

Hybrid Clustering Mechanisms for High-Efficiency Intrusion Prevention

Pin-Shan Lin, Yi-Cheng Lai, Man-Ling Liao, Shih-Ping Chiu and Jiann-Liang Chen

*Department of Electrical Engineering, National Taiwan University of Science and Technology
Taipei, Taiwan*

m11107510@mail.ntust.edu.tw, m11007505@mail.ntust.edu.tw, m11007502@mail.ntust.edu.tw,
spchiu@mail.ntust.edu.tw, lchen@mail.ntust.edu.tw

Abstract— With the advancement of information and communication technology, cyberattack techniques have evolved into increasingly complex trends. Malicious network traffic attacks have become one of the information security problems for all organizations. This study is aimed to combat malicious network traffic attacks by actively collecting commands from attackers using honeypots. It involves pre-processing the raw network traffic data, employing a K-means algorithm to group the payloads, and label payloads using the MITRE ATT&CK framework. To improve the accuracy of the generated snort rules, the system utilizes Locality-Sensitive Hashing (LSH) method for secondary clustering, combined with snort rule generation, to form a comprehensive intrusion prevention system. In addition, to speed up the experimental process, this study adapted a script for this system to simulate an attacker's attack automatically. Through experimentation, it can be observed that hybrid clustering techniques such as K-means and LSH mechanisms can yield a defensive effectiveness of up to 93% for malicious payloads. This result proves the system's ability to identify and prevent different packet attacks effectively.

Keywords—K-means Algorithm, MITRE ATT&CK, Snort, Locality Sensitive Hashing (LSH), Malicious Packet

I. INTRODUCTION

As technology continues to advance, the popularity of the Internet continues to expand, profoundly affecting people around the globe. This trend has changed how people live their daily lives and reshaped the nature of business, information dissemination, and social interaction. This phenomenon has led most people to adopt the Internet as the primary medium for shopping, information retrieval, and social interaction. The Internet contains a wide variety of private and public information, making it a complex environment full of opportunities and challenges.

The widespread adoption of the global network has been accompanied by a startling surge in cyberattacks in recent years. The Internet has become a lucrative tool or medium for various illicit purposes. According to the latest data, in 2022, the annual growth rate of global cyberattack incidents reached 38% [1], with organizations facing an average of 1,168 attacks per week. These attacks come from hackers targeting financial institutions, government agencies, healthcare organizations, and military organizations. Furthermore, according to the IBM

Data Breach Report [2], the average cost of global data breaches has reached \$4.45 million by 2023, a 15% increase from three years ago. According to the Trend Micro 2023 Mid-Year Roundup Full Report, cybercriminal groups are continually developing new tools and techniques to enhance attack rates while reducing the chances of their attacks being detected. For instance, APT34, in its latest campaign, employed Command and Control (C&C) communication concealed within legitimate Simple Mail Transfer Protocol (SMTP) email traffic. Another group, Earth Longzhi, also utilized a novel technique known as stack rumbling in its attacks [3]. The reports mentioned above highlight the escalating diversity in attackers' tactics. To mitigate harm and prevent economic or data losses, government entities, and private enterprises must adopt innovative approaches to effectively detect and defend against evolving hacker attacks.

This study employs a hybrid approach that combines machine learning algorithms and hashing algorithms to conduct further analysis of network packets and generate firewall rules for detecting malicious packets and issuing system warnings. The primary objective of this method is to enhance the performance of the Intrusion Prevention System (IPS) so that it can more effectively respond to various types of network packet attacks and achieve better prevention capabilities. The techniques mentioned above will be described in detail in the next section.

II. RELATED WORK

This section is an introduction to the techniques used in this research. Related work can be categorized into three parts: literature on malicious packet detection, analysis of the application of LSH and associated methods, and an overview of snort and its usage.

