Journal of Network and Information Security
11 (2) 2023, 17-29
http://www.publishingindia.com/jnis

AI – Enabled Honeypot

Taha Arshad[1] and Santhosh Menon[2*]
[1]BSc Information Technology Student, Middlesex University Dubai, United Arab Emirates.
Email: ta912@live.mdx.ac.uk
[2]Lecturer, Department of Computer Engineering and Informatics, Middlesex University Dubai, United Arab Emirates. Email: s.menon@mdx.ac.ae
[*]Corresponding Author

Abstract: The growing prevalence and impact of cyber-attacks have led many countries to rank cybersecurity failure as a top risk. Honeypots offer a means to detect attacks and enhance security measures by enticing attackers to compromised devices and collecting data during their interactions. Although Artificial Intelligence (AI) has the potential to strengthen cybersecurity by detecting attacks more quickly and accurately, its adoption in practice remains limited. This project was developed to address the increasing number of cyber-attacks in the era of cloud computing and remote work. The study employed a unique methodology of using AI and Machine Learning to identify patterns in data and improve security measures. The research focused on SSH attacks, which involved mass scanning, brute force attacks, reconnaissance commands, and file uploads. The data extracted from the Cowrie log files was heterogeneous, making it challenging to analyze and utilize for training a machine learning model. To address this, feature engineering was performed to create new features and transform existing ones. The study shifted from a binary classification of traffic to analyzing the behaviour of attackers and predicting their next moves. The machine learning algorithm used was LSTM, which achieved an accuracy of 98% after tuning the parameters. The study concluded that AI could ease the burden on SOC analysts and allow them to be more productive by learning adaptively and finding new patterns that could speed up the process of identifying, containing, and responding to attacks.

Keywords: AI honeypot, Cyber-attacks, ELK stack, LSTM, Machine learning, SSH attacks.

I. Introduction

The importance of network security has increased with the increased network penetration and rise of cyber-attacks. On 28 October 2022 alone, over 200 million cyber-attacks were observed [1]. Cyber-crimes are expected to cost the world around $10.5 trillion annually by 2025.

According to The World Economic Forum [2], many countries like the UK, New Zealand and Australia rank “cybersecurity failure” as the number one risk. With the increased sophistication of attacks, detecting them has become increasingly complex, resulting in the detection time being either multiple months or no detection at all [3].

Artificial Intelligence is emerging as one of the top International Risk Mitigation Efforts in cybersecurity. Research has shown that AI has the potential to strengthen cybersecurity by detecting attacks faster and with higher accuracy than human teams.

This rise in cybercrimes highlights the deficiencies of devices that every company depends on for day-to-day tasks. Organizations, large and small, are forced to devote significant resources and utilize the latest technologies to ensure data confidentiality, integrity and reliability.

One of the active measures available to the security team is a honeypot. This is a deception-based approach where network administrators deploy specially designed devices inside a monitored

network to lure attackers to scan and compromise the device. The value of honeypots comes from data collected during the attackers' interaction with the machines. The network administrators can use this data to detect attacks, understand their taxonomy, and enhance security measures [4].

A. Aim

This project aims to understand the techniques used in honeypot solutions to increase network security. The findings will be used to identify research gaps and understand how new technologies, such as Artificial Intelligence, can be deployed to create more intelligent security devices. Kiah and Zakaria [4] state that all security devices depend on human configuration and maintenance efforts. This dependency makes it challenging to deploy adequate security solutions. Incorrect configuration results in missed detections, thereby voiding its purpose. Furthermore, with new technologies, services, operating systems, and devices being introduced frequently, modern security solutions need to be dynamic and adaptable to changes.

Here we are presenting a more comprehensive system that aims to improve the overall security of a network by detecting attacks faster and with greater accuracy than humans. We look at utilizing the power of AI to analyse data gathered from the honeypot to understand the tactics used by attackers. The design of a honeypot solution that uses AI techniques, its configuration and deployment are shown. Finally we test the solution to show how the AI-based solution aids in detecting attacks faster without human intervention.

II. Background

A honeypot is a security tool used to attract and detect malicious activity on a network. It is often used to gather information about cyber attackers' tactics, techniques, and procedures and identify new types of attacks. The literature reviewed commonly focuses on analysing and predicting SSH bot attacks, which have become more prevalent with the increase in IoT devices and the capabilities of attackers to leverage automation and AI to mimic human behaviour and launch distributed attacks.

One example of using a honeypot for cybersecurity is a Duckiebot, a low-cost robot built to study autonomy, to detect vulnerabilities in self-driving cars. Joyon [5] had configured the SSH and Telnet module and demonstrated how an attacker could access the self-driving vehicle and install malware. The honeypot enabled the author to log all the attacker's actions and alert the administrator through email and Discord that a potential attacker had gained unauthorised access.

