Threat Detection based on System Credibility by Logging Analysis and Visualization

Wei Qiao
Purple Mountain Laboratories
Nanjing, China
qiaowei@pmlabs.com.cn

Youjun Bu
Information Engineering University
Zhengzhou, China
buyoujun@pmlabs.com.cn

Yao Chen
Purple Mountain Laboratories
Nanjing, China
chenyao@pmlabs.com.cn

Xiaoxiao Jiang
Purple Mountain Laboratories
Nanjing, China
jiangxiaoxiao@pmlabs.com.cn

Bingbing Jiang
Purple Mountain Laboratories
Nanjing, China
jiangbingbing@pmlabs.com.cn

***Abstract*—The novel theory of Endogenous Safety and Security was proposed from a system architecture perspective is striving to address the current complex cybersecurity threats dilemma. It utilizes multiple heterogeneous and functionally equivalent systems (called mimic systems) to detect threats because different implementations have different vulnerabilities and are dynamically scheduled on some feedback strategies, making it impossible for a single attack to simultaneously compromise all of these implementations. The threat detection heavily relies on the adjudication of outputs from multiple heterogeneous and functionally equivalent systems because it is possible to adjust the outcomes of the majority of compromised systems as correct. Therefore, the credibility of the adjudication should be evaluated for verifying the trustworthiness of the mimic system, but no research is currently available on the system credibility in the mimicry environment. In this paper, we propose a logging analysis algorithms to evaluate the credibility of the adjudication which is related to each single system's disturbance event history, disturbance factors, disturbance number and one-time runtime duration. The experiments also prove the positive performance of the proposed algorithm. The lower the credibility, the higher the possibility of the system being compromised.**

***Index Terms*—cybersecurity, threat detection, credibility.**

I. Introduction

To address the current complex cybersecurity threats dilemma, several innovative approaches have been proposed, including Moving Target Defense (MTD) [1], Zero Trust Architecture [2], Cyber Resilience [3], and Endogenous Safety and Security [4] [5]. Unlike the latter two techniques, which are still in the early stages of development, the former techniques offer systematic implementation methods. However, when it comes to MTD, there is a concern regarding potential leaks due to redundancy considerations. With MTD, only one target is active at any given time, regardless of how the targets move. This creates non-negligible opportunities for successful attacks. To address this issue, the technique of Endogenous Safety and Security encompasses dynamic, varied, and redundant properties. Building upon this concept, J. Wu proposed the Dynamic, Heterogeneous and Redundant (DHR) architecture [4] [5], which is capable of defending against "unknown unknown" threats. Concrete implementations of this approach, such as mimic routers, mimic web servers, and mimic cloud systems [16], have been developed.

In simpler terms, the principle of DHR relies on the idea that it is extremely difficult for a single attack to penetrate multiple implementations that have different functionalities but are equivalent in performance. This is because vulnerabilities in software or hardware from different manufacturers, or in different operating systems like Windows, Linux, or macOS, are often unique to each of them. Leveraging this fact, DHR's collective awareness surpasses the limited perception of individual components, making it more resilient against (unknown) threats. In the current approach, multiple components with equivalent functionality process the same input, and the verdict module compares their outputs based on predefined rules to determine the final result. However, in this process, all the components are treated equally, regardless of their vulnerability. In reality, the more vulnerable a component is, the less trustworthy it should be. Therefore, it is important to evaluate the credibility of each component individually rather than giving them equal trust in the final verdict.

Log analysis proves to be a valuable tool in addressing the aforementioned concerns. In simple terms, it helps in determining the credibility of each component by analyzing the logs generated by the mimic devices and assessing the potential threat perturbations they have experienced. The idea is that, if an executor suffers a larger number of threat perturbations, which are reflected in the logging system, it becomes less trustworthy. Furthermore, by sharing and analyzing threat log information among all the mimic devices, the entire mimic network can achieve a collective threat awareness. This means that the swarm awareness of an individual mimic device can be extended to the entire network through log analysis. By leveraging this approach, the network can effectively detect and respond to potential threats based on the insights gained from the analysis of the logs. Therefore, we assess the credibility of the component participating in the adjudication process by evaluating its history of being attacked, the reasons and

 ISBN: 978-1-68558-106-0

frequency of the attacks, and whether its runtime duration is normal. Then, by combining the credibility of each component with the adjudication strategy, we can calculate the trustworthiness of the final output result and determine whether cyber threats are detected.