The diversity of cyberattacks has made malicious network traffic detection a widely researched area. Inigo Perona et al. [4] combined payload-based and packet-header-based techniques for pre-processing raw packets. They analysed raw packets using standard compression algorithms like GZip, n-gram models, and the byte-frequency idea. Hongyu Liu et al. [5] proposed a novel payload classification method based on neural networks for detecting network attacks. This method utilizes deep learning models to analyse payloads and employs

PL-CNN and PL-RNN to automatically extract features without feature engineering, finally achieving an effective classification result. Bontupalli et al. [6] proposed memristor-crossbar-based architectures for high-speed packet classification and intrusion detection systems. This system combines software and hardware, enabling rapid and accurate detection of malicious packet attacks.

LSH is an algorithm that reduces the data query range while preserving data information. It is an important tool for fast similarity matching in various applications, such as searching for similar documents, images, and audio [7], [8]. Its applications include different fields, including image deduplication and recommendation systems. To make the snort rule more adaptable to different attack packets, this study uses LSH to calculate the similarity based on the payload content and performs secondary clustering.

Snort is an open-source Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) widely evaluated in the network security domain with excellent real-time traffic analysis and matching rules. Using snort makes it possible to effectively monitor abnormal behaviour, potential attack patterns, and the transmission of malicious packets in network traffic [9]. Fahmida Alam Rafa et al. [10] developed a Cloud-based Intrusion Detection System to deal with different cloud-based attack techniques, which can be applied in both traditional and virtual networks within a cloud environment.

This paper combines the above three techniques based on the system proposed in [11], [12]. It offers a malicious payload prevention system, which can detect the payloads with pre-written rules into the snort rules and prevent malicious packets without extracting the packet features.

III. PROPOSED METHOD

This study proposes a malicious packet-based detection and prevention system. This system uses a honeypot to collect threatening packet information and commands actively, further analyses them, and generates corresponding preventive measures and rules. As shown in Figure 1, there are five main steps: A. Pre-processing the data collected from the honeypot. B. Grouping similar payloads together into each cluster. C. Labeling payloads with MITRE ATT&CK. D. Calculating payload similarity and clustering using Locality-Sensitive Hashing. E. Generating snort rules through regular expressions.

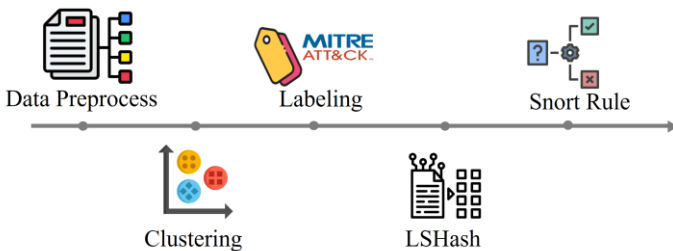


Figure 1. Proposed System Architecture

A. Pre-processing Data

This system uses the Cowrie Honeypot as a trap to lure potential attackers into the system. Cowrie is a highly interactive honeypot designed to capture and log sessions initiated by attackers using SSH and Telnet protocols. This tool is primarily used to record messages sent by attackers or actions related to attackers, enabling information security experts to study attack behaviours, attack patterns, and strengthen defensive measures. In this system, we pre-process the data once the honeypot receives packets sent by potential attackers. The initially received data is in JSON format, and to analyse these packet details more effectively, this study segments the packets based on timestamp and reorganizes them into a tabular dataset. Each packet in the table contains the following information: session ID, packet ID, packet payload, and timestamp. These actions simplify the original complex data structure and make the data content more organized and readable.

B. Payload Clustering

The data collected by Cowrie Honeypot is pre-processed to ensure optimal utilization of the data, including checking and cleaning of redundant data. Subsequently, to enhance data consistency and analyzability, payload data is transformed into vector form using natural language processing techniques such as the BERT encoder. To provide more helpful information for snort rule generation in the future and to help understand the distribution of malicious packets sent by attackers, the transformed payloads are then clustered using the K-means algorithm.

K-means, as an unsupervised learning algorithm, is utilized to identify data clusters with similar behaviours or distributions without labels. It is commonly applied in tasks like image compression and document classification. In the K-means algorithm, clustering is based on distance calculations between data points to determine cluster centroids. The distance between each point and the centroid is computed, as shown in Figure 2.