A. AI-Powered Honeypots

Artificial Intelligence (AI) allows machines to replicate human thought and action by learning from experience. The approach of using Artificial Intelligence in honeypots, known as adaptive honeypots, has been enhanced by using machine learning to gather more information from the attacker and predict cyber threats quickly and accurately. Pauna and Bica [6] and Suratkar *et al.* [7] proposed using reinforcement learning to increase the interaction with the attacker in order to extend the session time and retrieve more meaningful logs. The AI's goal was to find the best response to the command executed by the attacker to increase the session time. Five actions were available: allowing the command, blocking the command, delaying the command, substituting the command, or displaying an insulting message. An interesting observation in the test in [6] is the distinction between bots and human attackers, with only 10% of attackers leaving the system when an insulting message was displayed. It is worth noting that medium to high interaction honeypots are ideal for these implementations because they allow services to be used with a specific or complete level of access. The honeypots used in these projects were Kippo and Cowrie.

There has been increasing interest in using AI-powered honeypots for classifying network traffic. A study by Lee *et al.* [8] showed how honeypots could be coupled with machine learning for botnet detection in smart factories with many IoT devices. The study implemented a simulation of a smart

factory environment using two Raspberry Pis and a few IoT devices and applied machine learning techniques through Weka and R-studio to compare the results. The results showed similar accuracy and false positive rates, achieving up to 0.96 accuracies and 0.26 false positive rates. However, more working parameters could have been used to meet the scale of a smart factory.

As network traffic volume increased, researchers began to see the potential of honeypots and started treating the traffic collected as data input for machine learning models. Zammit [9] presented an AI-enabled honeypot that applied a decision tree algorithm to classify traffic as malicious or non-malicious and alert the user through a web interface when a malicious IP address was detected. Martínez Garre *et al.* [10] conducted similar research but with a significantly higher number of features, including session duration, first packet length, sent packet minimum and maximum size, received packet minimum and maximum size, and URLs. This enabled the authors to make better predictions, as a more significant number of features allowed the model to find subtle nuances and patterns in the dataset. However, a large number of features can also result in overfitting, in which the model is specifically tailored to fit the dataset.

Both studies emphasize the importance of false positive and false negative results in the testing phase, which is one of the main drawbacks of using AI in cybersecurity. A false positive would mean that a malicious IP is detected as non-malicious by the machine learning model, allowing the attacker to gain access to the network without the administrator being alerted. A false negative would mean a non-malicious IP is detected as malicious, leading to false alarms and restricted access for authorised users.

Owezarski [11] used unsupervised learning techniques in classifying network traffic through honeypots. The author stated that unsupervised learning presents many advantages, such as the ability to work in an unsupervised manner without initial configuration and the ability to run in real-time on any network monitoring device to detect and classify attacks.

B. Intrusion Detection System

Artificial Intelligence (AI) is increasingly being deployed in security solutions to enhance Intrusion Detection and Response. Honeypots, which can detect novel attacks, provide invaluable data that can be used to refine IDS signatures. Two notable products leveraging AI for real-time cyber threat detection and response are Vectra AI and Darktrace Cyber AI.

Vectra AI emphasizes automating threat detection by understanding attacker behaviour and focusing on their Tactics, Techniques, and Procedures (TTPs) [12]. They study these methods before creating their detection models and use algorithms to correlate various attacker behaviours. A notable technique they employ is LSTM to identify command-and-control channel attack behaviours.

Darktrace Cyber AI, in contrast, employs a multi-layered machine learning approach. It gathers data from diverse sources, uses unsupervised learning to model typical behaviour, and then employs supervised learning to score detected anomalies [13]. The system can autonomously act to counter threats. Advanced techniques like packet capture, deep packet inspection, and natural language processing are used to extract data insights. Furthermore, they utilize reinforcement learning and game theory to determine the best response strategy for each threat.

Challenges faced by these systems include accurately identifying new or complex attack techniques, ensuring data privacy and integrity, managing noisy data, and ensuring that threat neutralisation actions don't disrupt organizational operations.

III. Methodology

The methodology being employed for this project is *Simulation*. A real-world honeypot solution is usually complex and part of a more extensive network. A virtual machine is utilized for deploying the honeypot solution. Various software tools and frameworks are used to configure the honeypot and implement the chosen AI technique.

A. Honeypot Implementation and Architecture

T-Pot was utilized for the project with an additional focus on traffic gathered from the Cowrie honeypot [14].

The proposed system consists of a virtual machine being deployed in the cloud. The T-pot honeypot is operated in the virtual machine along with other services to visualise the sessions and extract logs to perform machine learning algorithms. The below diagram provides a general overview of the framework. When an attacker is in the honeypot, all his actions and commands are recorded and stored as a log file. The ELK stack is used for visualisation to understand the type of attackers in the honeypot. Once the logs are extracted, pre-processing is done with the help of Python to prepare the data for training the machine learning model to determine malicious and non-malicious users.

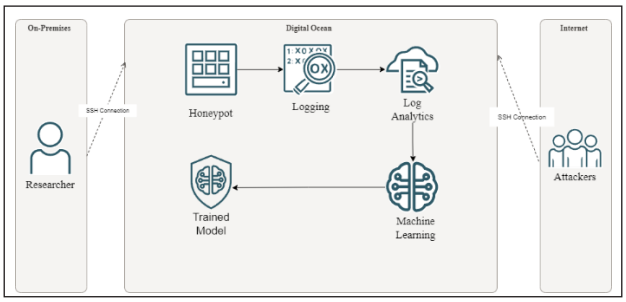


Fig. 1: Network Diagram

B. Log Analysis

For the analysis, the ELK stack provided by T-pot was utilized. This centralised logging and monitoring system allowed for analysing and visualising data from various sources, including logs generated by the T-pot honeypot.

