sensors

MDPI

Article

MTD-Diorama: Moving Target Defense Visualization Engine for Systematic Cybersecurity Strategy Orchestration

Se-Han Lee [1,2], Kyungshin Kim [3], Youngsoo Kim [4] and Ki-Woong Park [2,*]

[1] SysCore Lab., Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea; sehanlee141@gmail.com
[2] Department of Computer and Information Security, Sejong University, Seoul 05006, Republic of Korea
[3] Agency of Defense Development (ADD), Daejeon 34186, Republic of Korea; updatekim@add.re.kr
[4] Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea; blitzkrieg@etri.re.kr
* Correspondence: woongbak@sejong.ac.kr

Abstract: With the advancement in information and communication technology, modern society has relied on various computing systems in areas closely related to human life. However, cyberattacks are also becoming more diverse and intelligent, with personal information and human lives being threatened. The moving target defense (MTD) strategy was designed to protect mission-critical systems from cyberattacks. The MTD strategy shifted the paradigm from passive to active system defense. However, there is a lack of indicators that can be used as a reference when deriving general system components, making it difficult to configure a systematic MTD strategy. Additionally, even when selecting system components, a method to confirm whether the systematic components are selected to respond to actual cyberattacks is needed. Therefore, in this study, we surveyed and analyzed existing cyberattack information and MTD strategy research results to configure a component dataset. Next, we found the correlation between the cyberattack information and MTD strategy component datasets and used this to design and implement the *MTD-Diorama* data visualization engine to configure a systematic MTD strategy. Through this, researchers can conveniently identify the attack surface contained in cyberattack information and the MTD strategies that can respond to each attack surface. Furthermore, it will allow researchers to configure more systematic MTD strategies that can be used universally without being limited to specific computing systems.

Keywords: moving target defense; cyberattack surface; data visualization; classification

Citation: Lee, S.-H.; Kim, K.; Kim, Y.; Park, K.-W. MTD-Diorama: Moving Target Defense Visualization Engine for Systematic Cybersecurity Strategy Orchestration. *Sensors* **2024**, *24*, 4369. https://doi.org/10.3390/s24134369

Academic Editors: Hwa-Young Jeong and Neil Yuwen Yen

Received: 4 June 2024
Revised: 3 July 2024
Accepted: 3 July 2024
Published: 5 July 2024

1. Introduction

Today, with the advancements in information and communication technology (ICT), modern society is developing into a hyperconnected society in which various things are connected through the Internet [1,2]. The Internet of Things (IoT) is currently being used in various industrial fields, such as smart medical devices, autonomous vehicles, smart factories, and smart cities [2–4].

However, cyberattacks are also evolving with the development of ICT [2–8] and go beyond the simple threat of personal information leakage to significantly impacting human life and urban infrastructure (e.g., in-body medical devices and nuclear facility systems). A representative example is the IoT attack using the Mirai Botnet [9], where a vulnerable IoT device infected with malware took control of the host system connected to the IoT device and used it for a large-scale Denial-of-Service (DoS) attack, paralyzing the IoT service. In another case, an attack caused IoT devices to stop functioning or malfunction, paralyzing the smart network infrastructure and causing casualties [10,11].

A moving target defense (MTD) strategy was designed to respond to the evolving cyberattacks. The MTD strategy provides proactive actions that target mission-critical systems, and many research results are currently emerging [12,13]. However, there is a

lack of indicators that suggest which system components can be used, making it difficult to configure a systematic MTD strategy. Additionally, there is a need to use visualization data as indicators to decide from an existing MTD strategy when a specific cyberattack occurs.

Accordingly, in this study, we surveyed and analyzed various existing MTD strategy results to derive critical components from three key perspectives [13]. Additionally, we used the Open Indicator of Compromise (OpenIOC) framework [14] to derive components for cyberattack information and build a dataset using the derived components. Next, we designed and implemented a component data visualization engine to provide visual information connecting the MTD strategy and cyberattack information components. The proposed data visualization engine can help existing MTD strategy researchers confirm whether the currently studied MTD strategy can respond to an actual cyberattack. In addition, it provides future researchers with various component combination indicators to configure new MTD strategies through component connection indicators between the existing MTD strategy research results and actual cyberattack information.