The contributions of this paper can be summarized as follows:

- Log analysis for credibility assessment: This paper suggests leveraging log analysis as a powerful tool for assessing the credibility of individual executors within the DHR architecture. By analyzing the threat log information recorded by mimic devices, we establish a correlation between the suffered threat perturbations and the credibility of the executors. This contributes to the overall security and reliability of the system.
- Mitigation of unknown threats: This paper highlights the capability of the DHR architecture, especially when combined with log analysis, to defend against "unknown unknown" threats. By integrating dynamic, varied, and redundant properties within the architecture and utilizing log analysis for credibility assessment, the system becomes more resilient and capable of addressing unforeseen or evolving cyber threats.
- Practical implementations: The experiments for the DHR architecture, such as mimic routers, mimic web servers, and mimic cloud systems demonstrate the feasibility and effectiveness of the proposed approach in enhancing cybersecurity and mitigating threats in various domains.

The rest of the paper is structured as follows. The related work is presented in Section 2. We present the tool of log system of mimic devices for collecting, processing and analyzing threat data In Section 3. The system credibility evaluation algorithm is introduced in Section 4. The experiment is presented in Section 5. Finally, we conclude our work in Section 4.

II. Related Work

Security Information and Event Management (SIEM) is an advanced technology that combines Security Information Management (SIM) and Security Event Management (SEM). It offers real-time monitoring and analysis of events, as well as the tracking and recording of security data for compliance or audit purposes. It encompasses various functions such as log management, event correlation and analysis, event monitoring and security alerts, network visualization, and threat intelligence. Although there are threat log analysis tools like Splunk [6], Elastic stack [7], and Azure Sentinel [9] based on SIEM, they do not incorporate the specific characteristics of mimicry systems, thus lacking the ability to reflect the swarm awareness of threat from the log analysis level. Furthermore, existing research [10]–[14] on threat awareness or hunting through log analysis does not consider mimic systems. A proposed framework [15] for threat analysis in a heterogeneous log environment focuses on different types of information systems, excluding both mimic and heterogeneous systems, making it unsuitable for such environments.

III. Log System of Mimic Devices

Based on the standard for mimic logs, we have designed and developed a cloud-based management system specifically for mimic logs. This system provides visual management and analysis capabilities for mimic logs, allowing for the detection and warning of potential threats. Security personnel can then take prompt and effective defensive actions, such as implementing patches, upgrades, and executing cleanup procedures, to counter cyberattacks.

The mimic log cloud management system effectively reduces service interruptions caused by differential mode disturbances and minimizes the likelihood of common-mode escape. Additionally, it supports accurate and efficient scheduling and decision-making based on data. The system offers several key functionalities related to log data, including centralized collection, unified preprocessing, normalized parsing, taxonomy indexing, centralized storage, real-time querying, multidimensional analysis, and visualization. These features enable comprehensive management and analysis of log data, empowering security personnel to identify and respond to threats in a timely and efficient manner. Here, we mainly describe what capabilities the log system should have, as well as the implementation methods or tools of various capabilities, without involving specific implementation details.

- **Collection**: Our system is designed to simplify the distribution and deployment process of various log collectors, such as filebeat and metricbeat, across numerous mimic devices. It empowers the distributed collection of log data while maintaining centralized reception and supports an array of log collectors, accommodating diverse use cases. Once deployed, it facilitates the collection of log data from these distributed devices. Through centralized reception, the log data from a single location can be conveniently accessed and analyzed. This centralized approach enhances efficiency and provides a comprehensive overlook of the log data generated by the mimic devices. Our system can streamline the distribution and deployment of log collectors, enabling efficient distributed log collection and centralized log data reception.
- **Preprocessing**: This program is designed to handle log data transmitted to the system with a wide array of operations. It offers features such as field filtering, allowing users to extract specific fields from the log data based on their needs. The program also supports field format conversion to transform the format of fields within the log data, ensuring compatibility and consistency. Additionally, it provides functionality for field duplicate removal, eliminating redundant entries and improving data integrity. With these capabilities, users can efficiently manipulate log data to derive meaningful insights and optimize analysis processes.
- **Normalized Parsing**: By adhering to the standard mimic log format, this program simplifies the development of log data parsing programs. It achieves this by implementing a unified field encoding format, ensuring normalized

