVocab-1.0">Create</cybox:Type>
      <cybox:Associated_Objects>
        <cybox:Associated_Object idref="example:Object-8b463e0d-cc16-4036-950e-5eeb09bc51aa">
          <cybox:Association_Type xsi:type="cyboxVocabs:ActionObjectAssociationTypeVocab-1.0">Initiating</cybox:Assoc
        </cybox:Associated_Object>
        <cybox:Associated_Object id="example:Object-b7e0bc39-f519-4878-8fb0-5902554efe1c">
          <cybox:Description>
            The file (us.exe MD5: FD1BE09E499E8E380424B3835FC973A8
            4861440 bytes) is created in the logged in user %Temp%
            directory. The size of the embedded file is 22.5 KB (23040
            bytes) and the size of the created us.exe is 4.63MB. It is an
            odd discrepancy until you look at the file and it looks like the
            code is repeated over and over - 211 times. The file resource
            section indicates the file is meant to look like a java updater,
            which is always larger than 22.5KB and that would explain all
            this padding, which is done at the time when the file is being
            written to the disk.
          </cybox:Description>
          <cybox:Properties xsi:type="FileObj:FileObjectType">
            <FileObj:File_Name>us.exe</FileObj:File_Name>
            <FileObj:File_Path>%Temp%</FileObj:File_Path>
            <FileObj:Size_In_Bytes>4861440</FileObj:Size_In_Bytes>
            <FileObj:Hashes>
              <cyboxCommon:Hash>
                <cyboxCommon:Type>MD5</cyboxCommon:Type>
                <cyboxCommon:Simple_Hash_Value condition="Equals">FD1BE09E499E8E380424B3835FC973A8</cyboxCommon
            </cyboxCommon:Hash>
          </cybox:Properties>
        </cybox:Associated_Object>
      </cybox:Associated_Objects>
    </cybox:Action>
  </cybox:Actions>
</cybox:Event>

```

Figure 5.8: MANTIS import data

After the data import, MANTIS user interface available at on figure 5.9, <https://192.168.186.129:8000/mantis/View/InfoObject/> with the using username and password, user can log into the MANTIS.

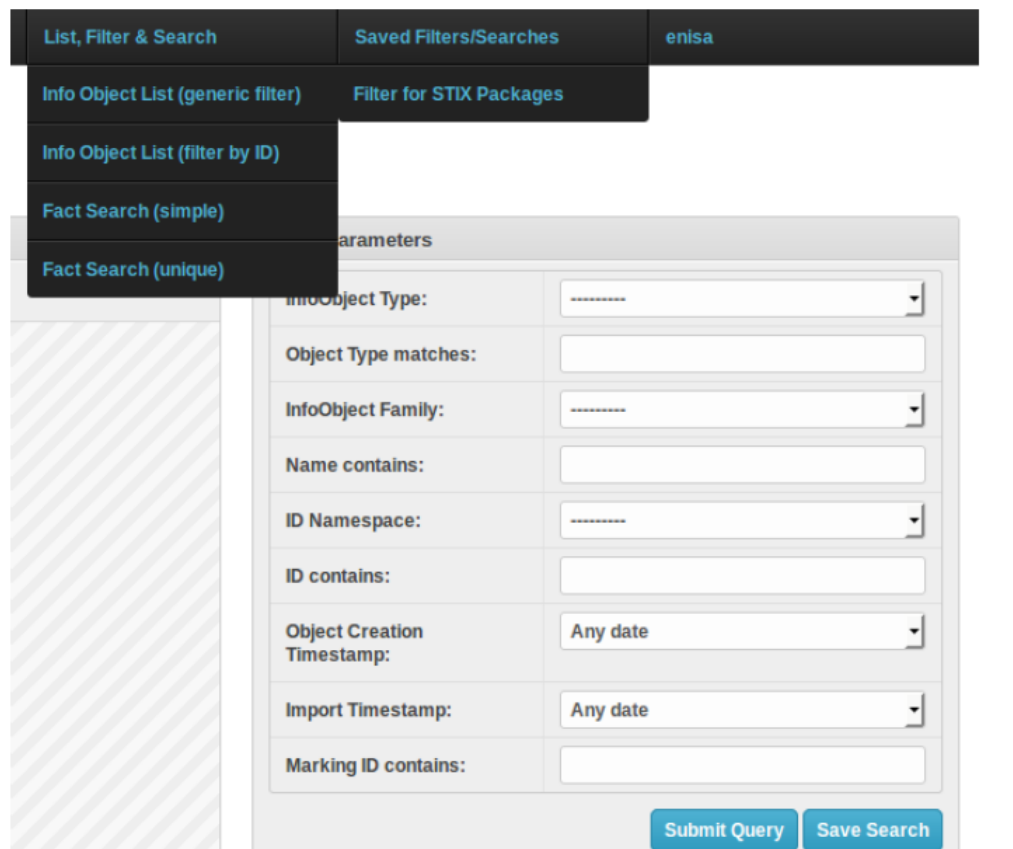


Figure 5.9: MANTIS User Interface