This study is an extension of our previous study [15] that focused on designing and implementing a data visualization engine. In the present study, we show that various components in a general computing system structure can be identified through the implemented data visualization engine, and connection points between existing MTD strategies and cyberattack information components can be identified. Additionally, we show that the proposed data visualization engine can be continuously used to manage and utilize MTD strategy information by constructing a new MTD strategy and immediately adding it to a data visualization engine.

The novelty of the proposed data visualization engine, named *MTD-Diorama*, is evident in two aspects.

First, *MTD-Diorama* effectively visualizes MTD strategies, enabling mission-critical system administrators and cybersecurity experts to respond to various cyberattack scenarios. Unlike traditional MTD strategies that primarily focus on predicting and mitigating attacks, *MTD-Diorama* enhances the visual representation and interactivity of these strategies, facilitating a more intuitive understanding and rapid response.

Second, *MTD-Diorama* integrates multidimensional security data analysis and visualization to help briefly understand the various threat factors that can occur in a complex security environment. Beyond being a simple data visualization tool, it provides a way for cybersecurity practitioners or cybersecurity researchers to evaluate and optimize the effectiveness of MTD strategies in real time.

The remainder of this paper is organized as follows. Section 2 provides an overview of the MTD strategy and describes OpenIOC, which was used to derive the cyberattack information components. Section 3 describes the derivation and classification of the components for the MTD strategy and cyberattack information, as well as the dataset configuration. Section 4 describes the design and implementation of the *MTD-Diorama* data visualization engine, while Section 5 describes its use. Finally, Section 6 presents the conclusions and directions for future research.

2. Preliminary Background

2.1. Moving Target Defense Strategy

A moving target defense (MTD) is a systematic protection strategy that actively and continuously changes the attack surface targeted within a mission-critical system [12]. Thus, the attack surface exposed to the attacker appears chaotic, and the vulnerabilities discovered in advance by the attacker can be nullified over time. An overview of the MTD strategy is shown in Figure 1.

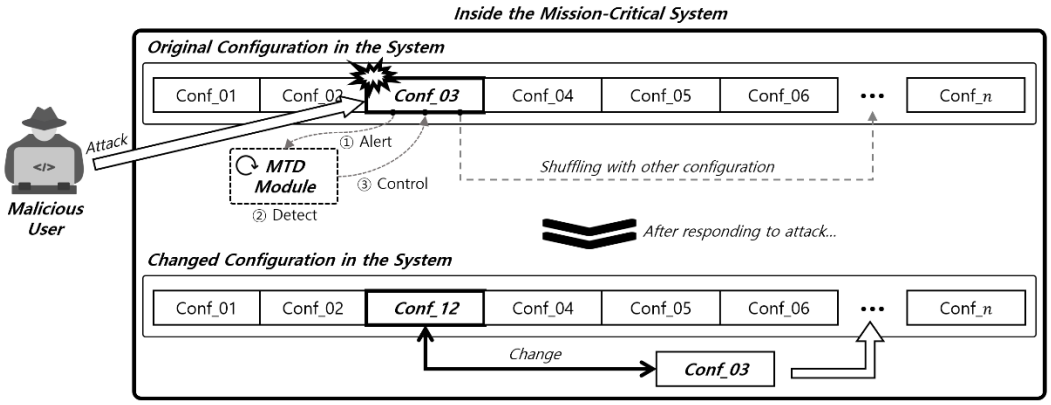

Figure 1. An overview of the moving target defense strategy.

Figure 1 illustrates the workflow for the concept of an MTD strategy within a mission-critical system. Each step involves detecting an attack, responding to it, and reconfiguring the system. The following is a detailed explanation of the workflow:

1. Attack Detection and Alert
   - A malicious user attempts an attack on the system, targeting Conf_03.
   - When the attack reaches Conf_03, the MTD Module within the system detects it.
   - Conf_03 sends an attack notification to the MTD Module (① Alert) to notify that an attack has occurred.
   - Simultaneously, in the ② Detect phase, the module identifies the nature and location of the attack.
2. Response and Control
   - The MTD Module issues a ③ Control command to respond to the attack.
   - This control command instructs the system to change its configuration.
3. System Reconfiguration
   - After the attack is detected, the system reconfigures itself by replacing the compromised configuration Conf_03 with another configuration Conf_12.
   - This change can affect other parts of the system, ensuring that Conf_03 is replaced with Conf_12 to respond to the attack.
4. System Reoperation
   - After the reconfiguration, the system operates with the new configuration Conf_12.
   - Other configurations such as Conf_01, Conf_02, Conf_04, Conf_05, Conf_06, and Conf_n remain unchanged, while Conf_03 has now been replaced by Conf_12.

