National College of Ireland

Efficient Cyber Threat Intelligence automation using Machine Learning algorithm

MSc Research Project
MSc. in Cybersecurity

Likhith Umesh Salian
Student ID: 23205237

School of Computing
National College of Ireland

Supervisor: Khadija Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Likhith Umesh Salian		
Student ID:	23205237		
Programme:	MSc. in Cybersecurity	**Year:**	2024
Module:	Practicum Part 2		
Supervisor:	Khadija Hafeez		
Submission Due Date:	12/12/2024		
Project Title:	Efficient Cyber Threat Intelligence automation using machine learning algorithm		
Word Count: 7274	**Page Count:** 19		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: LIKHITH UMESH SALIAN

Date: 12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Efficient Cyber Threat Intelligence automation using Machine Learning algorithm

Likhith Umesh Salian

23205237

Abstract

This research attains the focus towards achieving the goal of enhancing the Cyber Threat Intelligence (CTI) automation capabilities by utilising Machine Learning. The CTI aims towards collecting, structuring, detecting, and analysing, the logs gathered from the Network Traffic Analysis tools like Snort. The Information Technology industry constantly faces high severity threats, considering the importance of ensuring the preparedness towards the various cyber threats occurring online. A Network Intrusion detection system will largely help in the detection and analysis of the suspected paranormal events by analysing the behavioural patterns in the logs. The gathered unstructured logs are generated through Snort by self-simulated threat incident from a local Kali Linux virtual machine. The logs are parsed changed to required structure of format which shall be analysed using the Unsupervised machine learning algorithm like k-means clustering algorithm. The resulting data is then represented graphically using a dashboard. This proposed model based on the K-means algorithms aims to provide security solution to businesses and small-scale IT companies in need to deploy its own automated CTI systems.

1 Introduction

1.1 Research Background

In the cyber space, we observe that every day there are various events happen that may have occurred due to a possible incident. The incidents may have either occurred due to a possible cyber-attack or from a mistake by an insider. These incidents are recorded in the computer systems on consecutively as “logs”. The cyber investigator will monitor the log data when an incident is reported by the security analyst. The log data consists of the incident details such as date, time, location, source IP address, and destination IP address, where the incident has been reported.

1.2 Importance

The Cyber Threat Intelligence (CTI) is crucial to detect and analyse various threats occurring in the cyber security landscape. This provides insights on the identity and understanding of the attackers and their preferred targets. The CTI Detection Maturity Level (DML) proposed by (S. Bromander et. al., 2021) is the levels are defined as per hierarchical data an organisation can process to obtained outcomes based on cyber threats at a specific level. This model being adopted by the organisations identifies the threats based on the Tactics, Techniques, and Procedures (TTPs) this approach is a tactical method of CTI. This can also be identified by other methods like strategic, operational, tactical and technical techniques of Cyber Threat Intelligence gathering process. The operational is a short term and less detailed CTI approach consisting of subdomains like campaigns. Whereas the strategic approach consists of the attributes, goals and strategies to identify the threats with long term usable but

less definite approach. The tactical approach makes use of the TTPs and are applicable for long term and includes higher specifications. Lastly, a highly detailed but short term based technical approach of CTI is conducted by identification of Tools, Artifacts, and various indicators. Once the information is detected then the information is shared using a developing solution called as Malware Information Sharing Platform (MISP) by (Wagner et. al., 2016), where the indicators of compromise are shared rapidly on detection. However, the compatibility with the data model shall have to be verified.

The cyber threat intelligence allows the collaboration of knowledge ensuring the businesses remain updated with the latest cyber threats and vulnerabilities challenging the security. The implementation of threat intelligence minimises the risks related to cyber threats on businesses. With proper investments on cyber threat intelligence can enhance our capabilities to detect various threats occurring on the IT infrastructures. The use of Artificial Intelligence (AI) enhances the efficiencies to validate raw data without need for manual work. The use of AI based defence strategies can be automated using threat detection hence minimises the response times and false alarms reported. The adoption of AI for CTI detection application in financial sector where the quantitative application of the AI for identification of the cyber threat intelligence with visual representation from research (E. R. Ndukwe and B. Baridam, 2023).

The CTI also provides an in-depth analysis of the cyber threats in a broad range of understanding of various cyber threats occurring by analysing the various aspects such as tactics, strategies, and actions taken by attackers to perform intrusion into the IT systems. Furthermore, the implementation of the cyber threat intelligence can become a cost-effective approach as a single breach could easily cost millions of dollars and securing from such attacks will generally be cost effective approach and profitable in terms of handling the sophisticated threats occurring across the world. A paper on reference model on the CTI systems highlights the four phases of the Efficient CTI reference model, including CTI problem identification, CTI frame reference construction, CTI frame model construction, and validation (G. Sakellariou et. al., 2022).

The sub-domain of cyber threat intelligence under is rising faster than ever due to emerging threats in the cybersecurity landscape. The various advanced analytics using the machine learning algorithms helps in the detailed identification of the threat patterns and anomalies taking occurring over the network hence enabling the threat identification systems to swiftly detect the anomalies occurring in the system.

1.3 Research Question

Understanding the various criteria and approach being used in the project a suitable research question has been framed as appropriate here. The research question for the CTI automation project is as follows:

“How to automate the recognition of cyber threats over the network by performing network traffic analysis over a local network?”

1.4 Objectives

The overall outcome from the Cyber threat intelligence implementation comprises of:

- Perform network traffic analysis using Snort Intrusion Detection System (IDS).
- Self-Simulated threat incident logs using a local Kali Linux virtual machine.
- Parsing the generated logs to format necessary to conduct log analysis.
- Perform log analysis and automate analysis using Machine Learning algorithms.
- Graphically represent the analysed information through a real time dashboard.

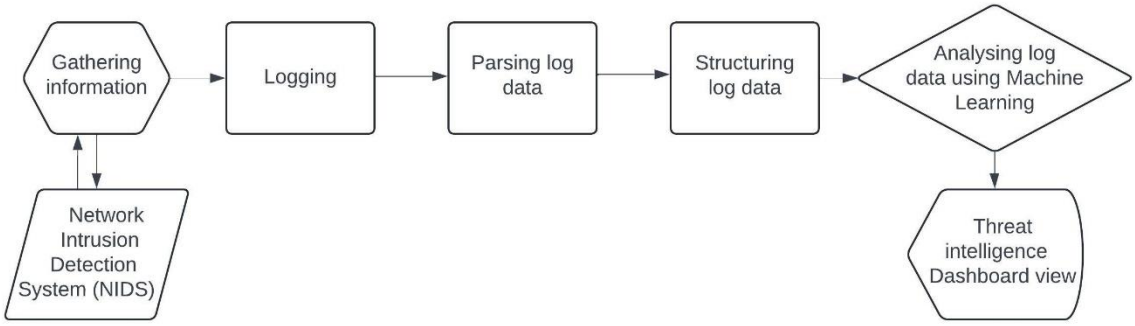

Figure 1.1: System diagram of the CTI automation

1.5 Limitations

The major downsides of the cyber threat intelligence process involve a minimum number of available data being generated from the logs. This may prove to be a challenge on the accuracy of the results being generated. Thereby, it becomes a serious problem in flagging and responding to certain threats in the cybersecurity landscape. Furthermore, yet another problem with the CTI involves the integration of collected intelligence with the decision making due to lack of structured framework or experience to interpret the CTI data.