In the Fact-based filtering part of the window will see the search results, while in the Value column there will be e-mail addresses. Select one of these from the list by clicking on the Info Object element. Window as figure 5.10, provide standard e-mail details such as e-mail addresses, subject, attachments etc. Using provided feature on MANTIS is provide the file hashes.

Identifying data						
Identifier	http://example.com/Object-51359587-f201-4383-b032-5a64522fcd7d				Timestamp	2014-08-20T11:48:53.417976+02:00
Type	cybox.mitre.org:EmailMessageObject 2 (http://cybox.mitre.org/objects#EmailMessageObject)				InfoObject Family	cybox.mitre.org 2

Facts						
					Value	Datatype
Properties	Header	To	Recipient	@category	e-mail	String
Properties	Header	To	Recipient	Address_Value	william.abnett@gmail.com	String
Properties	Header	From	@category		e-mail	String
Properties	Header	From	Address_Value		wmorrison89@gmail.com	String
Properties	Header	Subject	Iran's Oil and Nuclear Situation			String
Properties	Header	Date	2012-03-02T07:42:24Z			String
Properties	Raw_Header	Return-Path: Received-SPF: pass (google.com: domain of wmorrison89@gmail.com designates 10.236.185.4 as permitted sender) client-ip=10.236.185.4; Authentication-Results: mr.google.com; spf=pass (google.com: domain of wmorrison89@gmail.com designates 10.236.185.4 as permitted sender) smtp.mail=wmorrison89@gmail.com; dkim=pass header.f=wmorrison89@gmail.com Received: from mr.google.com ([10.236.185.4]) by 10.236.185.4 with SMTP id l4mr5301660yhm.129.1330692273662 (num_hops = 1); Fri, 02 Mar 2012 04:44:33 -0800 (PST) MIME-Version: 1.0 Received: by 10.236.185.4 with SMTP id l4mr4236541yhm.129.1330692265380; Fri, 02 Mar 2012 04:44:25 -0800 (PST) Received: by 10.147.35.14 with HTTP; Fri, 2 Mar 2012 04:44:24 -0800 (PST) In-Reply-To: References: Date: Fri, 2 Mar 2012 07:44:24 -0500 Message-ID:				String

Figure 5.10: MANTIS Observables

Among the facts you can find the information that this file was downloaded by Iran's Oil and Nuclear Situation.doc from <http://208.115.230.76/test.mp4> as shown on figure 5.11.