MTD strategy can reduce the likelihood of successful attacks and effectively improve the resilience and security of mission-critical systems. Moreover, the MTD strategy changes the system security paradigm from the existing passive form of defense to an active defense against cyberattacks [16–22].

This strategy actively changes the mission-critical system components (attack surfaces possibly subjected to cyberattacks). Thus, the effect of obfuscating system components can be achieved such that attackers cannot analyze the system. It also reduces attack opportunities and requires attackers to invest more time in analyzing mission-critical systems.

2.2. Three Perspectives of the MTD Strategy

The MTD strategy can be classified from three perspectives [12,19,22]: What, When, and How to move. The components that make up each perspective come together to form one MTD strategy and determine the direction of cybersecurity technology development using the MTD strategy.

The following is an explanation of each perspective:

- What to move—This perspective concerns which components (the attack surface that an attacker can identify) of the mission-critical system should be moved or mutated when implementing security technology using the MTD strategy. The attack surface defined here incorporates one or more components that comprise the operating system, hardware, and software subject to cyberattacks or containing vulnerabilities, such as IP addresses, MAC addresses, and port numbers in the network area.
- When to move—This time-series perspective determines when to move or mutate the mission-critical system components (the attack surface an attacker can identify). This perspective can affect performance (or availability) when implementing the MTD strategy technology in mission-critical systems. If the frequency of the protection process corresponding to that aspect within the protection technology is too low, the likelihood that an attacker attacks quickly increases. In contrast, if the frequency is too high, although a high level of security service can be provided to the protected system, the resulting large overhead can deteriorate the mission-critical system's performance and service availability.
- How to move—This perspective determines how to move or mutate mission-critical system components. The two tasks that are performed to achieve this goal are selection and replacement. The selection operation selects a new component based on the available data using various methods, such as random data selection or assigning new data to a previously moved or mutated component. The replacement operation transforms one or more components selected through a protection technology process into a new component or exchanges data with one or more components in a mission-critical system.

2.3. Open Indicators of Compromise (OpenIOC) Framework

The OpenIOC framework is an open-source-based cybersecurity incident indicator framework developed by Mandiant [23–26]. It provides various indices to identify data from different attack surfaces contained in a single piece of cyberattack information [24,25]. This incident indicator is widely used when analyzing cyberattacks within governments or companies and is provided as extensible markup language (XML) documents that help capture various artifacts about specific cyberattack information. It is highly recognized when sharing information on cyber incidents and has excellent interoperability with signature-based security devices such as intrusion prevention or detection systems [23,24].

2.4. Cyberattacks in IoT Systems

IoT systems are evolving to enable their use in various fields by the addition of real-time networking characteristics to the structure of existing computing systems [1]. As their usability becomes more important, the frequency of cyberattacks targeting IoT systems is gradually increasing [2–11].

An example of cyberattacks targeting IoT systems includes Reverse Engineering to analyze device firmware vulnerabilities in IoT systems, Man-in-the-Middle attack, Sniffing, Spoofing, and Replay attacks to attack network communication between IoT devices, and Denial-of-Service (DoS) and Side-Channel attacks to attack services operated by IoT systems [27]. From a game theory perspective, these attacks on IoT systems and the defensive actions to stop them can be viewed as a competition to maximize the rewards (reward for attack, reward for defense) from each perspective of the attacker and the defender [28].

Zhang et al. [29] introduced an anti-jamming scheme based on a Colonel Blotto Game to prevent jamming attacks on underwater acoustic backscatter communication used in gliding autonomous underwater vehicles (AUVs). In their study, they modeled the competition between two players, an attacker and a defender, for a limited resource budget to analyze the competitive interaction between surface sink nodes (SNs) and AUVs.

Pirozmand et al. [30] introduce a method that applies game theory to develop an effective intrusion detection system performance in a cloud-fog-based IoT network environment.

In their study, the operation of the intrusion detection system was structured as a dynamic game between two players, an attacker and a defender, and a non-participatory dynamic game, and the parameters for attack and defense were extracted and analyzed.