 ISBN: 978-1-68558-106-0

log data parsing. As a result, developers can save time and effort in handling various log formats and focus on core functionalities. The use of a consistent field encoding format allows for seamless integration with existing parsing tools and libraries. This streamlined approach greatly enhances the efficiency of log data parsing, resulting in faster and more accurate analysis of log data. With this program, developers can optimize their parsing workflows and maximize the value extracted from log data.

- **Categorical Index**: With this program, users have the capability to generate both typal and statistical log-data indexes tailored to their unique requirements. These indexes serve as powerful tools for querying and analyzing log data. Users can create typal indexes to categorize log data based on predefined patterns or structures, enabling efficient searching and filtering. Additionally, statistical indexes enable users to extract meaningful insights by aggregating and analyzing log data based on statistical measures such as counts, averages, and trends. Overall, these customizable indexes empower users to unlock the full potential of their log data for query and analysis purposes.
- **Centralized Repository**: After performing the three steps mentioned above, the log data from diverse mimic devices is centralized into a central storage system. This centralized storage ensures that log data is securely stored and easily accessible for query and analysis purposes. With a centralized storage system, users can conveniently retrieve and analyze log data without the need to navigate through multiple sources or locations. Moreover, this centralized storage system provides reliable data backup capabilities.
- **Real-time retrieval**: The system is equipped to collect real-time logs from a wide range of mimic devices. It can gather logs related to verdicts, schedules, and performance metrics from devices such as routers, switches, WEB, Unified Data Management (UDM), Home Subscriber Server (HSS), and Advanced Driver Assistance Systems (ADAS). By collecting logs from these diverse devices, the system offers a comprehensive view of the network and its components. This allows for a holistic understanding of system performance and operational status. Furthermore, the accumulated logs can be promptly queried. Users can easily access and retrieve specific logs based on their requirements. This real-time log querying capability enables users to monitor system events and performance metrics, aiding in troubleshooting, performance optimization, and overall network management.
- **Comprehensive Analysis**: In addition to statistical analyses, this system offers correlation analysis of log data to address challenges in analyzing mimic component or device logs. Correlation analysis identifies meaningful relationships and patterns, revealing hidden insights not apparent through individual statistical analyses. These correlations provide a deeper understanding of interactions among log events or variables, facilitating comprehensive data analysis. By leveraging deep learning technology, the system efficiently processes and analyzes large log data volumes, extracting complex patterns and empowering users to make informed decisions.
- **Visualization**: This system provides powerful visualization capabilities for mimic logs, allowing users to gain insights and understand the data more intuitively via Kibana [8]. Users can visualize various types of information, including raw logs, current system status, analysis results, numerical distributions, and trend predictions. By representing log data visually, users can easily identify patterns, anomalies, and trends, facilitating quicker and more effective analysis of the system's behavior and performance. These visualizations offer a comprehensive overview of the system's performance and help users identify potential bottlenecks, optimize resource allocation, and make data-driven decisions for improving the overall functionality and efficiency of the mimic devices.

IV. Log Analysis for Credibility

A. Factors Influencing Credibility

In order to enhance the resilience of the DHR architecture against "unknown" threats in real-world engineering, it is crucial to assess the credibility of executors. Executors (referring to heterogeneous and functionally equivalent systems) play a significant role in determining the credibility of the final verdict results, which indicate whether mimic devices recognize and respond to threats. Thus, measuring the credibility of executors becomes essential.

To measure the credibility of executors, the first step is to identify the factors that influence the reliability of their outputs. Based on the mimic defense theory and benchmark function experiments conducted by Wu et al. [16], we have identified the following four key factors related to executors. Other influences can affect the trustworthy of executors' outputs, but the experiments show that the four factors below characterize the executors' credibility.