Identifying data				
Identifier	http://example.com/Object-49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7		Timestamp	2014-08-20T11:48:53.417976+02:00
Type	cybox.mitre.org/FileObject 2 (http://cybox.mitre.org/objects#FileObject)		InfoObject Family	cybox.mitre.org 2
Facts				
	Value		Datatype	
Description	The word document contains flash, which downloads a corrupted mp4 file. The mp4 file itself is not anything special but an OC filled (22kb) mp4 file with a valid mp4 header.		String	
Properties	File_Name	Iran's Oil and Nuclear Situation.doc	String	
Properties	Size_In_Bytes	106604	String	
Properties	Hashes	Hash Type MD5	String	
Properties	Hashes	Hash Simple_Hash_Value	EB0A4FC2B3ED802AD8D0E24C7FC0857	
Properties	Hashes	Hash Simple_Hash_Value	@condition Equals	
Association_Type	Affected		ActionObjectAssociationTypeVocab-1.0	

Figure 5.11: MANTIS Hashes

The description of this file reads This mp4 file causes memory corruption and code execution via heap-spraying code injection and you can infer from the information in the right-hand side box that this file was included in an exploit running the payload us.exe.

MANTIS provide the file hashes on figure 5.11, using extracted file hashes virus total will give the analysis on the mentioned hashes observable where it is malware or not. Sample simulation shown on figure 5.13.

5.4 Summary

This section focused on the proposed framework simulation using open source tools. It was proved that, framework worked as automated defensive systems with threat Intelligence platform integration and with the complex cyber threat intelligence and sharing methods.

Identifying data				
Identifier	http://example.com/Object-8b463e0d-cc16-4036-950e-5eeb09bc51aa		Timestamp	2014-08-20T11:48:53.417976+02:00
Type	cybox.mitre.org/FileObject 2 (http://cybox.mitre.org/objects#FileObject)		InfoObject Family	cybox.mitre.org 2

Facts				
	Value			Datatype
Description	This mp4 file causes memory corruption and code execution via heap-spraying code injection.			String
Properties	File_Name	test.mp4		String
Properties	Size_In_Bytes	22384		String
Properties	Hashes	Hash	Type	MD5
Properties	Hashes	Hash	Simple_Hash_Value	8933598C8B1FA5E493497B11C48DA4F2
Properties	Hashes	Hash	Simple_Hash_Value	@condition Equals
Related_Objects	Related_Object	Iran's Oil and Nuclear Situation.doc (7 facts)		Related_Object
Related_Objects	Related_Object	Relationship	Downloaded_By	ObjectRelationshipVocab-1.0
Related_Objects	Related_Object	http://208.115.230.76/test.mp4 (3 facts)		Related_Object
Related_Objects	Related_Object	Relationship	Downloaded_From	ObjectRelationshipVocab-1.0

Figure 5.12: MANTIS alert generation

```

enisa@enisa: ~/examples
enisa@enisa:~/examples$ cd examples/
enisa@enisa:~/examples$ python cybox_xpath_virustotal.py
b305b543da332a2fcf6e1ce55ed2ea79
{"response_code": 0, "resource": "b305b543da332a2fcf6e1ce55ed2ea79", "verbose_message": "The requested resource is not among the finished, queued or pending scans"}
23e371b816bab10cd9cfc4a46154022c
{"response_code": 0, "resource": "23e371b816bab10cd9cfc4a46154022c", "verbose_message": "The requested resource is not among the finished, queued or pending scans"}
5e17055c51724b0b89ff036d02f5208a
{"response_code": 0, "resource": "5e17055c51724b0b89ff036d02f5208a", "verbose_message": "The requested resource is not among the finished, queued or pending scans"}
e62dadb2856c099a066713883bc12788
{"response_code": 0, "resource": "e62dadb2856c099a066713883bc12788", "verbose_message": "The requested resource is not among the finished, queued or pending scans"}
05552a77620933dd80f1e176736f8fe7
{"response_code": 0, "resource": "05552a77620933dd80f1e176736f8fe7", "verbose_message": "The requested resource is not among the finished, queued or pending scans"}
079028d315d039da0ffec2728b2c9ef6
{"scans": {"Bkav": {"detected": true, "version": "1.3.0.4959", "result": "W32.WoletixC.Trojan", "update": "20140603"}, "MicroWorld-eScan": {"detected": true, "version": "12.0.250.0", "result": "Backdoor.Agent.AAZI", "update": "20140604"}, "nProtect": {"detected": true, "version": "2014-06-04.01", "result": "Backdoor/W32.Agent.14336.AG", "update": "20140604"}, "CMC": {"detected": true, "version": "1.1.0.977", "result": "Trojan-Downloader.Win32.Agent!0", "update": "20140604"},

```

Figure 5.13: MANTIS with virus total

Chapter 6

Analysis of Automated Threat Intelligence Architecture

6.1 Introduction

In the previous section simulate the proposed framework scheme using open source tools. The proposed framework totally complies with the tool used for the simulation process, it is assumed that the correctness of the threat intelligence guaranteed independently according to the tools used. This chapter is to analyze the automated defensive systems with threat Intelligence platform integration.