1.6 Assumptions

The log analysis from the network is significantly complicated process since it requires the various tools and strategies to acquire, process, and detect certain information. There are multiple approach methodologies being used to gain insight into the cyber threat landscape. At certain stages the implementation process could become uncertain, due to various approach strategies of already existing cyber threat automation and number of machine learning models in place. Meanwhile, once the definite approach has been decided and outcomes of the CTI shall be efficiently represented in a dashboard.

2 Related Work

2.1 Cyber threat intelligence automation

As per the paper on Automated Cyber threat intelligence generation on multi-host network incidents (Cristoffer et. al., 2023), the proposed solution uses Network Intrusion Detection System and identify various Tactics Techniques Procedures (TTPs) where the various tactics and attack chains are being used. The strategy uses the combination of the network traffic analysis and MITRE ATT&CK matrix for network mapping of various TTPs. The combination of Whitelist and network mapping and analysing the structure provides rank and map provides us an auto generated CTI reports. Later an CSIRT analyst generates a structured validation support using auto generated CTI reports and CTI feeds. As per the paper on using modular approach to automatic cyber threat attribution with the help of opinion pools (Koen

T. W., 2023), solution provided by the author proposed the modular architecture as an alternative to the unchangeable automation approach. Pairing aggregator is being implemented to produce intermediary results before finally generating an output of Probability Mass Function (PMF). The cyber threat intelligence report is generated by Network Intrusion Detection System (NIDS) from network traffic and Host Intrusion Detection System (HIDS) from host event logs. The alerts generated shall be utilised for the incident detection based on threat attributed from threat attributor. A paper based on the automated cyber threat sensing and responding integrating threat intelligence into security policy-controlled systems (Peter Amthor et. al., 2019), combined the threat intelligence sharing platforms and security policy-controlled systems. The paper discussed various forms of Threat Intelligence and Threat Intelligence Sharing Platforms (TISP). The threat intelligence is of the categories strategic, tactical, operational, and technical have been discussed. Alongside, the Security Policy Controlled System (SPCSS) automated threat response, where the conceptual design was proposed for direct and indirect integration methodologies. A conceptual design based on context aware risk-based Access Control (AC) policies have been put in place to enforcement of this policy in the software systems. The design proposed helps in resolving three major concerns related to strategies utilised by the AC policies to respond to threats, define threat intelligence communication medium to communicate AC policies and their place in security architecture, and standard of the TI architecture to be used. The addressing of the threats based on the various scenarios have been described related to malware attack and Denial of Service (DoS), where discretionary access control mechanism of end user controls and security of the critical IT infrastructure systems are designed respectively. Where the malware attacks are responded by the Discretionary Access Control (DAC) systems where the windows User Access Controls (UAC) are making use of the approach where the untrusted programs are heavily restricted by a security policy called Bell-LaPadula. However, due to the static nature of the security policy, does not make it possible to execute a legitimate functionality which doesn't allow the user decision to override the policy restrictions. This involves drawbacks which involves the access control issues and threat intelligence gathered by Endpoint Protection Platform (EPP) where the hash for the rootkit attack was identified and prevented, thereby the threat intelligence platform can implement the security with minimal changes to systema architecture and no user competence is affected, which also complements well with the antivirus systems in place. The DDoS attacks can be detected based on the heuristic changes in the significant timings and usage patterns used by the botnets which is also an example of static runtime knowledge on self-reliant policies. The areas of improvements in the future exists where the ontological descriptions of threat intelligence events compatible semantics made use by PDP. The automated threat intelligence has novel requirements for SP design on TISP implementation standards. Additionally, implementing a experimentally feasible prototype complimenting the approach used by the author. Lastly, the technical TI shall be formalised threat intelligence and with new design paradigms for a responsive security policy which supports efficient and definite approach in implementing threat responses.

2.2 Machine learning based detection system

As per the experience report on deep learning-based analysis for anomaly detection (Chen Z. et. al., 2022), the paper provides six state-of-the-art neural network-based anomaly detection system, where four methods are based on unsupervised learning and two based on supervised

learning. The research was conducted on the data consisting of 16 million log messages of which totally 0.4 million anomaly instances were detected. The primary phase of log analysis is collection of logs, where the system runtime logs are collected using software tools. Followed by log parsing involving the corresponding parameters like IP addresses, thread name, job ID etc. In the next step, the portioning of logs and feature extraction take place using Machine Learning algorithms. There involve two major portioning methods such as fixed partitioning, sliding partitioning, and identifier-based partitioning. Split partitioning involves the sorting of logs based on the chronological order. Sliding partitioning is based on the distance between the two parameters based on parameter size and stride. Identifier based logs are arranged in the chronological order and then separating them into sequences. Finally, the anomaly detection of the logs detects the anomalies found while searching for log anomalies, like log interruption exception. The ML based algorithms are being used to identify the anomalies. The unsupervised machine learning method used are DeepLog, LogAnomaly, Logsy, and Autoencoder whereas supervised machine learning methods adopted here are LogRobuster and CNN. The deep learning-based log anomaly detection presented a better F1 score compared to traditional machine learning based anomaly detection methods. Similarly, from the paper on the automated vulnerability detection using machine learning on risk assessment in cyber threat intelligence (Dr. Sarah Patel, 2024), where the model focusses on achieving best possible accuracy in Cyber threat intelligence risk and vulnerability assessment detection using machine learning where the supervised, unsupervised, and reinforcement learning methods have been adopted. After researching on the supervised learning methods on vulnerability detection system, it was found that the application of this methodology provided the optimal accuracy of 95% over the network traffic datasets. The major limitation of this methodology is that limited dataset on the analysis of real time network traffic as few zero-day attacks aren’t detected by the system. On the other hand, the use of unsupervised machine learning for the vulnerability detection using clustering algorithms liken k-means, this method evolves over time with attained highest accuracy in detecting the unknown malware signatures. Hence, making it ideal in detecting the zero-day vulnerabilities. Whereas a major disadvantage here would be high number of false positives as not all anomalies are being detected as a threat. Overcoming this problem is possible by using the contextual analysis in unsupervised learning method which will enhance detection capabilities. Similarly, reinforcement learning methods on being implemented on automated threat response system can be adopted in decision making in dynamic environments. The framework for automated threat detection minimises the impact of an attack by having time specific and effective countermeasures. In addition to this, the challenges and future advances to overcome these measures were suggested by the author where the use of ML-based deep learning techniques known as black box model tend to have lower transparency. Therefore, potential trust issues were found on the deployment of the ML-based CTI systems which can be eliminated with the Expansive Artificial Intelligence (XAI) that provides better transparency in the distributed processing. Another challenge in the scalability of the systems exists with the growing size of data being gathered, this issue can be overcome with the hardware and algorithmic improvements with the distributed computing and enhanced hardware accelerators like GPUs could be beneficial.