Abdalzaher et al. [31] introduced a game theory approach to enhance system security and data trustworthiness in wireless sensor network (WSN)-based IoT environments. In their study, a repeated game model between two players, an attacker and a defender, was proposed to enhance clustered WSN-based IoT security and data trustworthiness.

Hence, as the complexity, efficiency, and usability of IoT systems increases, the rewards from both attacker and defender perspectives are also maximized. Based on this, cyberattacks on IoT systems are becoming increasingly intelligent, and it is time for an active defense strategy to be developed for IoT systems.

3. Component Classification and Dataset Configuration

This section describes the derivation and classification of the MTD strategy and cyberattack information components for configuring a visualization dataset for use in a data visualization engine. We then describe how to configure a visualization dataset using the derived components.

3.1. Component Classification of the MTD Strategy

To derive the components of the MTD strategy, we first surveyed and analyzed various MTD strategy research results. The results are given in Table 1.

Table 1. Analysis of existing MTD strategy research results.

Author	Summary	Targeted System Component
Moon [32]	Block the continuity of advanced persistent threat (APT) attacks by deriving system environment elements vulnerable to APT attacks.	System Environment Elements
Leem et al. [33]	Use an attack target disruption mechanism based on a preposition hash table (PHT) to reduce the risk of exposure of hash values in communication network packets to identify unmanned flying objects (drones).	Hash Key in Network IP Structure
Park et al. [34]	Mutate the IP address and port number according to the network-based MTD strategy called hidden tunnel networking.	Network IP Address, Network Port Number
Hong et al. [35]	Provide optimal network configuration through an SDN network topology analysis using shuffle-based online MTD.	Network Topology
Narantuya et al. [36]	Shuffle IP addresses using multiple software-defined network (SDN) controllers in SDN-based network environments.	Network IP Address
Woo et al. [37]	Shuffle the controller area network (CAN) IDs to protect the in-vehicle network using network address shuffling.	CAN Network ID
Brown et al. [38]	Provide a new CAN bus protocol that uses randomization seeds to generate ECU IDs randomly.	ECU ID on the CAN Bus
Park et al. [39]	Use the network protocol variation patterns to ensure that only users who know the patterns can access the server.	Network Protocol
Yoon et al. [40]	Shuffle the network configuration properties (e.g., MAC address, IP address, port number) based on an attack graph of the host system to be protected.	Network Configuration Properties
Groza et al. [41]	Use a strategy to secure CAN network communications by configuring switches in the CAN bus circuit inside the vehicle.	CAN Bus Circuit Board

The current MTD strategy research proposes protection algorithms and systems based on software. In addition, many studies focus on protecting components (e.g., IP address, MAC address, port number, and protocol) identified in the network area. In addition to

software-based technologies, research has focused on hardware-based technologies. An example is in [36], which reconfigured the controller area network (CAN) communication bus circuit inside an autonomous vehicle using a switch and proposed a network flow control mechanism to transmit and receive authenticated CAN network packets.

Based on Table 1, the components that comprise the three perspectives of the MTD strategy were derived and classified. The components are given in Table 2.

Table 2. Critical components derived based on three perspectives of the MTD strategy.

Perspective	Derived Components
When to move	Prevention, Detection
What to move (Attack Surface)	Network IP Address, Network Port Number, Network MAC Address, Network Protocol, Network Packet ID, Network Topology, Hash Key in Network IP Structure, ECU Device ID, Circuit Board, System Information Elements
How to move	Randomization, Patternization, Decoy, Variation, Shuffling, Hardware Switch

First, from the perspective of "What to move", 10 components were derived. The components include attack surfaces that can be identified when an attacker performs a cyberattack targeting a mission-critical system. Second, from the perspective of "When to move", two components that were guaranteed to contain a single piece of time-series information when performing the protection process using the MTD strategy were derived. Finally, six components were derived from the perspective of "How to move". The components exhibit a characteristic that changes the attack surfaces an attacker can identify while performing the protection process using an MTD strategy.

3.2. Component Classification of Cyberattack Information Using the OpenIOC Framework

This study used OpenIOC to derive the components (attack surfaces) of the cyberattack information. It was confirmed that various artifacts of a single attack's information expressed through OpenIOC represented the attack surface from the perspective of "What to move". As an example, the classification of components for Stuxnet attack information is shown in Figure 2.