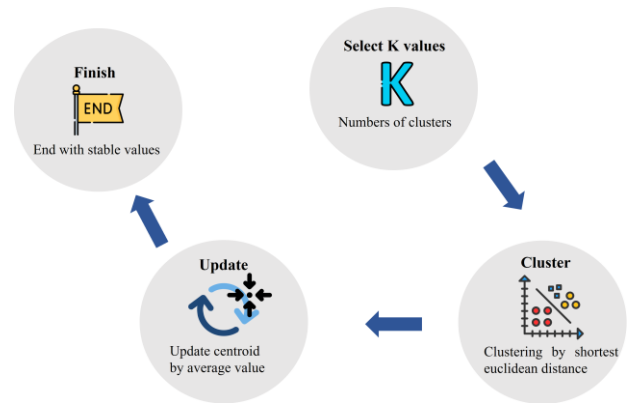


Figure 2. K-means Algorithm

C. Labeling with MITRE ATT&CK

MITRE ATT&CK is a concept of ATT&CK (Adversarial Tactics, Techniques & Common Knowledge) developed by MITRE, a non-profit organization, to help organizations

understand and analyse cyber attacks using a common language and framework. The core focus of the MITRE ATT&CK framework lies in the TTP components, which are tactics, techniques, and procedures. TTPs enable cybersecurity researchers to understand the attacker's tactics, allowing for further in-depth analysis of attack methods.

Following K-means clustering, data is labeled based on the payload content using MITRE ATT&CK's Tactic, Technique, and Subtechnique. Tactic reflects the adversary's objective in executing a particular action and is used for categorizing and understanding attack behaviour, as shown in Table 1. Technique describes the methods or techniques used by the attacker in achieving the tactical goal, including the related behaviours, the tools used, the impact of the attack, and other details, which are helpful to understanding the attacker's method of operation, as shown in Table 2. Subtechnique provides a more specific description of the methods attackers use when implementing a Technique. Each Technique may involve multiple Subtechniques, including different tools, attack approaches, or subtle variations, as illustrated in Table 3.

TABLE 1. THE CURRENT MITRE ATT&CK TACTIC

Tactic	Decription
TA0043 Reconnaissance	Attacker collect information about the target network or system to understand the potential target of the attack.
TA0042 Resource Development	The tools, vulnerabilities, and resources that an attacker intends to use in an attack, including developing malware or gathering effective attack resources.
TA0001 Initial Access	Attackers gain first access to a target system through a variety of approaches such as vulnerabilities, social engineering
TA0002 Execution	Attackers execute malicious code or commands on a target system in order to achieve his or her attack objectives

TABLE 2. SOME TECHNIQUES IN TACTIC: EXECUTION

Tactic: Execution(TA0002)	
Technique	Decription
T1053 Scheduled Task/Job	Adversaries use task scheduling for malicious code execution, requiring admin privileges.
T1204 User Execution	Adversaries use social engineering to trick users into executing malicious code via files or links.
T1059 Command and Scripting Interpreter	Adversaries may exploit command and script interpreters for executing commands, scripts, or binaries.
T1072 Software Deployment Tools	Adversaries utilize third-party software suites present within an enterprise network.

TABLE 3. SUBTECHNIQUES IN TECHNIQUE: COMMAND AND SCRIPTING INTERPRETER

Technique: Command and Scripting Interpreter (T1059)		
ID	Name	Decription
T1059.001	PowerShell	Adversaries exploit PowerShell for the purpose of execution.
T1059.002	AppleScript	Adversaries exploit AppleScript for the purpose of execution.
T1059.003	Windows Command Shell	Adversaries exploit Windows Command Shell for the purpose of execution.
T1059.004	Unix Shell	Adversaries exploit Unix Shell for the purpose of execution.
T1059.005	Visual Basic	Adversaries exploit Visual Basic for the purpose of execution.
T1059.006	Python	Adversaries exploit Python for the purpose of execution.
T1059.007	JavaScript	Adversaries exploit JavaScript for the purpose of execution.
T1059.008	Network Device CLI	Adversaries exploit Network Device CLI for execution.
T1059.009	Cloud API	Adversaries exploit Cloud API for the purpose of execution.