1) **Disturbance Event History**: If an executor has previously experienced disturbances, it indicates a potential security risk, and therefore, its credibility should be diminished. This is because the ability of an executor to reliably produce accurate results may be compromised due to the past disturbances. Consequently, it is necessary to reduce the credibility of such an executor in order to maintain the overall trustworthiness of the system. Furthermore, it is important to analyze the specific characteristics and patterns of past disturbances encountered by the executor. By studying the nature and extent of these disturbances, one can gain insight into the potential vulnerabilities or weaknesses of the executor, thereby providing strategies to mitigate the associated security risks. This additional analysis helps in enhancing the overall security and reliability of the system by effectively addressing the identified risks.
2) **Disturbance Factors**: When an executor experiences the same disturbed cause as a previous instance, it implies

that the executor possesses a recognized weakness that has been reused. As a result, their credibility should be diminished. This emphasizes the importance of identifying patterns in an executor's behavior and taking necessary precautions to mitigate potential risks associated with their known weaknesses. By acknowledging and addressing these weaknesses, trust in the executor can be maintained or restored.

3) **Disturbance Number**: If an executor has a disturbed times greater than the average of all executors' disturbed times, it indicates that the executor is more susceptible to attacks. Consequently, their credibility should be diminished. This suggests the need for additional scrutiny and security measures to protect the executor and prevent any potential breaches. It is crucial to identify the reasons behind the increased vulnerability and take appropriate actions to strengthen the executor's defenses. By reducing their credibility, it serves as a cautionary measure to ensure that the executor's actions are carefully monitored and their weaknesses are addressed promptly.
4) **One-time Service Runtime Duration**: If an executor runs for an extended period of time, surpassing the predefined threshold for a standard runtime, it is deemed to be potentially disrupted or compromised, and, as a result, its credibility should be diminished. Additional measures may need to be taken to investigate the root causes of the prolonged runtime and ensure the reliability and security of the executor.

B. Credibility of Executors

Based on the previous analysis, the credibility measurement algorithm for executors is designed as follows:

- **Step 1:** Set E as the current evaluated executor, DEH_list as the executors' disturbance event history, DF_map and DN_map as two maps storing executors and their disturbance factors and number respectively, E_cred as the credibility of the executor and the initial value 1.
- **Step 2:** Query the list and determine whether E is in DEH_list. If $DEH\_list.Query(E)$, $E\_cred = E\_cred - \alpha_h$; ($\alpha_h$ is the weight value of the historical disturbed factor).
- **Step 3:** Query the map DF_map and determine whether E has had the current disturbed factor before. If $DF\_map.find(E).Query(dc)$ (dc is the current disturbance factor), $E\_cred = E\_cred - \alpha_f$($\alpha_f$ is the weight value of the disturbed cause factor).
- **Step 4:** Query the map DN_map and calculate E's disturbed frequency and all executors' average value. Set $en = DN\_map.find(E)$, $an = DN\_map.avg()$; If $en > an$, $E\_cred = E\_cred - \alpha_n$ (α_n is the weight value of the disturbance number).
- **Step 5:** Calculate E's running time t at this service. If $t > T$ (T is the threshold of the runtime of one normal service), $E\_cred = E\_cred - \alpha_t$ (α_t is the weight value of the runtime factor).
- **Step 6:** Output the executor E's credibility E_cred.

This method involves the collection and documentation of significant indicators that influence the credibility of executors. Through a combination of experience and iterative experimental testing, the algorithm determines the appropriate weights for each of these factors. By assigning weights to the different indicators, the algorithm can effectively measure the relative importance of each factor in determining the overall credibility of an executor. This enables a comprehensive and systematic evaluation of an executor's trustworthiness. The algorithm dynamically calculates the credibility of each executor based on the weighted factors and their corresponding values. This dynamic output reflects the evolving nature of an executor's credibility, as it can be influenced by changes in performance, client feedback, or other relevant factors.

C. Credibility of System

Several methods can be used to measure the credibility of the verdict results based on the credibility measurement of each online executor. One common approach is to calculate the average credibility of the executors, either simply or conditionally. In this paper, we propose a method that takes into account the current verdict information. Typically, in the majority verdicts, the opinions of the minority do not significantly impact the final result. However, to ensure accuracy, adjustments can be made based on the mean confidence of the online executors.