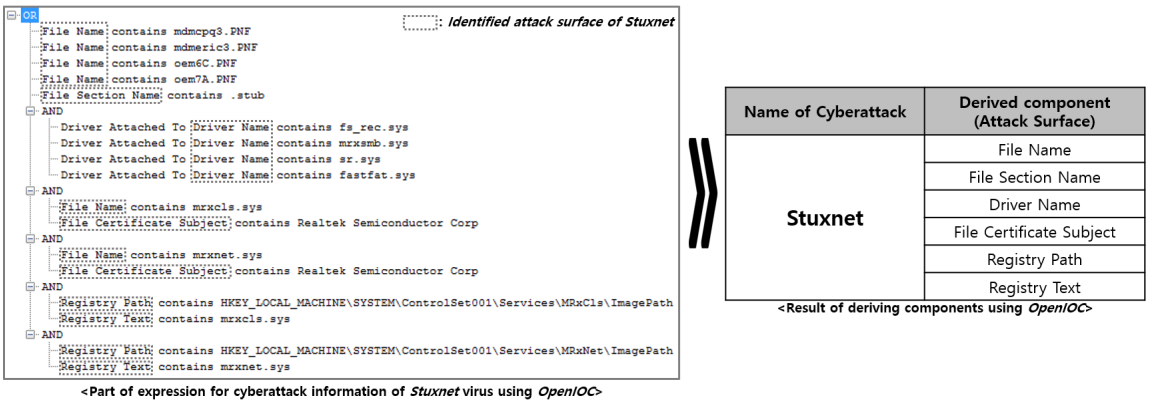

Figure 2. A result of deriving components for Stuxnet virus information.

3.3. Configuration of the Dataset for the Data Visualization Engine

From the previously derived MTD strategies and cyberattack information components, we identified the attack surface as the common component. Therefore, the proposed data visualization engine uses attack surface components to express the correlation between the MTD strategy and cyberattack information. In addition, we aimed to provide

visual information regarding existing MTD strategies that could be used for single-attack information.

To this end, the need to express the components of the attack surface according to the general system configuration was confirmed, and various sections comprising one system and the attack surface corresponding to each section were classified. For deriving and classifying components of cyberattack information, we identified five system sections (Network, Storage, OS Configuration, Application, and Log Sections) by analyzing various artifact indexes provided by the OpenIOC framework [25].

The method for configuring the MTD strategy and cyberattack information datasets is shown in Figure 3. And an example of the dataset configuration based on the previously derived component classification results is shown in Figure 4.

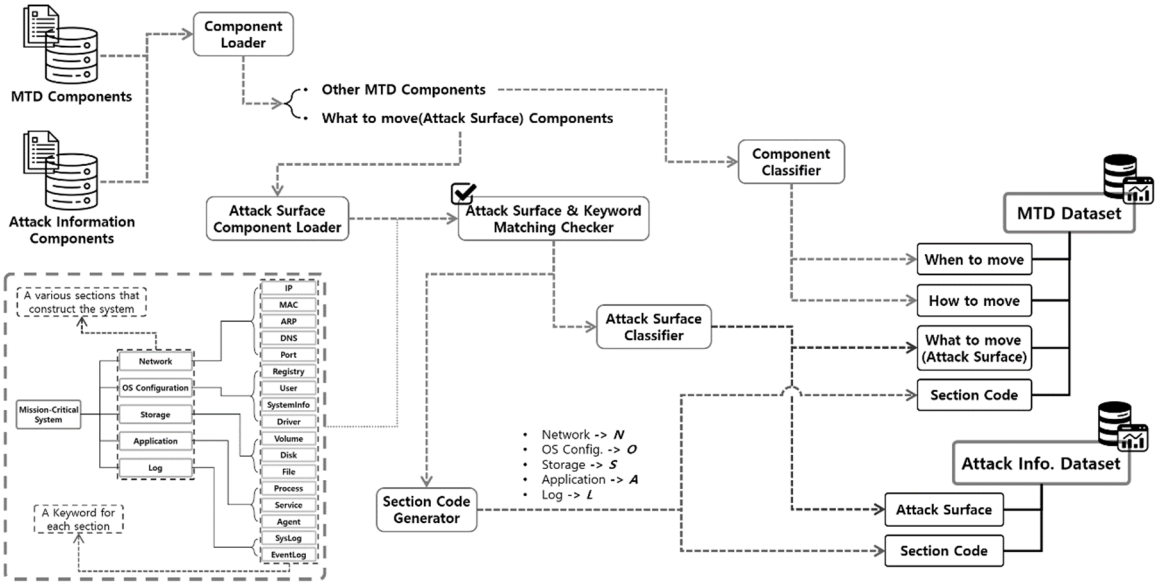

Figure 3. Design diagram for dataset configuration.