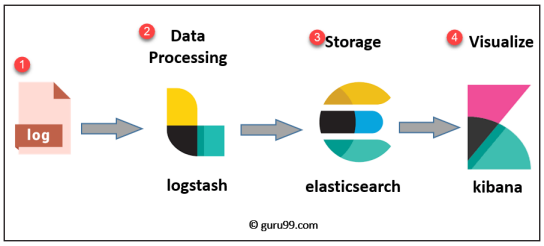


Fig. 2: ELK Stack

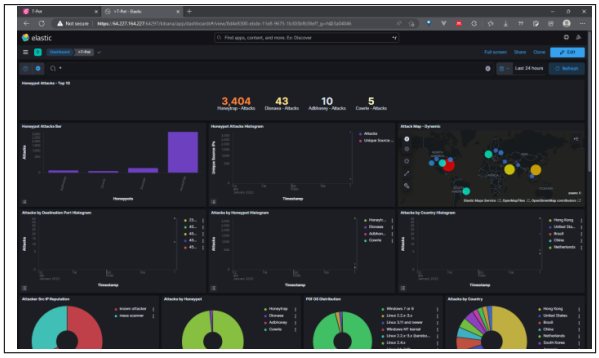


Fig. 3: Kibanna Dashboard

Fig. 4 shows the logs the honeypot captured on the deployment day. Receiving attacks soon after deployment could mean that mass scanners or bots are scanning cloud IP ranges. The visualisations below show the attacks that took place between 19-Feb-2023 to 26-Feb-23.

Fig. 4: Cowrie Dashboard

In one week, the Cowrie honeypot logged 50,296 attacks.

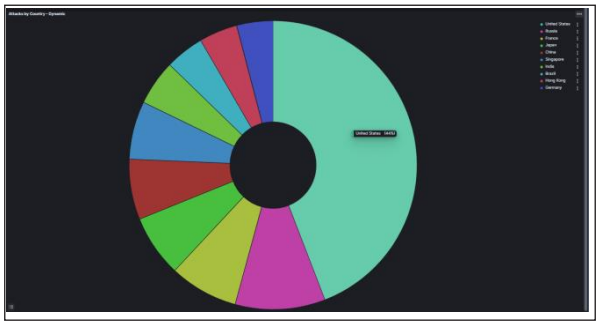


Fig. 5: Attacks by Country

Examining the attacks by country revealed that 44% of attacks originated from the United States. This could be due to hackers using US cloud services to organize attacks or botnets. Many attacks came from

known attackers with a reputation for malicious activity.

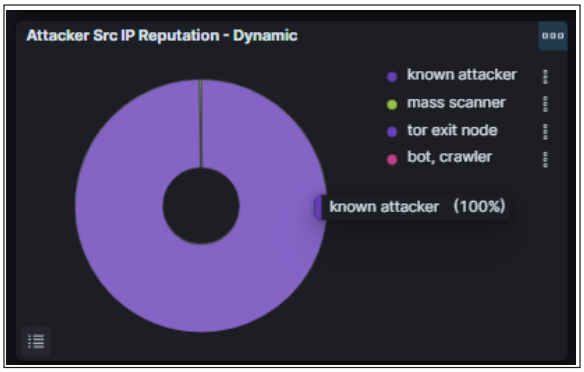


Fig. 6: Source IP Reputation

Tag clouds for usernames and passwords show the attempts made by attackers to gain unauthorised access into the system by using various credentials.

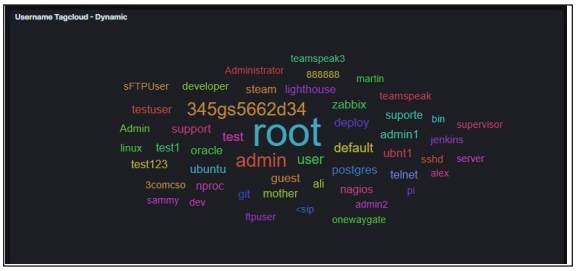


Fig. 7: Tag Cloud for Usernames

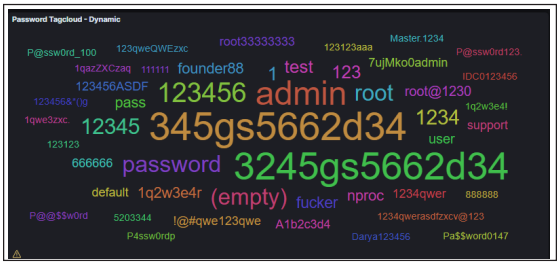


Fig. 8: Tag Cloud for Passwords

The larger the word, the more frequent its usage. Several basic password attempts appear on commonly used username and password combinations in brute force attacks.

Command inputs also provided interesting observations. Most commands, like “uname -a” and “cat /proc/cpuinfo,” were attempts by attackers to gain more information about the system to plan their attack.