2.3 Security decision using CTI sharing framework

In a review and research agenda for practice for cyber-threat intelligence security decision making (S. Ainslie et. al., 2023) the background of the paper was achieved from the threat intelligence lifecycle coming from traditional origins of CTI which is mainly practiced in the military intelligence studies. Further the studies on the systems were conducted on various artifacts, objects and information systems useful for practising the CTI. The approach used in this methodology makes use of forward chaining and backward chaining of information. Firstly, a search approach was defined where the certain model and framework for analysis. The research methodology here makes use of the Google Scholar based data source with about 1.4 million records datasets. Out of the chosen records the relevant articles are shortlisted by backward chaining process. Later, the shortlisting of the articles will provide the expected results. The searching and filtering phase of the forward chaining generates the expected results. The critical limitation here is that the authors only referred small number of articles to produce the results. This is due to lack of reference to the relevant CTI information gathering model being analysed. Additionally, a paper on automated machine learning based discovery and cyber threat intelligence using online resources (Rafail A. E. et. al., 2024), where the researchers focussed on discovery of CTI information from the unstructured online resources like online forums, dark web etc. with aid of machine learning algorithms. The threat information gathered is then represented in the structured STIX CTI form. The cyber threat information gathering is assessed based on proof of concept from real life data sources of unstructured data. This information can be represented in a structured shareable information inclusive of Tactics Techniques Procedures (TTPs), Indicators of Compromise (IoC), exploits among many others. The Data Analysis Threat Hunting (DATH) or DATH-CTI relies on Artificial Intelligence and machine learning applications. This concept is applied in 6 different phases consisting of front-end initialisation, gathering CTI process STIX data, processing STIX CTI to OpenCTI, and based on threat generated is displayed on the front end depending on the event type the alerts based on Knowledge Based Name Entity Recognition (KB NER). The CTI-DATH is using a dockerised services based on OpenTAXII fork server compatible with STIX 2.1 and VYU Ache Crawler server. Upon testing the small amount of the CTI data from the data sources were used implementing the ElasticSearch option available on the OpenCTI tool. The Proof of Concept (PoC) combines the Crawling using OpenCTI tool, Preprocessing, scrapped data storage, CTI extraction, STIX mapping, data storage and data dissemination process. The KB NER method of CTI information detection provides highest accuracy on various aspects of detects on known threat actors, attack patterns, infrastructure, tool and malware with 99% accuracy and 1% false positives. The PoC of multi-layer concept of KB NER is based on the real-world unstructured data model can be leveraged with upgradation with better research. Furthermore, several improvements in the CTI-DATH exists where usability and functionality can be enhanced by integrating more data sources, addition of CTI datasets to represent STIX 2.1 model training pipeline, having software-level improvements with user friendly interface and API are recommended and evaluating entity extraction model internals with LLMs are preferrable. The researchers finally suggest having an enhanced CTI capabilities for improved cyber defence. Similarly, from the paper on Investigation of shared cyber threat intelligence with enabling automation in knowledge representation and exchange (Siri B. et. al., 2021), where it focusses on the automation of threat intelligence report sharing automation using a standardised model like STIX.

2.4 Anomaly detection and log analysis for computer networks

On reviewing the paper on deep learning-based anomaly detection and log analysis for computer networks (S. Wang et. al., 2024), the solution for anomaly detection of logs generated over computer networks addresses currently persisting problems with high false positive rates, complex topologies of networks with high dimensional data, and non-stable performance in existing techniques using this approach. The suggested approach combines the Isolation Forest (IF), Generative Adversarial network (GAN) and transformer. This approach utilises the deep learning methods in the network security which further directs towards the future research and development. The model is validated by using the fusion model for anomaly detection, which includes multiple tests on the system performance with notably increase in the accuracy with reduced false positives. Other studies on evaluation of the network security and performance were conducted using Convolutional Neural Network (CNNs) and Recurrent Neural Network (RNN) on identification of anomalous behaviour. The authors compared the previous work related to the anomaly detection using log analysis, where the previous work was based on the Support Vector Machine (SVM), Autoregressive integrated moving average model (ARIMA), rule-based approach using Natural Language Processing (NLP) technique, and rule-based approach with CNN and RNN evaluation models. Few major drawbacks from these papers includes sophisticated knowledge necessary for feature extraction and inflexible to cope with changes, substantial levels of noise in real world applications, manual rule-based strategy comes with higher false positive, machine learning and training of machine learning and deep learning requires huge quantities of data. The integrated model proposed by the researchers has an accuracy of 94.67 % compared to the accuracies of IF-GAN, IF-transformer, and transformer-GAN combined being 86.5%, 88.49%, and 90.25% respectively. There are certain shortcomings of this approach which they shall provide a room for future improvement including the data quality and labelling, the training models requires high computational resources and time and additionally interpretability of the model helps detection of anomaly detection and abnormal activities. The possible solutions to address these issues includes the use of semi-supervised model and unsupervised learning methods to address the data being labelled and make the model more applicable, leverage the computation capabilities to minimise the risks associated, and make the model interpretability transparent and credible.

Limitations and future work: The papers referred here are using various approach in conducing the cyber threat intelligence Automation process. The major limitations in the approach methodologies adopted by other researchers include lack of approach on local network-based log analysis system, lack of preference towards specified approach by authors, and lacking use of combined approach which addressed solving the local network level cyber threats with proper representation using dashboards. The approach used here also attempts to overcome the drawbacks present in anomaly detection system using combination of the IF-GAN-transformer by (S. Wang et. al., 2024).

To address these limitations, the cyber threat intelligence on a local network shall determine the threats generated on network level being detected and represented in a graphical representation of major threats detected on the network. The unsupervised machine learning approach using k-means algorithm is preferred for analysing. This model doesn’t require the data labelling hence saves a huge amount of time in performing the classification of the data. The referred papers lack the representation of the analysed information in a graphical way which the proposed model aims to overcome using a dashboard.

3 Research Methodology

The primary set up of the Cyber Threat Intelligence system includes the logging the various events occurring from in the locally incidents and events occurring in the system. The logs are collectively monitored and analysed from the Ubuntu virtual machine whereas the Kali Linux Virtual Machine is used as an attacker machine which performs the exploitation on the systems. The Kali Linux VM is the attacker machine which simulates the incidents on the target Ubuntu VM where based on the predefined rules which are defined manually. The log alerts are visible and can be used for the analysis of the logs generated by the snort. The Python 3 programming language is used for the analysis and automation of the CTI process. The k-means clustering algorithm based on the unsupervised machine learning method adopted in this project as it is one of the most highly efficient and time saving approach of the machine learning. Additionally, a third-party cloud CTI API database is utilised consisting of the vulnerability blocklists defending from various vulnerabilities and alert generation when incidents are detected. The database datasets of the certain blocklists are used alongside the machine learning program as a training dataset. Similar approach by (S. Wang et. al., 2024) proposed research based on machine learning applications but widely based on the Unsupervised Machine Learning algorithm making use of the Isolation Forest, Generative Adversarial network, and Transformer to obtain the threat intelligence data. Although the combined model provides better accuracy, the development of the model is time consuming and requires expert supervision.