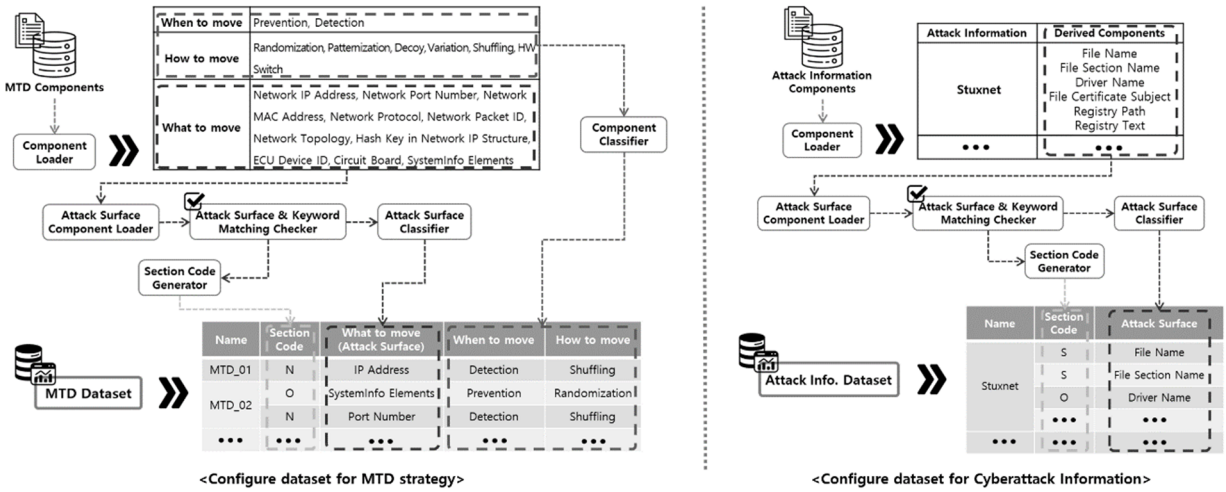

Figure 4. Process of configuring the MTD strategy and cyberattack information datasets.

The section codes corresponding to each section of a general computing system are generated based on the section classification method of the attack surface component. We store the data in the Section Code attribute commonly configured in the two datasets. In addition, to classify the attack surface corresponding to each section, a dataset is configured by classifying the "What to move" component of the MTD strategy and the components of the cyberattack information derived and classified through the keywords for each section.

The explanation of Figure 4 is as follows. The component derivation results are commonly loaded through the Component Loader module when configuring the MTD strategy component and cyberattack information datasets. Subsequently, the attack surface components are separated and added to the dataset, along with the remaining components. Attack surface components are added to the dataset by classifying the system section to which each attack surface component corresponds according to the system section classification method shown in Figure 3.

An example of the MTD strategy component dataset configured using the process shown in Figure 4 is presented in Table 3, and an example of the cyberattack information component dataset is presented in Table 4.

Table 3. Example of the MTD strategy component dataset.

Name	Section Code	What to Move (Attack Surface)	When to Move	How to Move
MTD_01	N	IP Address	Prevention	Shuffling
MTD_01	S	File Path	Prevention	Shuffling
MTD_02	N	MAC Address	Detection	Decoy
MTD_02	A	Process PID	Detection	Decoy
MTD_03	N	Protocol	Detection	Randomization
MTD_03	N	IP Address	Detection	Randomization
MTD_04	S	File Name	Prevention	Randomization
MTD_04	S	File Path	Prevention	Randomization
MTD_04	O	Registry Path	Prevention	Randomization
MTD_05	A	Process BaseAddress	Prevention	Variation
MTD_06	N	IP Hash Key	Detection	Decoy
MTD_06	N	MAC Address	Detection	Decoy
MTD_06	N	Protocol	Detection	Decoy
MTD_06	L	EventLog EID	Detection	Decoy
MTD_07	S	File Type	Prevention	Variation
MTD_07	S	File Timestamp	Prevention	Variation
MTD_07	L	EventLog Index	Prevention	Variation
MTD_08	S	File SectionName	Detection	Shuffling
MTD_08	O	Registry Path	Detection	Shuffling
MTD_08	O	Registry Text	Detection	Shuffling
MTD_09	A	Process PID	Prevention	Shuffling
MTD_09	A	Process Timestamp	Prevention	Shuffling
MTD_10	N	Topology	Detection	HW Switch
MTD_11	N	DNS Host	Prevention	Decoy
MTD_12	S	File sha256sum	Prevention	Randomization
MTD_12	S	Volume DevicePath	Prevention	Randomization
MTD_12	L	EventLog User	Prevention	Randomization
MTD_13	N	Topology	Prevention	Patternization
MTD_14	L	EventLog Type	Detection	Shuffling