D. Locality-sensitive Hashing

LSH is one of the most common techniques in the Approximate Nearest Neighbor Algorithm (ANN). Its core concept revolves around applying a hash function to two neighboring data points in space, such that after hashing, these two data points remain close to each other in the area. LSH transforms data vectors into hash values while preserving information about their similarity. This approach reduces the search space, as illustrated in Figure 3. The algorithm mainly identifies similar data points from a large dataset. It finds practical applications in tasks such as text similarity detection and web search.

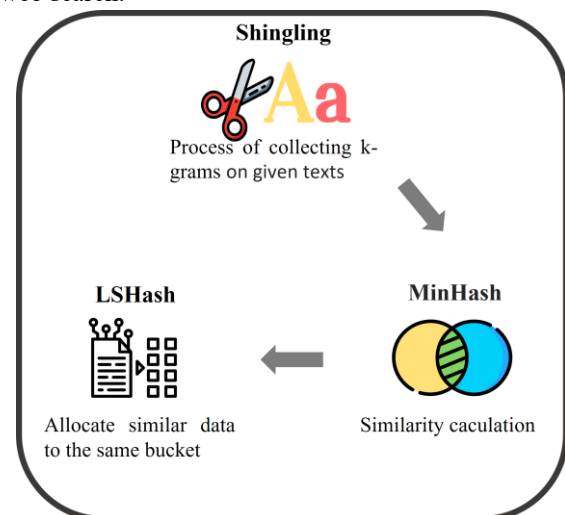


Figure 3. Locality-sensitive Hashing Algorithm Flow

In this phase, further processing is applied to the data that was previously clustered using K-means and labeled with

MITRE ATT&CK. LSH is utilized for clustering, grouping similar payloads into the same clusters to generate more clusters. These clusters will be used in the future to create snort rules, with each rule targeting the data from different clusters. The goal of this step is to precisely address specific attack patterns and behaviours within different clusters, thereby effectively generating more accurate snort rules. The typical processing involves the following steps:

1) Shingling: Shingling is a method based on k-grams, where k-grams are composed of a series of consecutive tokens. Specifically, these tokens can be words or symbols, depending on the context of the text. The ultimate goal of Shingling is to encode each document by utilizing the previously collected k-grams, thereby achieving a representation and analysis of textual data. This encoding process is a crucial step in data processing, as it allows for capturing patterns and features within documents, serving as the foundation for subsequent analysis and search tasks.

2) Min-Hashing: Due to the computational cost of calculating similarities between two one-hot encoded vectors, min-Hashing is used to convert one-hot encoded vectors into smaller-dimensional but denser vectors while retaining their similarity information. Min-Hashing is a hash function that rearranges the input vector and returns the index of the first value equal to 1. To obtain a denser vector of length n, referred to as a signature, n min-Hash functions are used to obtain n min-Hash values.

3) LSH: After applying Min-Hashing, we obtain signatures that retain similarity information. However, these dense signatures are still high-dimensional, and it is challenging to compare them efficiently. To address this issue, we use LSH to solve the above problem. LSH utilizes a signature matrix divided into b parts, each containing r rows known as subsignatures. Each part has r rows called subsignatures, which are then processed by a hash function. For instance, there are subsignature A and subsignature B, which need to compare their similarity. At this time, they are divided into b parts, and their respective b subsignatures are extracted and processed by the hash function. The final step involves determining whether the subsignatures of subsignature A and subsignature B are distributed into the same bucket based on hash values. Suppose the hash values of subsignature A and subsignature B indicate that they belong to the same bucket with a sufficient number of shared hash values. In that case, it can be concluded that subsignature A and sub-signature B are similar.