The following model is used for filtering the identified security threats over logs:

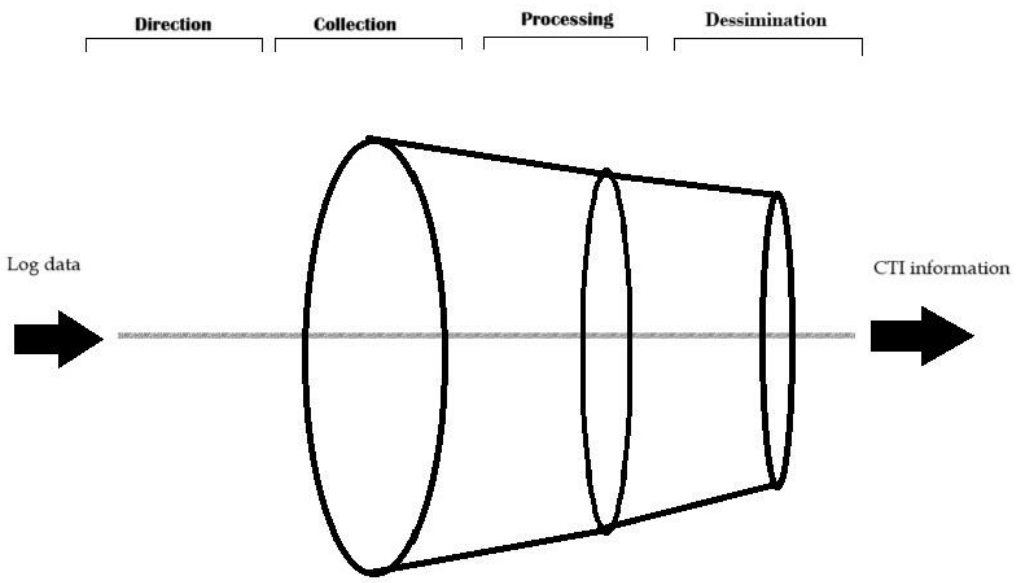

Figure 3.1: Filtering logs to identify relevant security threats

The CTI lifecycle model is used for the implementation and analysis is based on segregating the log data being collected from the Network Intrusion detection System (NIDS). This consists of the various categories of the log data being collected based on the predefined rules and local rule defined by the user which can identify the log data. The four-phase based threat

intelligence gathering process is based on the Australian military standard CTI analysis approach (Army, 2018) presented in the paper (S. Ainslie et. al., 2023). The four major phases of the threat intelligence gathering includes:

i) Direction: Analyse the goals and outcomes desired from the cyber threat intelligence. Identify strategies to achieve various scope, priorities, and certain intelligence queries to be looked upon. Knowing the key intelligence requirements. The priority of the tasks shall be having a goal on strategic, operational, and tactical goals of CTI. The predefined snort rules and the local rules defined by the user enables the Intrusion Detection System (IDS) capabilities to filter specified set of anomalous logs. The crucial automation goals in this phase includes having a definite idea on the previous assessment on risk and data. Identification of the potential indicators of compromise related to the threats and having a leveraged overview on the threat landscape.

ii) Collection: The collection of log data is based on the tasks collecting the cyber threats based on common log threats, dark web, Open-Source Intelligence (OSINT), and human intelligence. The threats are identified based on the Indicators of Compromise (IoCs); for example, ransomware attacks can be detected based on anomalous logs patterns with the help of TTPs. The log collection with respect to the automation, it is necessary to have a multiple APIs and integrations in place working with low redundancy rate. Additionally, enhancing the model with real time log monitoring will further enhance the capabilities.

iii) Processing: The term processing is coined for the process of converting the data into a structured and further usable format. The key directions of data processing include the filtering, deduplication of data and normalisation of log data. Analysis of relationships, trends and patterns are essential. Additionally, it is recommended to use of Natural Language Processing (NLP) and Machine Learning (ML) to analyse the unstructured data plays a important role in automation of the processed data. The automation can also be enhanced with the use of automated tags and threat group classification tools. Once the automation is complete the analysis of the automation scores for future improvements is necessary to ensure the continuous development of the project.

iv) Dissemination: This phase of the CTI ensures to have consistent and useful details of the threat intelligence. The key roles to be upheld here sharing necessary intelligence with Security Operations Centre (SOC) and incident response management teams upon identification of threats. Visualise decision making using the graphical annotation in the threat intelligence report. The timely delivery of the threat information is essential. These tasks can be automated with the integration of the SIEM tools including threat notifications and alerts which can be done via email or any other messaging platform. Generating reports of the alerts from the real time data with automation this ensures that various data stakeholders receive the threat notification on time.

The major importance of the automation in CTI gathering process will enhance the speed with which the incidents are being responded thereby leveraging the detection, analysis, and dissemination capabilities. The scalability of the data also increases with the automation as the necessary data log requirements are fetched on time. The automation ensures the right data is being collected without affecting the accuracy of the data collection system. The

humans working in the threat analysis platform can address other major issues since the log collection takes place in an automated manner reducing the human reliance on the systems.

The use of machine learning capabilities in the python programming will enable us to perform the log analysis. The unsupervised machine learning algorithm K-means has been adopted in this model due its scalable capabilities to advance the systems when necessary. The major benefits of using the k-means algorithm are that unlike supervised machine learning algorithms like SVM, there is no necessity to label the data. Meanwhile, the K-means algorithm can perform the classification analysis using clustering algorithm where the data characteristics are solely identified by inherited properties present in the data. Subsequently, data labelling is a time-consuming process and therefore requires expert intervention in the use of supervised machine learning algorithms. Furthermore, the K-means algorithm performs clustering using the outlier data points. These data points help in the detection of the new or unknown threats identified. This also further makes sure that the data is handled seamlessly by the K-means algorithm.

4 Design Specifications

The project implementation includes a wide aspect of the technical domains to be considered including cloud, computer networks, and virtual machines. The virtual machines consist of the Ubuntu and Kali Linux virtual machines which are target and attacker machines respectively. The machines are configured to their optimum configurations to be capable of performing detection and analysis of the logs using Snort and code execution using VisualStudio Code. The attacker machine uses the tools such as Metasploit, Sparta and Nmap for performing network scanning, reconnaissance, and exploit the machine. The target machine Ubuntu is configured with the Snort, where firstly user configurations are installed with dependencies upon which Snort relies on for its processing. The threat intelligence analysis is performed using the python code running on the VisualStudio Code within the target machine. The python environment variables are declared in the OS along with the python packages necessary to run the program. The main packages used in the program includes scikit-learn(sklearn), NumPy, Pandas, Regular Expressions(re) and Operating System(os).