To implement this method, the average credibility of all executors can be calculated, and then adjusted based on the level of confidence expressed by the majority. By weighing the credibility scores of the executors with their corresponding confidence levels, a more refined measurement of credibility can be obtained. This approach allows for a more nuanced assessment of the credibility of verdict results, taking into consideration both the collective opinion of the executors and the level of confidence they exhibit. It enhances the accuracy and reliability of the final verdict by appropriately weighting the influence of each online executor. Assume the number of online executors is $m = 2k + 1$.

- If more than k executors satisfy one of the above four factors, the confidence of the verdict result should be reduced to that of all satisfying the factor.
- If it is less than k, the confidence should be increased to that of none satisfying the factor.
- If exactly k, we also check if these k executors are the current abnormal executors.

The calculation formula is as follows:

$$V\_cred = \frac{1}{m} \sum_{E \in VLR.set} E\_cred + \sum_{i \in \{h,f,n,t\}} \Delta_i(VLR)$$

$$\Delta_i(VLR) = \begin{cases} \frac{Count(VLR,i)}{m} \times \alpha_i & , if\ j < k; \\ \frac{1}{2m} \times Norm(VLR,i) \times \alpha_i & , if\ j = k \\ -\frac{m-j}{m} \times \alpha_i & , otherwise. \end{cases}$$

$$Count(VLR,i) = \sum_{E \in VLR.set} isFactor(E,i)$$

$$Norm(VLR, i) = \begin{cases} 1 & , if\ VLR.abnorm \notin VLR.set(i); \\ 0 & , otherwise. \end{cases}$$

$$isFactor(E, i) = \begin{cases} 1 & , if\ E\ has\ the\ factor\ i; \\ 0 & , otherwise. \end{cases}$$

$$i \in \{h, f, n, t\}.$$

where VLR is the current verdict log record, $VLR.set$ is the set of online executors contained in VLR, $VLR.set(i)$ is the set of online executors in VLR satisfying the factor i, $VLR.abnorm$ is the abnormal executor in VLR, V_cred is the credibility of the verdict result, E_cred is the credibility of the online executor E, $\{h, f, n, t\}$ represents the above four factors, and Δ_i stands for the tuning parameter of the corresponding factor.

V. Experiments

In order to validate the threat awareness of mimic devices and mimic networks, an experiment was conducted in the Network Endogenous Security Testbed (NEST) environment. The experiment utilized a hardware setup consisting of an Intel(R) Xeon(R) Silver 4214R CPU with a clock speed of 2.40GHz and 12 processor cores. The software environment employed was CentOS Linux version 7.4.1708 (Core). To facilitate the experiment, several applications were utilized: Filebeat 8.3.3 (Linux-x86_64), Kafka 2.0, Logstash 8.1.0, Elasticsearch 8.1.0, and Kibana 8.1.0. The threshold parameters in our algorithm were all settled as 0.1 (this number is arrived through the parameter debugging according to expert advice) in the following experiment. Within this experimental setup, the mimic devices and mimic networks were subjected to various scenarios and situations to assess their threat awareness. The aim was to evaluate how well these entities could detect and respond to potential threats in a realistic and controlled environment. By conducting the experiment in the NEST environment and employing the aforementioned hardware and software components, the study aimed to gain insights into the effectiveness of the mimic devices and networks in identifying and mitigating potential security risks. The results obtained from the experiment will contribute to further enhancing the security measures and threat awareness capabilities of these systems.

By utilizing the mimic log system and implementing the proposed algorithm, the credibility scores of the verdict results are computed. This allows for the determination of the success of disturbances in the system. The obtained results are then compared with real-world data, enabling an assessment of the accuracy and precision of our algorithm. The mimic log system aids in generating credibility scores for the verdicts produced. These scores serve as an indication of the reliability and trustworthiness of the results. Based on this information, it becomes possible to determine whether any disturbances or attacks have successfully affected the system. To evaluate the performance of our model, a comparison is made between the results obtained from the threat awareness model and the actual observed data. This analysis allows us to assess the accuracy and precision of our model in detecting and responding to potential threats. By considering the consistency and alignment between the model's predictions and the real-world outcomes, we can gauge the effectiveness of our threat awareness system. Ultimately, this evaluation process provides valuable insights into the capabilities and limitations of our model.