Upon research, the command “which ls” returns the executable file path that would be executed if the user typed “ls” in the terminal. This command helps determine which version of a command is being used, especially when multiple versions are installed on a system. For example, some Linux distributions include GNU and BSD versions of standard utilities such as “ls”, which have different options and behaviours. This information can be used to identify potential vulnerabilities in the system, as different versions of the same command may have different options and behaviours.

Similarly, the command “-ia .ssh” was used in various ways. The command “-ia .ssh” could help a cyber attacker remove the immutable attribute from the “.ssh” directory, which may contain SSH-related files and configurations. This would allow the attacker to modify or delete these files, potentially compromising the system’s security. Additionally, if the “.ssh” directory contains SSH keys used for authentication between client and server, an attacker could use them to gain unauthorised access to other systems that trust those keys.

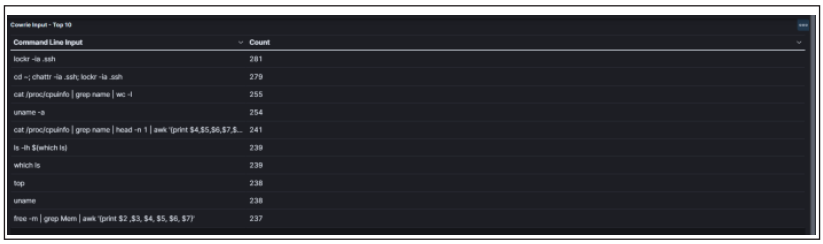


Fig. 9: Top 10 Command Line Inputs

Additionally, files and scripts uploaded by the attacker through the SSH/Telnet terminal were examined.

Cowrie - Top URI Downloads

Filename	T-Pot Path (/data/cowrie/downloads)	Count
http://80.68.196.6/ff	dl/b7d62d1a145ddda241e624ef94ab31fcca1a13f79e130d0a704586e35745282a	1
http://89.241.247.72:2172/.i	dl/020f1fa6072108c79ed6f553f4f8b08e157bf17f9c260a76353300230fed09f0	1
tftp://200.155.147.174/.i	dl/020f1fa6072108c79ed6f553f4f8b08e157bf17f9c260a76353300230fed09f0	1

Fig. 10: Top URL Downloads

URL analysis was performed on Virus Total. The first URL was categorised as potential spyware or malware, while any security vendors did not flag the second one.

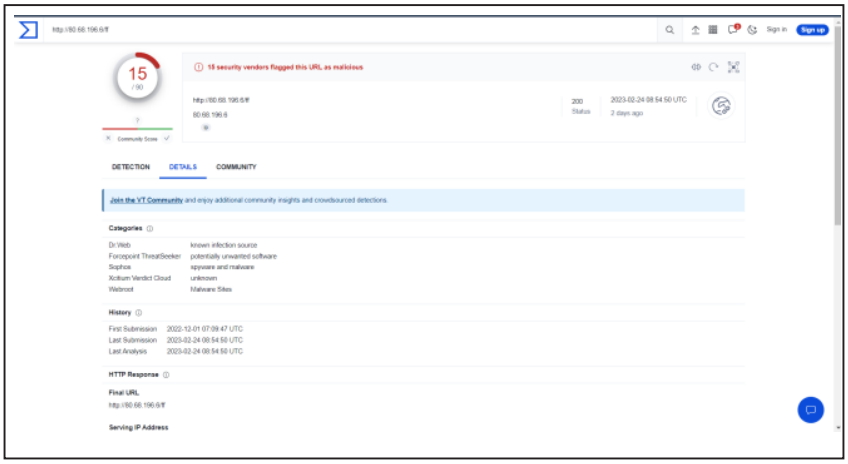


Fig. 11: Virus Total Report-1

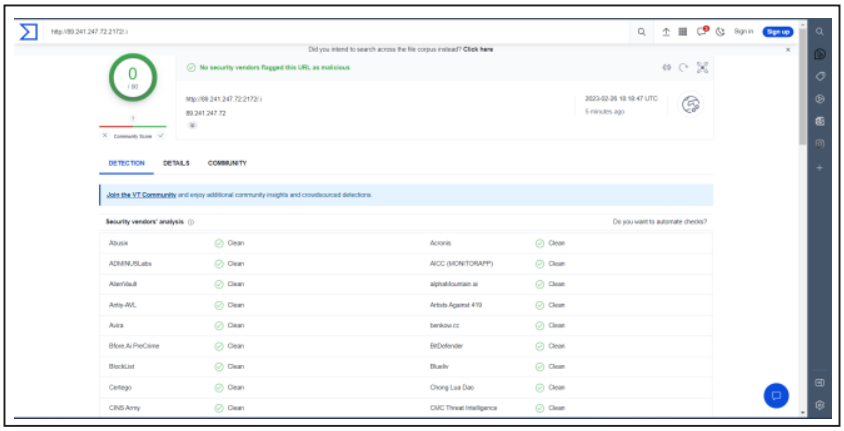


Fig. 12: Virus Total Report-2

In conclusion, the above observations emphasise the importance of implementing effective security measures in a system even before opening services for connectivity.