The Research already introduced the concept for self organized threat intelligence, whose requirements are to:

- Self Organized
- Threat Intelligence architecture
- Intrusion Detection

Threat intelligence sharing tools rely more or less on these needs and many software vendors sell products. While a variety of solutions already exist on the market, the majority of publications investigate fundamental requirements and challenges for the development of threat-intelligence platforms.

To what extent these platforms provide the necessary means for sharing threat intelligence remains unclear since there is no scientific analysis of the state-of-the-art platforms for sharing threat intelligence, and no empirical research has yet been conducted. To fix this void, study seeks to answer the following questions in the study:

- What is the state-of-the-art of threat intelligence sharing platforms?
- What are existing gaps in currently available threat intelligence sharing platforms?
- What are the implications for scientific research in this area?

Research has shown that although interest in this domain has increased considerably in recent years, a common definition of platforms for the sharing of threat intelligence is still missing. While STIX is the extensive de facto standard for describing data on threat intelligence, most platforms do not use its descriptive capabilities to their full. That is illustrated by the fact that most platforms focus primarily on sharing compromise indicators.

6.2 Key Findings

The methodology applied, as described in the previous section, identified the following threat sharing platforms: collaborative threat research (CRITs), malware information sharing platform (MISP), open threat exchange (OTX), collective intelligence framework (CIF).

6.2.1 Key Finding 1: On TI Sharing Platforms

Apart from the standards for the description (e.g. STIX) and sharing (e.g. TAXII) of threat intelligence, research and practice have developed a comprehensive definition and common understanding of what constitutes a platform for the sharing of threats.

Identified platform focuses on sharing the intelligence of threats between organisations. While aggregating information from the users participating in

the platform, the platforms only share data (and not intelligence in its strictest sense) that is automatically aggregated from various available pay-as-you-go sources of information security (cf. Open Source Intelligence). The identified platforms (CRITS) provide a hybrid form of a platform for the sharing of intelligence threats, where they share information. In addition, there are four tools consisting only of a central repository which provides a specific context security information (e.g. information about malware).

6.2.2 Key Finding 2: On Sharing of Indicators of Compromise

The observed networks concentrate mainly on exchanging vulnerability measures, e.g. network Open Threat Exchange (OTX). Compromise indicators provide information helping to detect potentially harmful activities. For instance, compromise indicators are malicious IP addresses, abnormal user activities, malicious file descriptions etc. While the OpenIOC standard is primarily intended to share them, the platforms analyzed use the Observable and Indicator constructs of TAXIIs to describe these.

6.2.3 Key Finding 3 : On TI Platform Availability

There are six free to use threat intelligence sharing frameworks on the market, four of which are open source tools published under the GNU General Public License, including the Malware Information Sharing Platform (MISP), Collective Intelligence System (CIF), Collaborative Research Into Threats (CRITs), and MANTIS Cyber Intelligence Management Framework. The Open Threat Exchange network (OTX) and Soltra Edge are free to use, but have not been released under an open source license.

6.2.4 Key Finding 4 : On TI Platform Availability

There are six free to use threat intelligence sharing frameworks on the market, of which four are open source tools published under the GNU General

Public License, including the Malware Information Sharing Platform (MISP), the Collective Intelligence System (CIF), and Collaborative Analysis. The intelligence lifecycle model, which involves many activities such as planning, data collection, analysis and dissemination, is developed in the sense of information security intelligence. However, from the fact that found, most tools focus primarily on data collection and more or less neglect the other Intelligence Lifecycle activities. The majority of threat intelligence services currently available therefore resemble data warehouses rather than real intelligence sharing systems.