The network design setup consists of internet, firewall, broadband router, and end point device (laptop). The virtual machine is connected to the host through the local network via Network Address Translation (NAT) network connection. The network relies on the internet connection hence the network with the host system shall have to be configured accordingly. The storage settings is configured to 50GB with four logical processors assigned on both the virtual machines for optimal performance. The Snort IDS is setup with the “snort.conf” file set with the right IP address range, perform the tests to ensure the Snort is working successfully with no errors, the privilege granted to the “/var/log/snort” file directory shall be modified for snort and user account access, and this ensures NIDS system is ready for deployment. The Snort Monitoring code is run on the target machine with the interface and console configured correctly on the command. When the attacker machine launches the network attack the logs are generated by the Snort. The logs generated are stored in the “/var/log/snort” directory in the Ubuntu file system. The analysis of the logs using ML based

Python program is performed on the VisualStudio Code platform installed on the Ubuntu VM.

The ML program is making utilises the K-means Unsupervised machine learning algorithm where the major variations take place on the type of the log collected for the analysis. With increasing types of attacks occurring on the systems, the analysis program can be integrated with the Cloud based API database. The Cloud based API integrations are the third-party feature based on cloud. The CrowdSec Cloud API database has been used in this project, which offers free blocklists on certain vulnerability types and API key generation. Integration of this feature with the machine learning systems will further enhance the security capabilities of the CTI model.

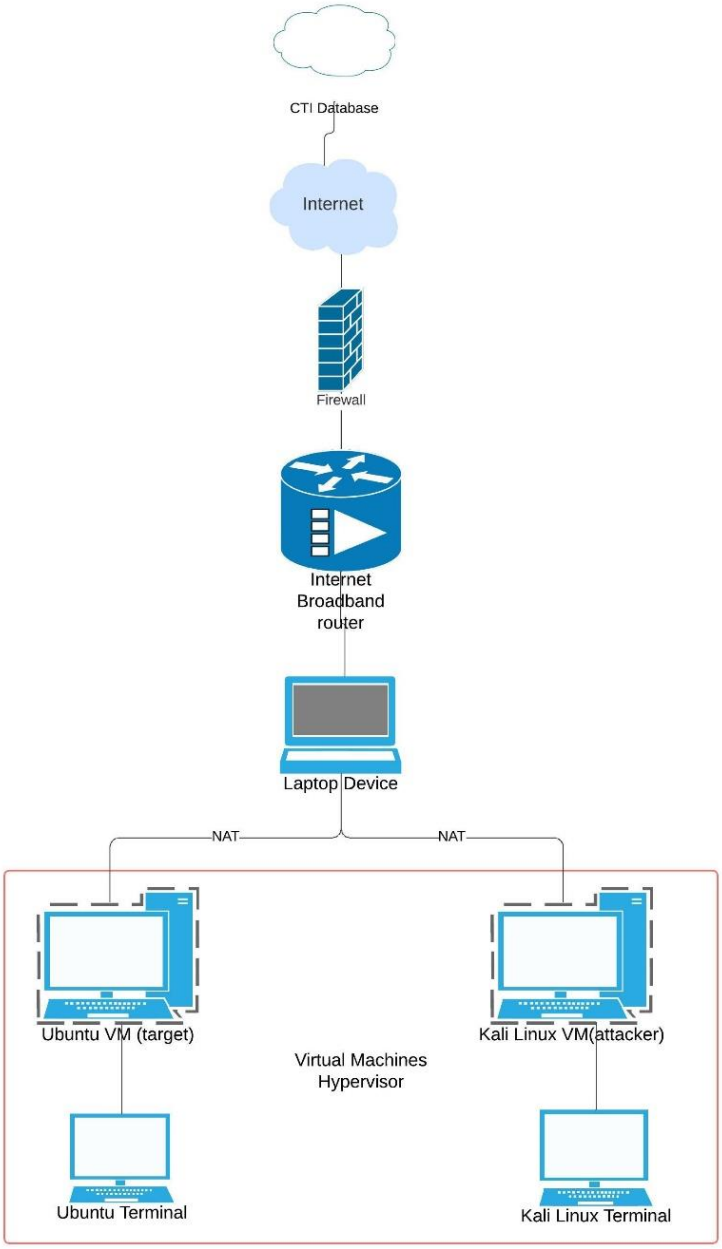

Figure 4.1: Network diagram of the CTI analysis model

5 Implementation

The primary phase of the implementation process starts with the setting up of the virtual machines. There are two different virtual machines in place which includes the Ubuntu 24.04 LTS and Kali Linux 2024.3. The virtual machine is configured to support the local host network with NAT configuration, with the machines having storage configuration of 50 GB hard disk space on each VM, and allocation of 2 logical processing capacity each. The attacker and target machine are set up with the necessary tools and dependencies necessary to install the Snort IDS tool. The Linux packages shall be updated prior installation of the all the dependencies. Upon installing snort, the host IP address and type of address ($HOME_NET or $EXTERNAL_NET) with exact port number with IP address is configured.

The snort dependencies like libpcap, check, DAQ and pcre-2 are installed to ensure the monitoring of the Snort IDS takes place effectively. The snort relies on the installation of all dependencies to function various perform various operations. The snort local rules which focus on the local rules defined by the user has the capability to identify vulnerabilities prevalent in the threat landscape. Most of the attacks taking place on the network level and other well known exploit vulnerabilities are defined in the local rules. The common attack vulnerabilities include reverse TCP attack, FTP attack, log4j, remote code execution etc. If the attack type is not defined in the rules the log data won't be show the type of the attack, however the changes occurring on the log will reflect the anomalous activities taking place in the system. To detect the logs effectively ensure that the test functions run normally the monitoring of the functionalities work optimally when no warning or errors appear on conducting the test on the console /etc/snort/snort.conf file. It is crucial to ensure that the interface and the console in the snort linux command is mentioned rightly. Upon monitoring the logs, the results are stored in the form of logs in the /var/log/snort directory of the Linux file system.

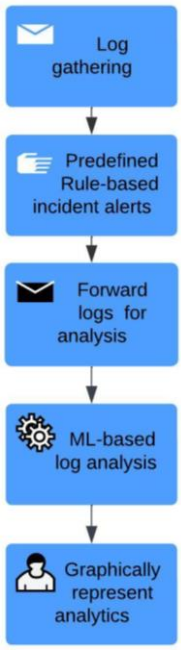

Figure 5.1: Working process of the CTI analysis and representation

The attacker machine Kali Linux has an essential role in the simulation of the log data as necessary. The Kali Linux VM performs exploitation using various tools such as Nmap, Metasploit-Framework (msfconsole) and Sparta. To run the tools with best efficiency it is necessary to have a configuration updated to latest version and virtual machine platform shall be compatible with the operating systems. The predefined rules in the target machine will enable the flagging certain logs if the log if it matches the rules. The snort IDS therefore provides flexibility to enhance the threat detection abilities by defining more rules.