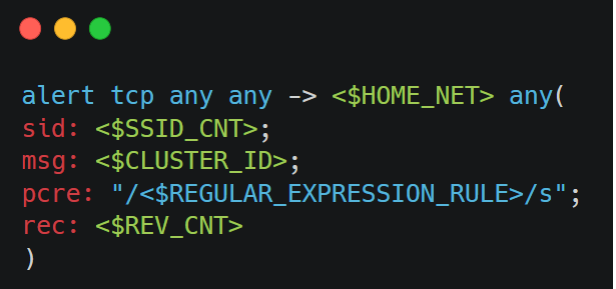
E. Snort Detection Rules

This study uses Snort as the primary tool for malicious packet detection and analysis. Depending on various objectives, rules for Snort can be crafted and optimized accordingly [14]. Through the previously mentioned steps, an analysis of payload similarity was conducted, resulting in a total of 42 clusters. Next, this system will utilize snort based on Perl Compatible Regular Expressions (PCRE) to provide a more structured approach to assist the well-clustered groups in

generating more standardized representations, as depicted in Figure 4. The meanings of each parameter are detailed in Table 4.

TABLE 4. DETAILS OF SNORT RULE

Snort Rule	Decription
sid	The sid keyword is used to be the representation of unique snort rule.
msg	The "msg" rule option specifies the message for log/alert, with packet dump or an alert.
pcre	The "pcre" allows user to design the Regular Expression to be the rules.
rev	The "rev" keyword identifies Snort rule's revision number uniquely.



```

alert tcp any any -> <$HOME_NET> any(
  sid: <$SSID_CNT>;
  msg: <$CLUSTER_ID>;
  pcre: "/<$REGULAR_EXPRESSION_RULE>/s";
  rec: <$REV_CNT>
)

```

Figure 4. Snort Rule with PCRE

This system employs an algorithm to generate snort rules based on a Regular Expression generator constructed using a Genetic Algorithm [15]. The concept of the Genetic Algorithm is to find a regular expression among a group of similar strings or texts that best matches the characteristics of all strings within that group. This optimal regular expression is then written into PCRE, enabling snort to detect and efficiently identify malicious behaviours associated with that cluster.

IV. EXPERIMENT

A. Test and Verify

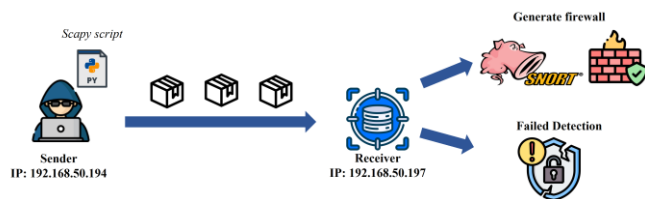
It is necessary to establish a corresponding testing environment and test the results with appropriate tools to validate whether the snort rule generator produces snort rules that can be detected effectively. The testing environment used here is shown in Table 5. The testing process involves two primary roles: the sender and receiver. The snort firewall is set up on the receiver side, with the Snort rules generated by the Genetic Algorithm being written into the configuration file. This configuration specifies the network segments and ports to be protected. Once the initial configuration is completed, testing can be started.

TABLE 5. TESTING ENVIRONMENT

Hardware	Use in this study
Virtual Machine	VMware Work Station
OS System	Kali-linux2023.3

The testing process for this research is illustrated in Figure 5. In this procedure, the sender transmits packets to the receiver, where the receiver uses the snort-generated firewall to inspect these packets. It takes appropriate actions based on preconfigured rules for the payload. The sender utilizes a Python script written with scapy to generate attack packets and send them to the receiver, simulating the behaviour of an attacker. To achieve process automation, the above process has been scripted, enabling the automated execution of packet sending. This script is designed to improve efficiency and reduce human intervention, make the experiment process more automated and increase repeatability, ensure the consistency and accuracy of the experiment, and reduce possible human errors.

The payloads used in the testing scripts of this study were collected by the Cowrie Honeypot in 2022. After data pre-processing, which involved removing duplicates and unusable data, 193 distinct payloads remained. Examples of payload content simulated during the attacks are shown in Table 6.