C. Log Extraction

Log files were transferred from the Droplet to a local system using Spaces, an S3-compatible object storage designed for large files. The command-line tool, S3cmd, facilitated the management of these Spaces. The process involved archiving and uploading the log files folder to the Digital Ocean Space (Fig. 13 and Fig. 14).

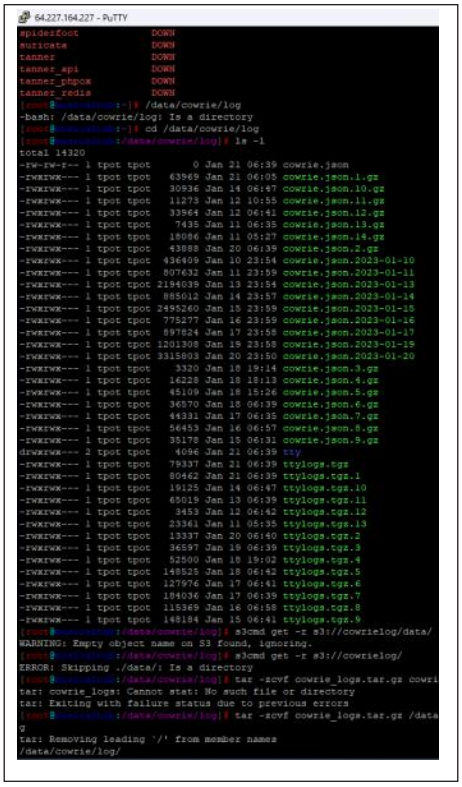


Fig. 13: Archive Folder

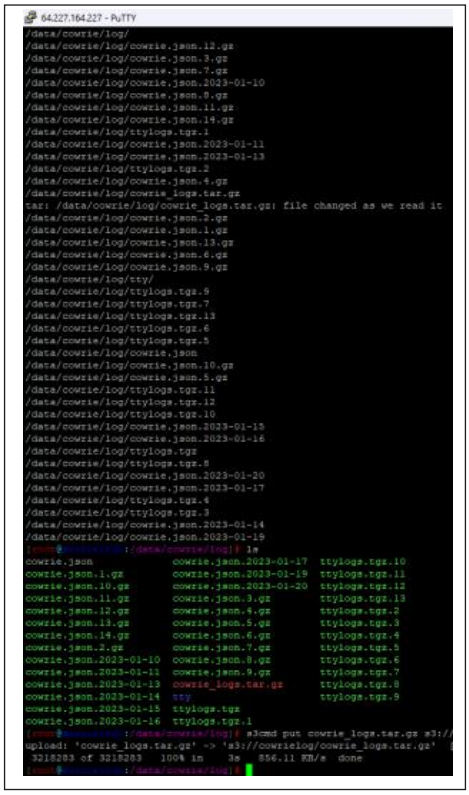


Fig. 14: Upload to Space

The archived folder was subsequently visible in the Digital Ocean Space.

D. Data Cleaning

The following attributes were initially shortlisted to be utilized in the model:

Event Id: A unique identifier for each event or log entry. This can be used to track specific events and determine patterns of activity.

Input: To identify the commands or actions associated with the malicious activity.

Ttylog: The terminal session's input/output (I/O) data. This can provide insight into the commands and actions taken by a user during a session.

Size: The size of the input/output data. This can be used to identify unusually large or small amounts of data, which may indicate malicious activity.

Username: The username associated with the event. This can help identify specific users who may be engaging in unauthorised activity.

Password: The password associated with the event. This can help identify attempts to gain unauthorised access to a system.

Src_ip and Src_port: To identify the source IP addresses and its port. This can help identify the location of the attacker or the compromised system.

Message: To examine the activity description for keywords or patterns associated with malicious behaviour.

Timestamp: To identify patterns of malicious activity over time.

E. Feature Engineering and Extraction

Closely analysing the dataset shows that similar data is linked through the session number. A session consists of a sequence of events that take place during a login session. This allows us to monitor the activity of the attacker inside the honeypot. These activities are further classified in the form of events. The most notable ones include login, command and file download.

Fig. 15: Session with Login Event

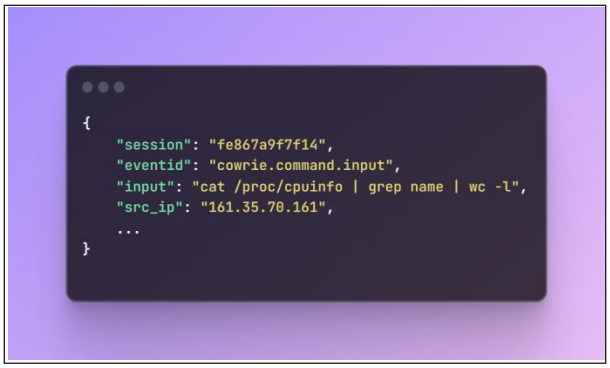


Fig. 16: Session with Command Event

Fig. 17: Session with Download Event

Cowrie produces varied data, with some events having unique attributes. To make this data usable for a learning model, it was consolidated into a single feature table. A "command" column was introduced to store this data, with destfile fields, usernames, and passwords (separated by "/") moved to this column. Commands were then extracted without parameters and categorized to reflect attacker tactics.