The logs which are generated by the Network IDS system can be analysed by parsing the logs and structuring the log data into a certain format. The parsing is a procedure which involves the segregation of the data present in the logs, and which makes it easier to analyse the information gathered. The machine learning model plays a key role in the analysis of the parsed information to further filter the information. The approach used here is K-means unsupervised machine learning model which clusters the information using various data points and hence does not require the data labelling unlike the supervised machine learning models. The clustering of data enables us to detect the anomalous log entries. This information gathered post the analysis is then graphically represented on a dashboard.

6 Evaluation

The evaluation of the model is most necessary to verify the matching the performance requirements. The tests based on multiple case studies on the snort and other programs are performed which provides the desired outcomes in the form of successful executing, tabular details and graphical representation of data.

6.1 Case Study 1

The snort is tested using the test “-T” in the command with interface “-i” and console “-c” defined accurately. The configuration file of Snort is tested, and the outcomes show the functioning capabilities in the Snort. This approves the utilisation of Snort for log monitoring.

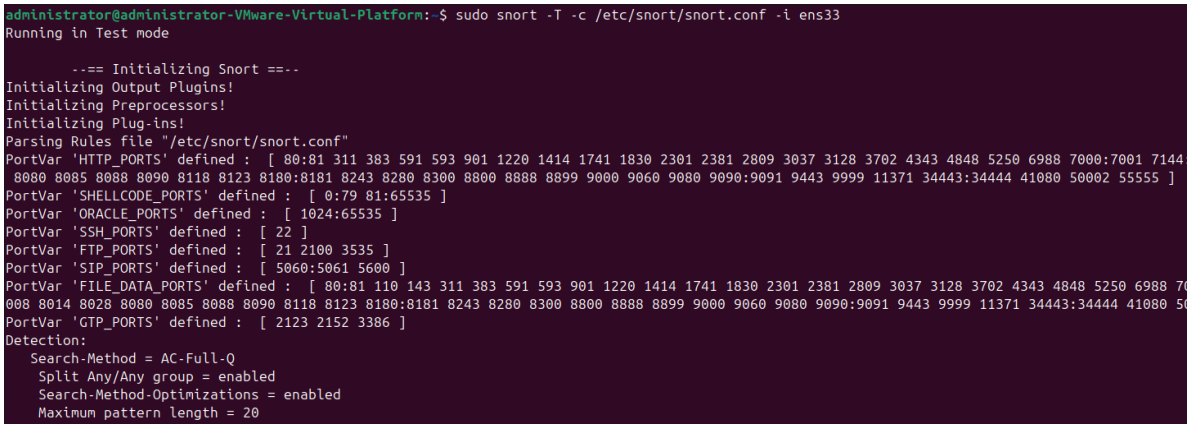

Figure 6.1.1: Snort configuration file test Linux Command

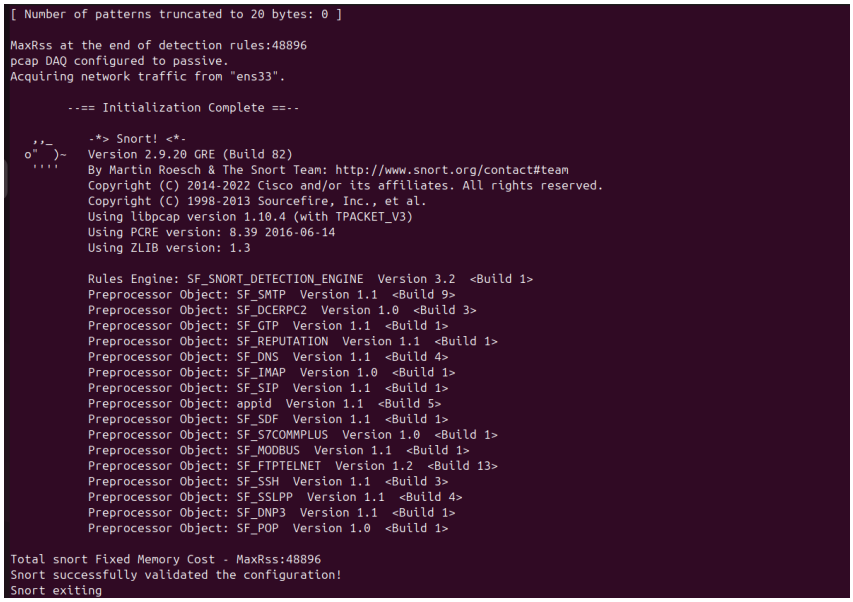

Figure 6.1.2: Successful configuration of snort configuration file

6.2 Case Study 2

The monitoring using snort is performed using the command which includes the interface -i and console -c. The attack is simulated from the attacker machine using the multiple security tools. To optimise the attack capabilities, Sparta tool can be used to automate the attack on the target machine.