In addition, they provide limited capacity for analysis and visualization and lag behind comparable knowledge sharing platforms and data mining solutions from other fields. This is surprising insofar as the value of these platforms is constrained by the ability of users to interpret, absorb, enhance and react to the information provided. Moreover, only a few platforms provide third-party tools with interfaces that would allow further analysis of the information received as a threat. Threat intelligence sharing platforms currently offer basic analytical capabilities, such as browsing, attribute-based filtering, and information search. In addition, only a small fraction of platforms incorporate pivot functionalities that allow visualization of the relationships between the constructs of the threat intelligence.

6.2.5 Key Finding 5 : On Lack of Automation

Threat Intelligence Sharing Platforms offer limited capabilities for automated data integration. A lot of manual user interaction is therefore required for exchanging and gaining useful knowledge. In addition, the success of a threat intelligence platform depends on the willingness of users to share intelligence that is limited by the free resources available to organizations and the motivation of employees to participate actively. Since most systems lack automated intelligence gathering tools, and more importantly, automatic sensitive intelligence sanitation, these activities still require manual effort. In addition to conventional file importing features, most threatening intelligence sharing platforms lack convenient user interfaces to quickly add new data records and require many user interactions to achieve the desired goal.

6.3 Discussion of Results

Because key findings 1 and 5 (in subsections 6.2.1 and 6.2.5) showed that software vendors have a different understanding of the sharing of threat intelligence, a standardized definition and characterization of threat intelligence sharing platforms is needed. In this context, adopting the wide-spread intelligence life cycle model, including planning, collecting, analyzing, and dissemination activities, into the threat intelligence sharing domain to generate intelligence might be beneficial. Hence, it may be necessary to investigate and define how a threat-intelligence sharing platform can address the different activities within the model. In addition, these attempts to standardize could pave the way for a prospective intelligence sharing network that offers "true" intelligence rather than data warehousing and restricted data analysis capabilities. Furthermore, organizations could also benefit from a common understanding, since it could simplify the selection of an appropriate threat intelligence platform.

Key finding 2 (in subsections 6.2.2) showed that three standards are used to facilitate the description of which TAXII is the most used for threat intelligence. It shows it becomes the de-facto standard in the field. TAXII is a detailed and comprehensive standard consisting of eight constructs which allow the description of a wide range of information related to security and its relationships. Although the number of standards and exchange formats available is currently small, a trend towards use of case-specific definition formats may be noted.

According to key finding 3 (in subsections 6.2.3) most tools share only compromise indicators which can be described by two TAXII standard constructs. The following two conclusions can be taken from that observation: (A) Intelligence standards describing threats are too generic and powerful, or (B) at the moment only low hanging fruits, that is, compromise indicators, are shared. To gain a deeper insight into this issue, empirical research on the information predicted, needed and exchanged within a forum for the sharing

of threat intelligence is required.

The argument for using a threat-sharing platform for intelligence is the reduction of resources. By sharing knowledge and information relating to security. However, as illustrated in main finding 5 (in subsections 6.2.5), the majority of resources are data-warehouses rather than channels for exchanging information. Consequently, organizations must often evaluate the received information which might result in a lot of additional work. To tackle this issue, research in this area should focus on moving away from mere sharing of security data to knowledge and ultimately sharing intelligence through the common framework.

Building blocks for Threat Intelligence Sharing Threat intelligence sharing systems in the proposed framework are composed of a number of functional elements or "building blocks" which include:

- TI exchange models and modes, i.e. who is sharing the information with and how? What is the guiding force behind knowledge sharing? Is it willingly exchanged, or a controlled requirement?
- Commitment rules and protocols, i.e. what arrangements, rules and procedures are in effect for a safe sharing of TI?
- Types of exchanged information, i.e. what information is shared and what is the intention of sharing it?