Originally, the aim was to label traffic as Malicious or Non-Malicious using APIs like AbuseIPDB API [15]. However, with 90% of the traffic found to be malicious, the focus shifted to analyzing attacker behaviour and predicting their actions, using the Command Type Column as the class attribute.

Command classifications were based on the MITRE ATT&CK tactic and research by Sadique and Sengupta [16]. The categories include:

Discovery: Checking system configurations, e.g., lscpu.

Initial Access: Trying to access the host, e.g., root/admin.

Escalate Privilege: Gaining superior system access, e.g., sh.

Download: Transferring files into the system, e.g., tftp.

Run: Executing programs or scripts, e.g., crontab.

Cover Track: Concealing malicious activities, e.g., dd.

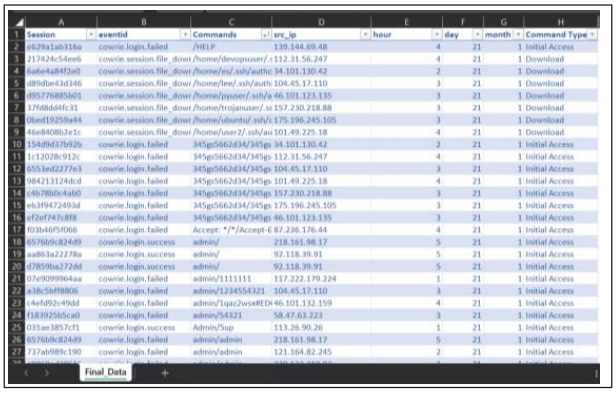


Fig. 18: Final Dataset

IV. Machine Learning Model

A. Recurrent Neural Networks

The ML model chosen for this dataset is Long Short-Term Memory. LSTM is a Recurrent Neural Network designed to learn and remember patterns in sequences of data. They are different from traditional neural networks because they have inputs and outputs that are independent of each other and can only be passed forward.

B. Training and Validation

The classification report is a statistical summary of the performance of the trained classification model. It provides several statistics that are useful in evaluating the accuracy and effectiveness of the model.

The report includes metrics such as precision, recall, F1-score, and support for each class label in the dataset. After experimenting with the parameters, the following adjustments resulted in the highest accuracy of 98 percent:

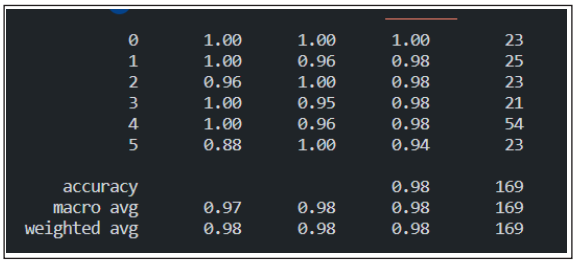
epoch size = 100.

batch size = 64.

dropoutrate = 0.3.

optimizer = RMSprop.

Results



0	1.00	1.00	1.00	23
1	1.00	0.96	0.98	25
2	0.96	1.00	0.98	23
3	1.00	0.95	0.98	21
4	1.00	0.96	0.98	54
5	0.88	1.00	0.94	23
accuracy			0.98	169
macro avg	0.97	0.98	0.98	169
weighted avg	0.98	0.98	0.98	169

Fig. 19: LSTM Model Final Result

Comparison

TABLE I: PERFORMANCE COMPARISON BETWEEN ML MODELS

Model	Weighted Avg	Macro Avg	Accuracy
Before	86%	90%	86%
After	98%	98%	98%

C. Model Performance

To enhance the productivity of network administrators, an automated system paired with a web interface was developed. This approach offers:

Efficient handling of large datasets, eliminating the need for manual processing in tools like Microsoft Excel.

Utilization of Python, a widely recognized language in network security, offering extensive libraries and resources.

A user-friendly dashboard for visualizing network traffic, demonstrating the seamless integration of machine learning.

TABLE II: USABILITY TESTING RESULTS

Sr. No.	Test Item	Status (Pass/ Fail)	Remarks
1	Correct display of the website on different devices and browsers	Pass	The product was developed with desktops and laptops in mind. Therefore, the dashboard does not look good in small-screen devices.
2	Proper functioning of the “Run” button for Step-1: Conversion	Pass	Error and Success messages are shown.
3	Successful conversion of .gz file to .csv file	Pass	Some of the files are not properly formatted in JSON. The script is able to handle these formatting errors.
4	Proper functioning of the “Run” button for Step-2: Pre-processing	Pass	Error and Success messages are shown.
5	Correct pre-processing of the dataset	Pass	Correct pre-processing is performed.
6	Proper functioning of the “Run” button for Step-3: Running ML Model	Pass	Error and Success messages are shown. Takes time because the model.

Sr. No.	Test Item	Status (Pass/ Fail)	Remarks
7	Correct application of the LSTM model for threat classification	Pass	Some IP tactics are not accurately classified because of the model having 98% accuracy.
8	Display of the table dashboard with key information	Pass	Dashboard appears after model is run. Only 25 records are shown.
9	Proper functioning of the "More Info" button in the Action column	Pass	New table is shown after information is generated.
10	Correct generation of additional information using AbuseIPDB API	Pass	Correct information is generated.
11	Successful display of IP reputation, country, ISP, and mitigation techniques	Pass	Information is displayed as expected.