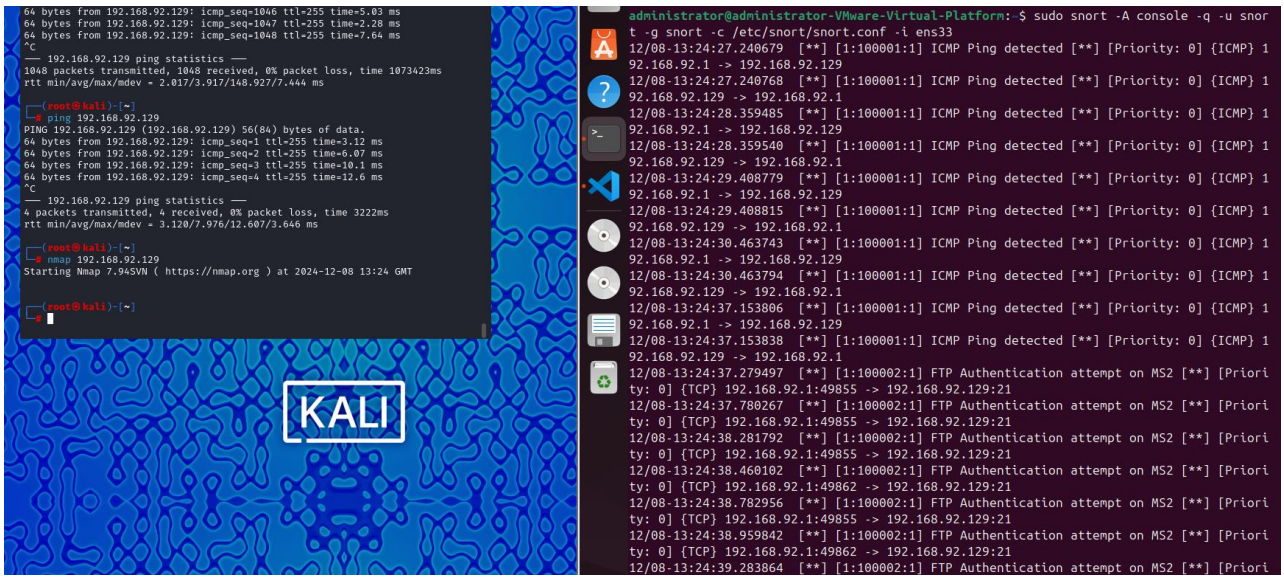

Figure 6.2.1: Monitoring network attack simulated by attacker machine

6.3 Case Study 3

The parsing of the log data is the primary step towards the log analysis process where the log data is parsed. The python code is used for parsing the logs. The sample logs are used for the parsing since the parsing of the log directory was not possible due to Linux permission issues.

```
(.venv) administrator@administrator-VMware-Virtual-Platform:~$ /home/administrator/.venv/bin/python "/home/administrator/Thesis project/log_parsing.py"
          sid                 message                          classification  priority protocol         src_ip src_port         dest_ip dest_port
0  1:2000001:1       ICMP PING detected             Attempted Information Leak         2     ICMP  192.168.92.128     None  192.168.92.129      None
1  1:2000003:1  SSH Brute Force attempt  Attempted Administrator Privilege Gain         1      TCP  192.168.92.128       22  192.168.92.129        22
2  1:2000004:1    Suspicious UDP packet                          Misc activity         3      UDP   192.168.1.150     5353   192.168.1.255      5353
3  1:2000005:1  TCP port scan detected             Attempted Information Leak         3      TCP   192.168.1.150       22   192.168.1.255        22
/home/administrator/Thesis project/log_parsing.py:73: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
/home/administrator/Thesis project/log_parsing.py:88: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.
```

Figure 6.3.1: Tabular representation of parsed log analysis

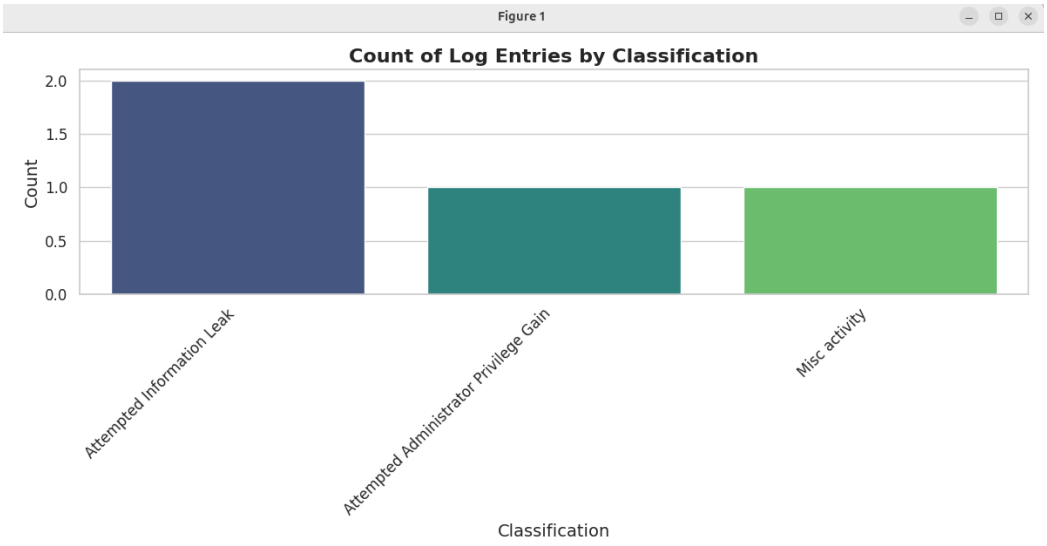

Figure 6.3.2: Graph showing the log entries by classification

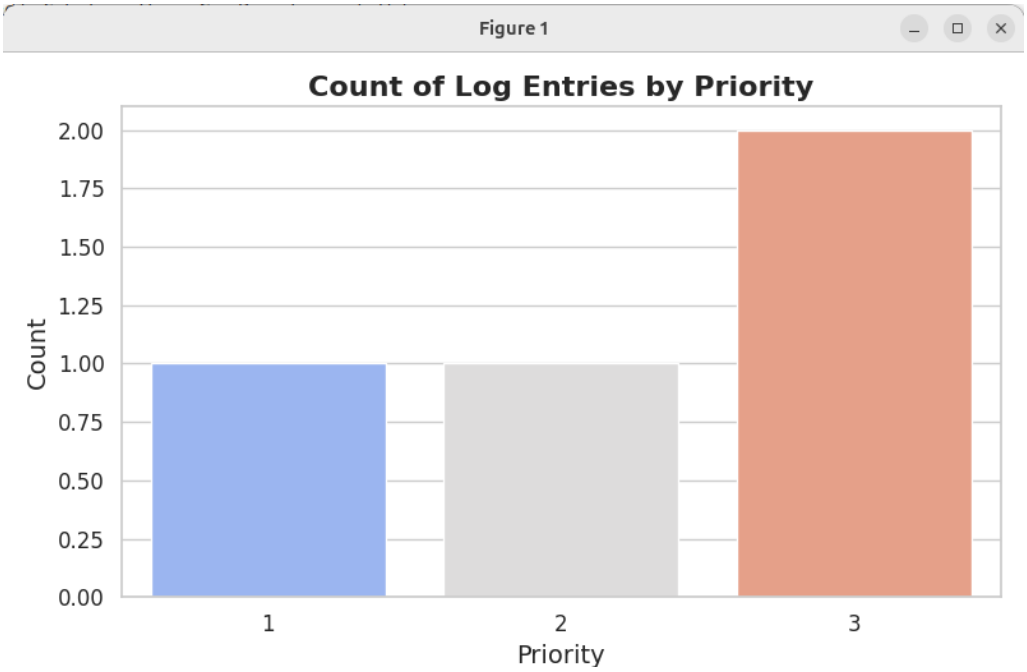

Figure 6.3.3: Graph showing the severity of incident based on priority

7 Discussion

The proposed approach of CTI automation aims towards the having a business solution to enhance the automation of the log gathering and analysis process the organisations and small-scale IT companies. The selection of the machine learning model is a tedious process since it involves major number of the challenges due to multiple algorithms can perform the similar tasks. Although, the efficiency of the system widely varies during the implementation, it is necessary to have a better idea about the resources and capabilities of the system to perform the CTI with respect to technical implementation using the log analysis. The higher amount of data will produce better results through log detection and analysis while using unsupervised ML model. Furthermore, knowing the need for the Cyber Threat Intelligence analysis is essential for organisations. Where the major priority of the CTI automation techniques for the organisation to how the application will enhance the productivity of the systems to provide better quality of security services to its clients and users. Moreover, automation of the CTI systems in the organisational level will help the security analysts to generate the threat intelligence report at a faster rate than manual CTI process. Yet another thing to keep in mind is to analyse how practical the CTI solution to be implemented on the existing systems considering the cyber security budget and overall investment an organisation will have to do to set up the proposed system.