- Mechanisms of exchange, i.e. How is the information actually shared?
- What type of IDS/IPS use for the threat detection

Security systems IDS/IPS or endpoint protection often give the possibility of creating own threat and vulnerability definitions to close or at least narrow the gap between threat/vulnerability detection and vendors response. Again, however, that is encounter the problem of unstructured security information that slows down the implementation of countermeasures and requires much higher skill level from the implementer.

Throughout the experiment the aim is to use low cost, open source and open architecture techniques to derive as much value from these feeds as possible for the purposes of automated network defense. This includes the deployment of indicators to a simulated network environment and sharing of the exploited and enriched data with other network defenders through automated threat intelligence sharing. The actual deployment of these assets to a network infrastructure greatly depends on the local requirements and the concept of network layout to support deployment of threat intelligence.

The experiment includes the development of Python scripting and the publically available `nyx` - Threat Intelligence distribution libraries to automate threat intelligence to common network defense tools. Threat Intelligence is complex and we need models to be able to conceptualize the problem space and be able to assimilate bulk data in an automated way. The use of structured threat intelligence languages, such as TAXII, allow network defenders to structure data in such a way as to allow automated defense deployments without compromising on strategic context.

As a result, the 'Self-Organized Threat Intelligence System' definition was introduced. A Threat Intelligence Architecture aims at handling cyber threat intelligence data and transforming this data into actionable information that can be distributed to the various platforms and stakeholders. The three principal requirements of a Threat Intelligence Framework are established:

- Facilitates information sharing
- Enables automation
- Facilitates the generation, refinement and vetting of data

A solution for security automation and response would provide flexibility and additional chances of collaboration. Updated threat library and threat collector for the threat sharing platform is provide the reliable data source for the intrusion detection and prevention at the firewall level. Also it is provide the adaptive scalable solution for the threat detection.

6.4 Summary

This chapter analyzed operational level of intelligence mainly concerned with automating the threat detection. According to the simulated tool the Self-Organized Threat Intelligence System definition was introduced at the end of the chapter.

Chapter 7

Conclusions

7.1 Introduction

Actionable Cyber Threat Intelligence (CTI) collection has been one of the key priorities in the information security field in recent years. Given the growing complexity of adversaries, it has now become a requirement for organizations to use CTI's full capacity to track, detect and protect their network assets through automation and readily process IOCs. While the number of organizations interested in designing and implementing CTI powered expert systems at their Security Operation Center (SOC) is that by the day, many still do not know how to take full advantage of CTI and fewer still do so..

Properly using Threat Intelligence might help defend against the advanced attacker as signatures by themselves are proving increasingly less useful. By being able to know a bit more about the adversary and to codify that knowledge into some observables and indicators of compromise, defenders can render some of the attackers infrastructure useless and therefore increase the costs for the attackers. There are various open source and commercial solutions for storing Threat Intelligence and several formats for sharing it. Availability of tools and integration with existing systems is driving the market to a common set of features, focusing more and more on both detection and Incident Response as the quality of the observables makes it difficult to use them in pure blacklists.

Threat Intelligence should become the common information bridge between

security controls. In an optimistic future state, the SIEM and the Actuator should be able to take the information gleaned from the alerts and feed it into a feedback loop back into the Threat Library. For example, a Bro alert for Approximate Command and Control traffic should both tell the SIEM what computer has been compromised and is in need of re imaging as well as what foreign IP is connected to the botnet and which logs associated with it might need a second opinion. The SIEM should then be able to record the foreign IP as malicious and disseminate it to other controls.

The major risk associated with using automation for some of the basic Incident Response steps is that a false positive might lead to network disruption. Avoiding this is akin to optimizing the SIEM rules: the defenders can start with a few critical attack scenarios and create the logic to stop them, the defenders can focus on high-confidence, high signal-to-noise ratio alerts, or start with less disruptive scripts.