D. Data Processing with Python Scripts

Data Conversion: Log files, stored as .gz, are converted to CSV format for easier manipulation.

Command Extraction: Commands are extracted from input columns, ensuring only relevant data is retained.

Data Reorganization: Username and password data are merged, and other columns are restructured for consistency.

Event Filtering: Specific events are filtered, and timestamps are broken down into more detailed components.

E. Backend Development

The backend, powered by Express.js, offers various functionalities:

File Management: Temporary files are deleted post-processing to maintain a clean workspace.

Script Execution: Python scripts are executed as child processes, ensuring efficient data processing.

Traffic Classification: A pre-trained model classifies new traffic, identifying potential threats.

IP Address Analysis: The AbuseIPDB API provides insights into IP addresses, offering details like reputation and location.

F. Frontend Design

The frontend, crafted with Vue.js, offers a lightweight and user-friendly interface, showcasing the processed data in an intuitive manner.

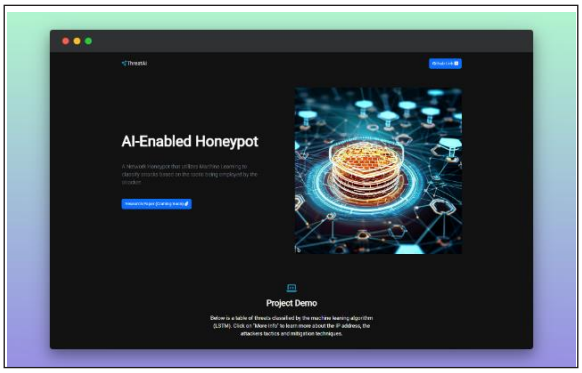


Fig. 20: Webapp Screenshot-1

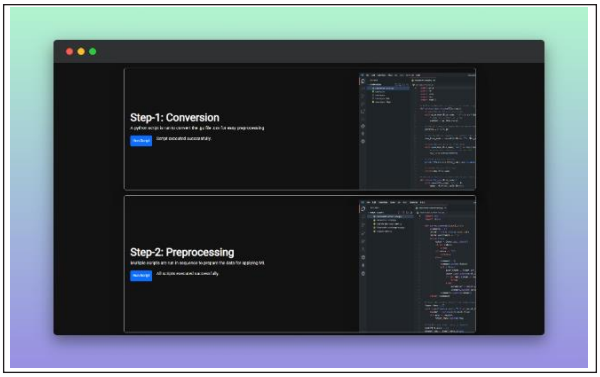


Fig. 21: Webapp Screenshot-2

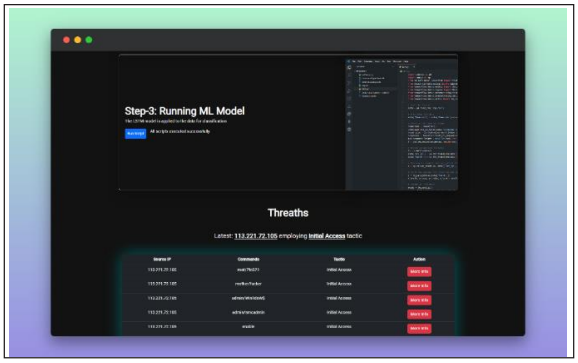


Fig. 22: Webapp Screenshot-3

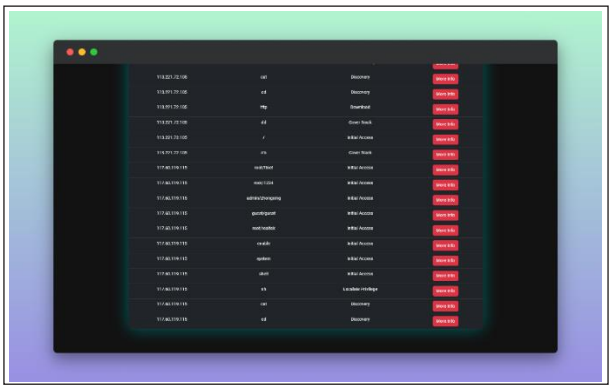


Fig. 23: Webapp Screenshot-3

Fig. 24: Webapp Screenshot-4

G. Testing

The system underwent rigorous testing to ensure its reliability. While the model showcased a high accuracy rate, certain challenges like device compatibility and incomplete links were noted. The system's feedback mechanisms, such as error and success messages, ensure users are always informed about the processing status.

V. Conclusions

A. Evaluation and Summary

The surge in cloud computing and remote work has escalated cyber-attacks, prompting researchers and organizations to explore AI and Machine Learning for solutions. These technologies can discern patterns in data and adapt with more data. The research highlighted the SSH attack approach, emphasizing the need for robust security measures. Data from Cowrie log files required feature engineering to be suitable for machine learning models. Instead of a binary classification of traffic, the focus shifted to analyzing attacker behaviour and predicting their tactics, inspired by the MITRE ATT&CK framework. LSTM was chosen for its ability to recognize patterns in sequences, achieving a 98% accuracy rate. The conclusion underscores AI's potential to enhance the efficiency of SOC analysts in identifying and responding to threats.