Once a CTI model with the Network Intrusion Detection System (NIDS) can is identified the first step towards deployment of the system is to perform testing. The testing will further help in identification of the right choice of the model to be adopted by the company. The integration with the cloud-based CTI database will leverage the security posture presented by the model. If the results fulfil the cybersecurity requirements of an orgnisation then the model can be deployed in the IT systems in the company with necessary improvements in some sections of the project like automation and log gathering process.

Although the proposed model can achieve the desired CTI results, however, there are a few major concerns in this project that shall have to be addressed. Firstly, the NIDS system shall be of advanced standards for the enterprises, like the Suricata or Splunk enterprise. As per the enterprise security standards, such professional tools are highly preferred by the organisations having its own security setup. These tools also provide a wide range of security features to detect the security threats more efficiently. Yet another solution here can be integrating the analysis of Snort logs with the enterprise standard SIEM for analysis of the logs. The machine learning program shall be used as an add on to this integration providing more practical ways to search for the security threats. The automation process can be done with sophisticated way than the approach in this project where the automation is performed by real-time monitoring of the logs from the Linux file directory system /var/log/snort. This approach towards the automation couldn’t achieve its desired results as the Linux file system didn’t support with the collection of logs although the logs were accessible from the Wireshark tool. On the other hand, the organisations would rather prefer automating the sharing of cyber threat information through the threat information sharing platforms like STIX and TAXII. Therefore, this could encourage in the healthy security practices where the current existing threats in the cyber threat landscape towards for which companies shall take necessary actions and have security preparedness against such forms of attacks. The dashboard visual represented is done with frontend web program using JavaScript using the

Node JS in this project. A more practical approach here can be having a desktop interactive application where the real time data can be visualised using a dashboard.

8 Conclusion and future Work

The research question stated earlier aims towards have an approach of automation Cyber Threat Intelligence (CTI) by log analysis method with the Machine Learning Application. The objectives stated in the research question are achieved to a certain extent including gathering logs and analysing the sample logs being gathered. Considering the research and implementation phase of this automation of the cyber threat intelligence involves log gathering, parsing, automation, detection and analysis. Due to limited number of resources to choose from for CTI analysis, the implementation of the log gathering process is done using the third-party like Snort which is capable of detection the attack performed over the network based on the predefined rules on the snort. However, the automation of the logs could not be fulfilled due to challenges with regards to the access to read the logs by the VisualStudio Code. The recommended approach towards this problem can be the integration of the gathered Snort logs with the enterprise standard tools like Splunk to perform further analysis in automation process of the logs.

The future development of this project can focus on leveraging to automation capabilities of log analysis. This could benefit by time and cost saving approach in the IT industry since the automation will reduce the human involvement in the collection of log data. The visualisation of the data can be enhanced with the development of the interactive real time data on a desktop application hosted in python is preferable over a dashboard hosted on a webpage.

In conclusion, the proposed CTI automation model has various multiple areas of improvement including the use of machine learning model, automation techniques and dashboards. Additionally, it is necessary also to understand that the objectives achieved in this CTI model is aiming towards the gathering logs, parsing and analysing the log data, automating the gathering process and achieving the desired outcomes of Cyber Threat Intelligence process

References:

Ainslie, S., Thompson, D., Maynard, S. and Ahmad, A. (2023). Cyber-threat intelligence for security decision-making: A review and research agenda for practice. *Computers & Security*, [online] 132(132), p.103352. doi:https://doi.org/10.1016/j.cose.2023.103352.

Army Australian (2018). *LWD 2-0 Intelligence. Department of Defence Retrieved from, Canberra, ACT*. [online] DRNET Defense. Available at: http://drnet.defence.gov.au/ARMY/Doctrine-Online/Pages/%20Home.aspx%20..

Bromander, S., Swimmer, M., Muller, L., Jøsang, A., Eian, M., Skjøtskift, G. and Borg, F. (2021). Investigating sharing of Cyber Threat Intelligence and proposing a new data model for enabling automation in knowledge representation and exchange. *Digital Threats: Research and Practice*, 3(1). doi:https://doi.org/10.1145/3458027.

Chen, Z., Liu, J., Gu, W., Su, Y. and Lyu, M.R. (2021). *Experience Report: Deep Learning-based System Log Analysis for Anomaly Detection*. [online] arXiv.org. Available at: https://arxiv.org/abs/2107.05908.

Eke Roberts Ndukwe and Barilee Baridam (2023). A Graphical and Qualitative Review of Literature on AI-based Cyber-Threat Intelligence (CTI) in Banking Sector. *European Journal of Engineering and Technology Research*, 8(5), pp.59–69. doi:https://doi.org/10.24018/ejeng.2023.8.5.3103.

Ellinitakis, R.A., Konstantinos Fysarakis, Panagiotis Bountakas and Spanoudakis, G. (2024). Uncovering Hidden Threats: Automated, Machine Learning-based Discovery & Extraction of Cyber Threat Intelligence from Online Sources. *2024 IEEE International Conference on Cyber Security and Resilience (CSR) Workshops*, pp.1–6. doi:https://doi.org/10.1109/csr61664.2024.10679473.

Leite, C., Hartog, J.D., Dos, D.R. and Costante, E. (2023). Automated Cyber Threat Intelligence Generation on Multi-Host Network Incidents. *2023 IEEE International Conference on Big Data (BigData)*, 23(979-8-3503-2445-7). doi:https://doi.org/10.1109/bigdata59044.2023.10386324.

Patel, D.S. (2024). Machine Learning-Driven Risk Assessment in Cyber Threat Intelligence: Automating Vulnerability Detection. *Journal of AI-Assisted Scientific Discovery*, [online] 4(2), pp.107–114. Available at: https://scienceacadpress.com/index.php/jaasd/article/view/177.

Sahrom Abu, M., Rahayu Selamat, S., Ariffin, A. and Yusof, R. (2018). Cyber Threat Intelligence – Issue and Challenges. *Indonesian Journal of Electrical Engineering and Computer Science*, 10(1), p.371. doi:https://doi.org/10.11591/ijeecs.v10.i1.pp371-379.

Sakellariou, G., Fouliras, P., Mavridis, I. and Sarigiannidis, P. (2022). A Reference Model for Cyber Threat Intelligence (CTI) Systems. *Electronics*, 11(9), p.1401. doi:https://doi.org/10.3390/electronics11091401.

Teuwen, K.T.W. (2023). A Modular Approach to Automatic Cyber Threat Attribution using Opinion Pools. *2023 IEEE International Conference on Big Data (BigData)*, [online] 23(979-8-3503-2445-7). doi:https://doi.org/10.1109/bigdata59044.2023.10386708.

Wagner, C., Dulaunoy, A., Wagener, G. and Iklody, A. (2016). MISP. *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, WICS 16'(978-1-4503-4565-1). doi:https://doi.org/10.1145/2994539.2994542.