- **Accuracy**: The proportion of correct forecast quantities to total quantities in both positive and negative cases.

$$Accuracy = \frac{TP + FN}{TP + FP + TN + FN}$$

- **Precision**: Percentage of correct prediction within the sample with positive prediction.

$$Precision = \frac{TP}{TP + FP}$$

- TP (True Positives): The positive result predicted by the model is consistent with the actual result of the disturbance suffered by the mimic system.
- FP (False Positives): The positive result predicted by the model is the opposite of the actual result of the normal operation of the mimic system.
- FN (False Negatives): The negative result predicted by the model is consistent with the actual result of the normal operation of the mimic system.
- TN (True Negatives): The negative result predicted by the model is the opposite of the actual result of the disturbance suffered by the mimic system.

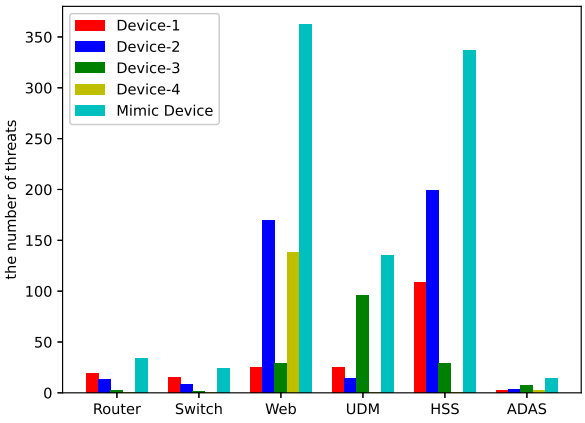

Fig. 1. The number of threats detected by a single-device vs. mimic device.

Figure 1 shows that the greater the number of heterogeneous and functionally equivalent devices, the greater the reliability and confidence in the verdict results, as well as enhanced threat detection capabilities, and Figure 2 illustrates that the threat detection capability of a combination of different types of mimic devices is stronger than that of a single mimic device. Figure 3 presents a comparison between the results of our threat detection algorithm in the experiment and the actual results.

 ISBN: 978-1-68558-106-0

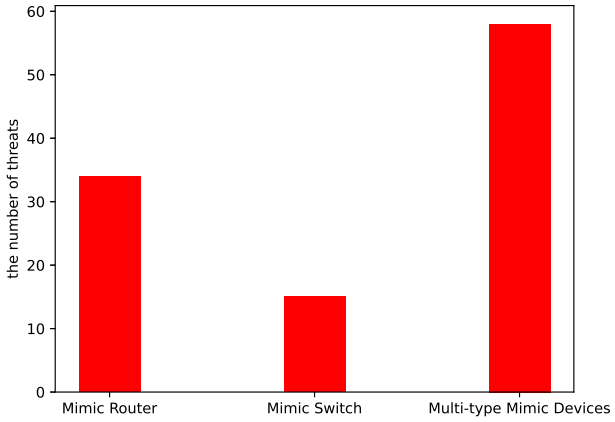

Fig. 2. The number of threats detected by a mimic device vs. multi-type mimic devices.

$$Accuracy = \frac{179 + 93}{179 + 71 + 11 + 93} = 0.768$$

$$Precision = \frac{93}{11 + 93} = 0.894$$

In the experiment, we set the scoring threshold as 0.85. That is, a score of less than or equal to 0.85 indicates a positive case prediction, whereas a score greater than or equal to 0.85 indicates a negative case prediction. We can calculate our model's accuracy as 0.768 and precision as 0.894 which shows it can effectively perceive threats.

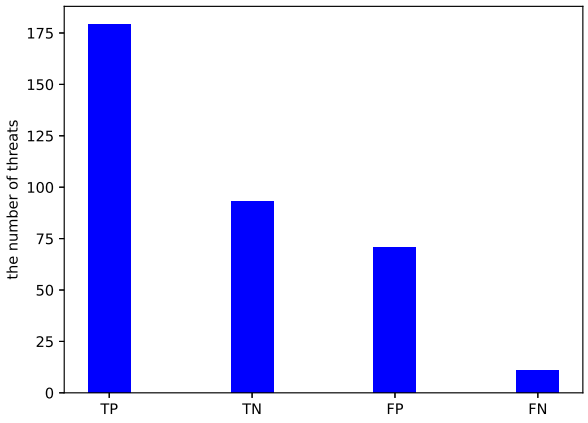

Fig. 3. The result of our threat detection algorithm.