**Figure 5.** Testing and Verifying**TABLE 6.** SOME EXAMPLE OF TESTING PAYLOAD

Subtechnique	Payload
T1059.004	\$ nohup MipsLinuxTF &
T1059.004	echo "/.syntem&">>/etc/rc.local
T1120.000	cat /proc/cpuinfo grep name wc -l

B. Performance Analysis

In this work, the honeypot packets were collected from various Internet Service Providers (ISPs). The honeypot-generated JSON files were processed by data pre-processing, K-means, and labeled with MITRE ATT&CK, LSH. Based on the similarity of payload contents, two clustering methods were employed to group similar payloads to generate more compliant snort rules. The distribution of clusters is presented in Table 7. It is evident from the table that the sixth cluster represents the largest portion of attack types collected during this period, comprising 63.45% of all attack clusters. Its Subtechnique corresponds to MITRE ATT&CK framework T1120.000: Peripheral Device Discovery, where attackers attempt to gather information about physical devices.

In this study, 193 different packets were used for experimentation, both before and after applying LSH. The

main purpose is to evaluate whether the secondary clustering of snort rules improved the defensive rate of malicious packets. The experimental results are presented in Table 8. Following the experiments mentioned above, it can be found that the defensive rate of malicious packets increased from 91% to 93%, which shows that adding LSH as one of the clustering techniques improves the detection effect of snort abnormal packets.

TABLE 7. DETAILS OF SNORT RULE

Cluster ID	Packet number	Percentage(%)
6	24454	63.45919
19	4192	10.87842
18	3438	8.92176
30	1663	4.31556
24	1143	2.96613
37	928	2.40820
35	735	1.90736
3	579	1.50253
23	241	0.62541
25	237	0.61503
26	214	0.55534
20	128	0.33217
38	113	0.29324
0	78	0.20241
27	69	0.17906
28	62	0.16089
29	61	0.15830
1	32	0.08304
2	24	0.06228
11	20	0.05190
4	20	0.05190
5	18	0.04671
15	17	0.04412
17	12	0.03114
14	11	0.02855
33	9	0.02336
8	8	0.02076
31	4	0.01038
10	4	0.01038
41	3	0.00779
32	2	0.00519
16	2	0.00519
34	2	0.00519
12	2	0.00519
36	2	0.00519
22	2	0.00519
21	1	0.00260
13	1	0.00260
9	1	0.00260
7	1	0.00260
39	1	0.00260
40	1	0.00260

TABLE 8. RESULT ANALYSIS

	K-means	K-means + LSH
Numbers of cluster	27	42
Dst IP: port	192.168.50.197:100	192.168.50.197:80
Time Execution(s)	13.991	14.069
#Success Packet Detection	176	180
Packet Detection Rate	91%	93%

V. CONCLUSION

This study proposes a real-time system to prevent malicious payloads collected by honeypots in network traffic. While prior research primarily focused on analyzing packet headers to detect malicious packets, this study demonstrates the effectiveness of payload analysis for malicious packets. After pre-processing the raw packets, similar packets are clustered based on payload similarity using the K-means algorithm. MITRE ATT&CK labels are applied to evaluate the results of the initial clustering. To enhance snort firewall's defensive rate of packets, the system adopts a secondary clustering technique. The LSH method is utilized to re-aggregate clusters with higher similarity from the previous step, enabling snort to generate more effective snort rules for different payloads. The system, as mentioned, undergoes experiments simulating attacks where packets are continuously sent by the attacker. The defensive rate of packets reaches 93%. This study makes a valuable contribution to preventing malicious packets. Secondary clustering and payload analysis are feasible research directions for detecting malicious packets. Future research could explore alternative clustering methods, such as fuzzy string analysis, to improve the detection rate of malicious packets.

REFERENCES

- [1] Check Point Research Team, Check Point Research Reports a 38% Increase in 2022 Global Cyberattacks, [Online]. Available: <https://blog.checkpoint.com/2023/01/05/38-increase-in-2022-global-cyberattacks/>, 2023.
- [2] IBM, Cost of a Data Breach Report 2023, [Online]. Available: <https://www.ibm.com/reports/data-breach>, 2023.
- [3] Trend Micro, Preventing Risk - Trend Micro 2023 Cybersecurity Report, [Online]. Available: https://www.trendmicro.com/zh_tw/security-intelligence/threat-report/2023-midyear-security-roundup.html, 2023.
- [4] I. Perona, I. Gurrutxaga, O. Arbelaitz, J.I. Martin, J. Muguerza, and J.M. Perez, "Service-independent payload analysis to improve intrusion detection in network traffic," *Proceedings of the 7th Australasian Data Mining Conference*, vol. 87, pp. 171–178, 2008.
- [5] H. Liu, B. Lang, M. Liu, and H. Yan, "CNN and RNN based payload classification methods for attack detection," *Knowledge-Based Systems*, vol. 163, pp. 332–341, 2019.
- [6] V. Bontupalli, C. Yakopcic, R. Hasan, and T. M. Taha, "Efficient memristor-based architecture for intrusion detection and high-speed packet classification," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 14, no. 4, pp. 1–27, Dec. 2018.