The market for Threat Intelligence is still developing, accounting for the difficulties in assessing the value of a Threat Feed. These difficulties are evident in the great oscillations in price between threat feeds. As the market matures, the prices should normalize and better reflect the quality of the observables offered.

7.2 Observation from CTI

This study of the current CTI landscape reveals that Threat intelligence is currently very loosely defined, with little agreed consensus on what it is and how to use it. There is a risk that in the hurry to keep up with the threat intelligence trend, organizations will end up paying large amounts of money for products that are interesting but of little value in terms of improving the security of their business.

It is understood that CTI should be utilized to collect actionable information on the adversary's capabilities, intentions, and ongoing activity useful to the enterprise defense. Shortening the window between a compromise and

when that compromise is detected is the key and possible only by fast and reliable CTI sharing. The goal of such effort leads to risk mitigation by profiling and predicting attacks to block on the left side of the kill chain. The challenge however, lies in collecting quality and actionable IOCs that can be minimized by having better interoperability across security tools through establishing common standards. Having common standards also allow CTI vendors to integrate off the shelf that saves money for both vendor and the client.

Valuable CTI is available in both industry and government. Regulatory compliance guidelines and laws are not enough to safeguard critical data. Collective defense approach is only possible when all parties increase their CTI sharing. Time critical nature of CTI demands quick action and seamless integration, but organizations are not always at liberty to share.

7.3 SOAR vs Proposed Framework

SOAR (Security Orchestration, Automation, and Response) refers to a collection of software solutions and tools that allow organizations to streamline security operations in three key areas: threat and vulnerability management, incident response, and security operations automation.

In the proposed framework initially focused on the built self organized threat intelligence framework for the intrusion detection systems. Which is mostly targeting on the rule set defined at the IDS/IPS threat collector by the advice of threat library. The proposed framework threat is block at the firewall. It directly talk with the firewall to execute the blocking rule set by the threat collector. Threat collector is responsible for the collecting threat information, update the threat library and the rule set generation for the firewall. SOAR is mainly focused on Threat and vulnerability management, Security incident response and the Security operations automation. The same concept is adopt to the proposed framework to enhance the threat action based sharing capabilities.

7.4 Data Feed Providers

Data Feed Providers (DFP) i.e. STAXX, Recorded Future, Hail A TAXII, OTX, Limo etc. act as producers of STIX 2.0 content and OSINT for threat library. Threat Intelligence Platforms (TIP) i.e. ThreatConnect, ThreatStream, Soltra, Arbor Networks, iSIGHT etc. acts as producer and/or respondent of STIX 2.0 content and are primarily used to aggregate, refine, and share CTI across security infrastructure with other devices or security personnel.

Security Incident and Event Management systems (SIEM) i.e. ArcSight, Splunk, QRadar etc. also acts as producer (typically creates incidents and indicators) and/or respondent (typically consumes sightings and indicators) of STIX 2.0 content. Threat Mitigation System (TMS) i.e. Hexadite, IBM, LogRhythm Phantom Cyber, Rapid7 etc. acts on courses of action and other threat mitigations such as firewall or IPS, Endpoint Detection and Response (EDR) etc. Threat Detection System (TDS) i.e. Snort, Bro, web proxy etc. monitors, detects and alerts based on signature matching or anomalies in data flow. Threats are constantly evolving, and the CTI tools used in a security infrastructure must constantly update to be at par with the trend. The effectiveness of defense is only as good as the ability of the network security devices that support it.

Each specific IOC, be it shared via intelligence collaboration or collected internally, has a reason for its existence and a corresponding set of network technologies that would make the best choice for the implementation of detective and preventive controls. More work needs to be done to accurately identify IOCs in a network traffic. Three things are important when it comes to IOCs, that they are accurate from the beginning, are actionable and they should be acted upon while they have a useful lifetime.

7.5 Future Work

A next step for this is to ingest more observables. Specifically, CRITs allows another class of observables called Indicator. It allows for more types of observables and a more detailed confidence and impact rating. If the down-

stream systems will be able to better ingest and utilize the observable, the level of detail would help by better expressing attacker TTPs.