B. Limitations

The MVP has several limitations:

- Data processing is batch-based, not real-time.
- The model sometimes misclassifies non-malicious traffic.
- File downloads aren't analyzed for malware content.
- The dashboard displays only the first twenty records.

C. Recommendations and Future Plans

Generative AI, exemplified by ChatGPT and Dall-E, offers a new frontier in AI capabilities. Integrating an AI-driven assistant into the web application could expedite threat analysis and response. Further, extracting logs and analyzing uploaded files can provide deeper insights into malware threats. By

integrating Generative AI and advanced threat intelligence, a more comprehensive cybersecurity solution can be developed. This approach would enable organizations to proactively identify threats, reduce false positives, and effectively respond to potential attacks. The future of cybersecurity looks promising with the integration of advanced AI solutions.

References

[1] Check Point, "Live cyber threat map." Accessed: Oct. 26, 2022. [Online]. Available: https://threatmap.checkpoint.com/

[2] World Economic Forum, "The global risks report 2022." Accessed: Oct. 27, 2022. [Online]. Available: https://www3.weforum.org/docs/WEF_The_Global_Risks_Report_2022.pdf?_gl=1*15glpwh*_up*MQ..&gclid=CjwKCAjwh4ObBhAzEiwAHzZYU7nPT0hkC0Uz7NeEBsQMnS2a7ZWYTS7sVCFIe7wdzGfV3Mu1YkdZGhoCemEQAvD_BwE

[3] A. Chomiak-Orsa, A. Rot, and B. Blaicke, "Artificial intelligence in cybersecurity: The use of AI along the cyber kill chain," in *Computational Collective Intelligence*. Cham: Springer International Publishing, pp. 406-416, 2019, doi: 10.1007/978-3-030-28374-2_35.

[4] M. L. Kiah, and W. Zakaria, "A review of dynamic and intelligent honeypots," *ScienceAsia*, vol. 39s, pp. 1-5, 2013, doi: 10.2306/scienceasia1513-1874.2013.39S.001.

[5] Joyon, "Intrusion detection in self-driving cars using honeypots." Accessed: Dec. 18, 2022. [Online]. Available: https://digikogu.taltech.ee/et/Download/b9109b28-3225-4ed0-bf74-ba6dfaa0714c/Rndetuvastusisesitvatesautodesmeepottideabil.pdf

[6] Pauna, and I. Bica, "RASSH - Reinforced adaptive SSH honeypot," *IEEE International Conference on Communications* [Preprint], 2014, doi: 10.1109/ICCOMM.2014.6866707.

[7] S. Suratkar et al., "An adaptive honeypot using Q-Learning with severity analyser," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 10, pp. 4865-4876, 2021, doi: 10.1007/S12652-021-03229-2/METRICS.

[8] S. Lee et al., "Honeypot coupled machine learning model for botnet detection and classification in IoT smart factory – An investigation," MATEC web of conferences, vol. 335, p. 04003, 2021, doi: 10.1051/MATECCONF/202133504003.

[9] D. Zammit, "A machine learning based approach for intrusion prevention using honeypot interaction patterns as training data." Accessed: Dec. 20, 2022. [Online]. Available: https://www.um.edu.mt/library/oar/handle/123456789/13446

[10] J. T. Martínez Garre, M. Gil Pérez, and A. Ruiz-Martínez, "A novel machine learning-based approach for the detection of SSH botnet infection," *Future Generation Computer Systems,* vol. 115, pp. 387-396, 2021, doi: 10.1016/J.FUTURE.2020.09.004.

[11] P. Owezarski, "Unsupervised classification and characterisation of honeypot attacks*," Proceedings of the 10th International Conference on Network and Service Management (CNSM'14),* 2014, pp. 10-18, doi: 10.1109/CNSM.2014.7014136.

[12] Vectra, "Prevent cyberattacks with vectra AI." Accessed: Mar. 24, 2023. [Online]. Available: https://www.vectra.ai/

[13] Darktrace, "Darktrace AI: combining supervised and unsupervised machine learning | white paper | resources | darktrace." Accessed: Mar. 24 2023. [Online]. Available: https://darktrace.com/resources/darktrace-ai-combining-supervised-and-unsupervised-machine-learning

[14] Telekom Security, "Introduction into T-Pot: A multi-honeypot platform." Accessed: Dec.

29, 2022. [Online]. Available: https://github.security.telekom.com/2015/03/honeypot-tpot-concept.html

[15] AbuseIPDB, "Configuring Fail2Ban – AbuseIPDB APIv2 documentation." Accessed: Mar. 25, 2023. [Online]. Available: https://docs.abuseipdb.com/?python#configuring-fail2ban

[16] F. Sadique, and S. Sengupta, "Modeling and analyzing attacker behavior in IoT Botnet using temporal convolution network (TCN)," *Computers and Security*, vol. 117, 2021, doi: 10.1016/j.cose.2022.102714.

[17] Data Basecamp, "Long short-term memory networks (LSTM) - Simply explained! | Data Basecamp." Accessed: Mar. 25, 2023. [Online]. Available: https://databasecamp.de/en/ml/lstms