- [7] K. Ozdem and M. A. Akcayol, "Locality Sensitive Hashing Based Clustering for Large Scale Documents," *Proceedings of 6th International Conference on Mathematics and Artificial Intelligence*, pp. 137–142, 2021.
- [8] A. Bahri, K. E. Moutaoikil, and I. Badi, "A combined CNN and LSH for fast plant species classification," *Proceedings of the 4th International Conference on Big Data and Internet of Things*, pp. 1–6, 2019.
- [9] M. Roesch, "Snort: Lightweight intrusion detection for networks," *In Lisa*, vol. 99, no. 1, pp. 229–238, 1999.
- [10] F. Rafa, Z. Rahman, M. M. Mishu, M. Hasan, R. Rahman and D. Nandi, "Detecting Intrusion in Cloud using Snort: An Application towards Cyber-Security," *Proceedings of the 2nd International Conference on Computing Advancements*, pp. 199–206, 2022.
- [11] Y. C. Lai, C. L. Yu, M. L. Liao, Y. S. Lin, Y. C. Chang, and J. L. Chen, "An Intelligence Defense System with SNORT Rules," *Proceedings of the 25th International Conference on Advanced Communication Technology*, pp. 249–254, 2023.
- [12] M. L. Liao, C. L. Yu, Y. C. Lai, S. P. Chiu, and J. L. Chen, "An Intelligent Cyber Threat Classification System," *Proceedings of 25th International Conference on Advanced Communication Technology*, pp. 189–194, 2023.
- [13] S. Abdulrezzak and F. A. Sabir, "Enhancing Intrusion Prevention in Snort System," *Proceedings of the 15th International Conference on Developments in eSystems Engineering*, pp. 88–93, 2023.
- [14] P.J. Chuang, P.C. Chao, H.C. Lu, "Regular Expression Generator by using Genetic Algorithm," Available: <https://github.com/maojui/Regex-Generator>, 2020.



Pin-Shan Lin was born in Hsinchu, Taiwan, in 2000. She received her B.S. degree in 2022. She is pursuing an M.S. degree in electrical engineering from the National Taiwan University of Science and Technology, Taipei. Her main research interests include data analysis, machine learning, and cyber threat intelligence.



Yi-Cheng Lai was born in Taipei, Taiwan, in 1998. He received his B.S. degree in 2021. He is pursuing an M.S. degree in electrical engineering from the National Taiwan University of Science and Technology, Taipei. His main research interests include cyber security, vulnerability research, and defense.



Man-Ling Liao was born in Tainan, Taiwan, in 1998. She received her B.S. degree in 2021. She is pursuing an M.S. degree in electrical engineering from the National Taiwan University of Science and Technology, Taipei. Her main research interests include data analysis, machine learning, and cyber threat intelligence.



Shih-Ping Chiu was born in Taipei, Taiwan, in 1984. She received the B.S. degree in 2007. She is a research assistant in electrical engineering from the National Taiwan University of Science and Technology, Taipei. Her main research interests include data analysis and the Internet of Things (IoT).



JIANN-LIANG CHEN (Senior Member, IEEE) was born in Taiwan in December 1963. He received a Ph.D. in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1989. Since August 1997, he has been with the Department of Computer Science and Information Engineering, National Dong Hwa University, where he is a Professor and the Vice Dean of the Science and Engineering College. He joins the Department of Electrical Engineering, National Taiwan University of Science and Technology, as a Distinguished Professor. His current research interests include cellular mobility management, cyber security, personal communication systems, and the Internet of Things (IoT).