As organizations mature and TI becomes an integral part of day-to-day operations, advanced security teams can start to use TI to understand what the next threats will likely be. TI can help the security team identify changes in attacks and trends in attacker TTPs and plan accordingly for those changes. For example, attackers have recently increased their skills and utilization of Windows PowerShell. Attackers are adding tools and techniques to their arsenals that take advantage of the built-in scripting platform. Organizations can use the change in trends via TI analysis to identify what may be the next attack vector. They can use this knowledge to take proactive steps, such as increasing system logging or disabling unused technologies at the enterprise level.

Finally as conclusion, to get started with Threat Intelligence to download some of the Open Source Intelligence (OS-INT) feeds available on the Internet. Combine comes with a few feeds in the inbound and outbound files, By using CRITs and Combine, performed creating a collection of observables. Nyx is a python tool, that (<https://github.com/paulpc/nyx>) is used to create automation (Self-organized). The focus was on integrating some of the more widely used technologies, including IDS, SIEM Intruder detection is done by using BRO/Zeek IDS rules. Combine and nyx tool used to get observables to CRITS. At the conclusion, TI is not a simple checkbox item. Establishing a program that learns about and acts upon threats to the organization takes time and effort. More often than not, teams that have put in the time have recognized a high return on their investment. The first step is to define what TI means to the organization, keeping in mind that definition that will differ by company, industry, organization size and many other factors. This first step enables the information security team to establish measurable expectations, which will not only aid in determining whether the team has completed its tasks, but also help guide the team as it builds its program.

7.6 Summary

This chapter contains a summary and discussion of the limitations and shortcomings of this approach. At the end of this chapter, conclude the thesis by summarizing the results and introducing possible future works.

References

- [1] M. Abu, S. Rahayu, D. A. Ariffin (DrAA), and Y. Robiah. Cyber threat intelligence issue and challenges. *Indonesian Journal of Electrical Engineering and Computer Science*, 10:371–379, 04 2018.
- [2] S. Brown, J. Gommers, and O. Serrano. From cyber security information sharing to threat management. pages 43–49, 10 2015.
- [3] M. Conti, A. Dehghantanha, and T. Dargahi. Cyber threat intelligence : Challenges and opportunities. *ArXiv*, abs/1808.01162, 2018.
- [4] A. Dehghantanha, M. Conti, and T. Dargahi. *Cyber Threat Intelligence*. Advances in Information Security. Springer International Publishing, 2018.
- [5] A. Fuchsberger. Intrusion detection systems and intrusion prevention systems. *Information Security Technical Report*, 10, 12 2005.
- [6] A. Ghorbani, W. Lu, and M. Tavallaee. *Detection Approaches*, volume 47, pages 27–53. 10 2009.
- [7] A. Ghorbani, W. Lu, and M. Tavallaee. *Detection Approaches*, volume 47, pages 27–53. 10 2009.
- [8] L. M. Kaufman. Data security in the world of cloud computing. *Security & Privacy, IEEE*, 7(4):61–64, 2009.
- [9] A. Lukatsky. *Protect your information with intrusion detection*. It-master. -, 2002.
- [10] M. F. Marhusin, D. Cornforth, and H. Larkin. An overview of recent advances in intrusion detection. In *2008 8th IEEE International Conference on Computer and Information Technology*, pages 432–437, 2008.

- [11] I. Mukhopadhyay, M. Chakraborty, and S. Chakrabarti. A comparative study of related technologies of intrusion detection prevention systems. *J. Information Security*, 2:28–38, 01 2011.
- [12] H. Smith, T. Dinev, and H. Xu. Information privacy research: An interdisciplinary review. *MIS Quarterly*, 35:989–1015, 12 2011.
- [13] L. Spitzner. Applying security awareness to the cyber kill chain.
- [14] I. Winkler and A. Gomes. *Advanced Persistent Security: A Cyberwarfare Approach to Implementing Adaptive Enterprise Protection, Detection, and Reaction Strategies*. Elsevier Science, 2016.