Table 4. Example of the cyberattack information component dataset.

Name	Section Code	Attack Surface
Attack_01	N	IP Address
Attack_01	N	MAC Address
Attack_01	S	File Path
Attack_01	A	Process PID
Attack_02	N	Protocol
Attack_02	N	IP Address
Attack_02	S	File Name
Attack_02	S	File Path
Attack_02	O	Registry Path
Attack_02	A	Process BaseAddress
Attack_03	N	IP Hash Key

Table 4. *Cont.*

Name	Section Code	Attack Surface
Attack_03	N	MAC Address
Attack_03	N	Protocol
Attack_03	S	File Type
Attack_03	S	File Timestamp
Attack_03	S	File SectionName
Attack_03	O	Registry Path
Attack_03	O	Registry Text
Attack_03	A	Process PID
Attack_03	A	Process Timestamp
Attack_03	L	EventLog EID
Attack_03	L	EventLog Index
Attack_04	N	Topology
Attack_04	N	DNS Host
Attack_04	S	File sha256sum
Attack_04	S	Volume DevicePath
Attack_04	O	Registry KeyPath
Attack_04	O	Registry Value
Attack_04	A	Process Username
Attack_04	A	Process SecurityID
Attack_04	L	EventLog User
Attack_04	L	EventLog Type

4. Design and Implementation of the Data Visualization Engine

In Section 3, based on the research results of the MTD strategy, we derived the MTD strategy components and constructed a dataset. We also constructed a dataset on cyberattack information. However, there are limitations to simply using a dataset to show the connections between two datasets. First, the dataset itself is simply a collection of data, so it is difficult to directly utilize it for other research. Second, even if it is claimed that there is a connection between the MTD strategy and cyberattack information, it is difficult to utilize it easily if quantitative information (graphs, probabilities, etc.) is not provided.

In this study, we derived and classified the components of existing MTD strategies and cyberattack information and provided visual information to understand the connection between the two. In addition, a component data visualization engine was designed to provide visual information to identify the MTD strategy that could be utilized among the various attack surfaces in actual cyberattacks.

The environment for implementing *MTD-Diorama* and the environment for running the implemented engine are shown in Table 5.

Table 5. The implementation and execution environment of the *MTD-Diorama*.

Type	Environment
Implementation	OS: Windows 10 CPU: Intel Core i5-8500 3.0 GHz Language: Python v3.10 Library: PyQT v5 (for GUI)
Execution	OS: Windows 10 CPU: Intel Core i5-8500 3.0 GHz Memory: 8 GB

The system design of the proposed *MTD-Diorama* data visualization engine is illustrated in Figure 5.