VI. Conclusion

In this paper, a threat detection model based on system credibility is proposed to enhance the network defense capabilities of the entire mimic network. It is specifically designed to handle the log processing and association analysis tasks of mimic devices and serves as a centralized platform for collecting, analyzing, and correlating logs from multiple mimic devices. It also enhances the overall situational awareness by providing a comprehensive view of the network defense status and facilitating prompt detection and response to potential threats. To evaluate the effectiveness of the threat detection model, a verification experiment is conducted in the NEST cyber range. The experimental results demonstrate the capability of the model in effectively perceiving and identifying threat events within the mimic defense context.

Acknowledgment

B. Jiang is supported in part by Jiangsu Province "Shuangchuang" Plan under Grant JSSCBC20221657. Y. Bu is supported in part by NSFC under Grant 62176264.

References

[1] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, "Moving Target Defense - Creating Asymmetric Uncertainty for Cyber Threats," Advances in Information Security. Springer, August 2011.
[2] S. W. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," NIST Special Publication 800-207. August 2020.
[3] R. Ross, P. Viscuso, G. Guissanie, K. Dempsey, and M. Riddle, "Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations," NIST Special Publication 800-171(R.2). February 2020.
[4] J. X. Wu, "Development paradigms of cyberspace endogenous safety and security," Sci. China Inf. Sci. vol. 65, pp. 1–3, March 2022.
[5] J. X. Wu, "Problems and solutions regarding generalized function safety in cyberspace," Security and Safety. vol. 1, pp. 2022001, June 2022.
[6] Splunk Enterprise 9.1.1, Splunk Inc. https://www.splunk.com/en_us/download/splunk-enterprise.html.
[7] Elasticstack, Elastic Security. https://www.elastic.com/security.
[8] Kibana, Elastic Security. https://www.elastic.co/cn/downloads/kibana.
[9] Microsoft Sentinel, Microsoft. https://azure.microsoft.com/en-us/products/microsoft-sentinel/.
[10] R. Yamagishi, T. Katayama, N. Kawaguchi, and T. Shigemeto, "HOUND: Log Analysis Support for Threat Hunting by Log Visualization," The 12th International Congress on Advanced Applied Informatics (IIAI-AAI), pp. 653–656, 2022.
[11] K. Lamshöft, T. Neubert, J. Hielscher, C. Vielhauer, and J. Dittmann, "Knock, Knock, Log: Threat Analysis, Detection & Mitigation of Covent Channels in Syslog Using Port Scans as Cover," Forensic Science International: Digital Investigation. vol. 40, no. Supplement, pp. 301335, April 2022.
[12] A. S. Malik, M. K. Shahzad, and M. Hussain, "A Forensic Framework for Webmail Threat Detection Using Log Analysis," The 14th International Conference on Innovative Security Solutions for Information Technology and Communications (SecITC), pp. 57–69, 2021.
[13] L. Liu, C. Chen, J. Zhang, O. Y. de Vel, and Y. Xiang, "Doc2vec-based Insider Threat Detection through Behaviour Analysis of Multi-source Security Logs," The 19th IEEE International COnference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 301–309, 2021.
[14] T. Qin, Y. Gao, L. Wei, Z. Liu, and C. Wang, "Potential Threats Mining Methods based on Correlation Analysis of Multi-type Logs," IET Networks, vol. 7, no. 5, pp. 299–305, 2018.
[15] J. Navarro et al., "HuMa: A Multi-layer Framework for Threat Analysis in a Heterogeneous Log Environment," The 10th International Symposium on Foundations and Practice of Security, vol. 10723, pp. 144–159, 2017.
[16] J. X. Wu, "Principles of Cyberspace Mimicry Defense: Generalized Robust Control and Endogenous Security," Wireless Networks. Springer, December 2019.

 ISBN: 978-1-68558-106-0