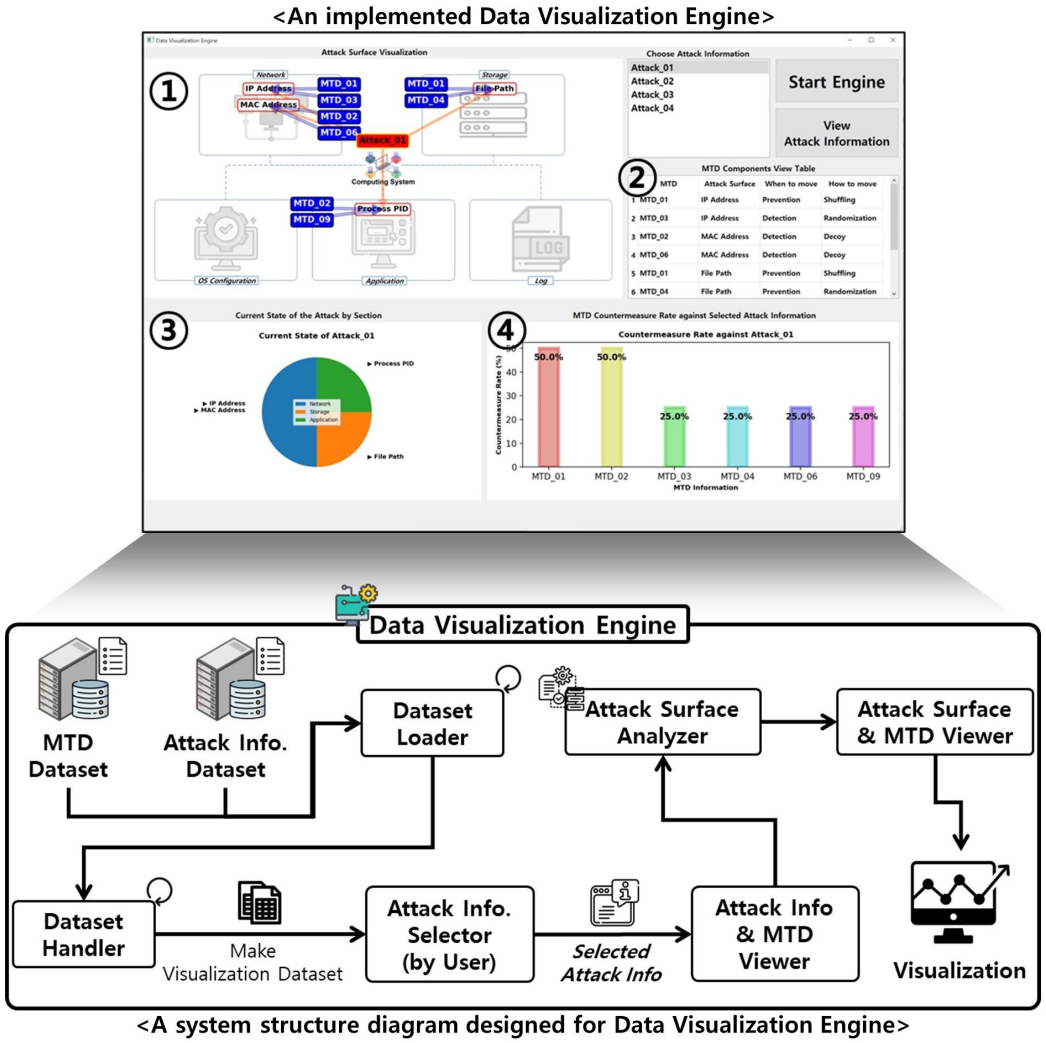

Figure 5. System design and implementation result for *MTD-Diorama*.

The system structure designed to implement the data visualization engine is as follows. The Dataset Loader module loads the previously configured MTD strategy and cyberattack information component datasets. The Dataset Handler module creates a new dataset for the data visualization engine. The Attack Information Selector module identifies attack information selected by the engine user. Finally, the Attack Information & MTD Viewer, Attack Surface Analyzer, and Attack Surface & MTD Viewer modules generate and provide various types of visual information based on the attack information the user selects. The top of Figure 5 shows the final implemented data visualization engine, which is described as the following:

① The computing system section classified along with data construction in Section 3 is expressed. When the user selects the desired cyberattack information and runs the engine, a computing system attack surface associated with the chosen cyberattack information appears and an MTD strategy that can respond to each attack surface is expressed.

② The MTD strategy component dataset that can respond to the cyberattack information selected by the user is displayed in a table format. Users can check which components of an MTD strategy can respond to each piece of cyberattack information and use it as an indicator when constructing a new MTD strategy.

③ The chart provides visual information to check which section of each computing system the cyberattack information selected by the user is attacking and the attack surface in that section.

④ The chart identifies MTD strategy information that can respond to the cyberattack information selected by the user and provides visual information on the rate at which the identified MTD strategy information can respond to the cyberattack information.

5. Utilization of a Data Visualization Engine

The *MTD-Diorama* visually shows the connectivity of existing MTD strategy information and the components of cyberattack information using the OpenIOC framework. In addition, it helps formulate an MTD strategy to respond systematically to various cyberattacks and can be used as an indicator to determine the configuration direction of a new MTD strategy. Figure 6 shows an example when running *MTD-Diorama*.

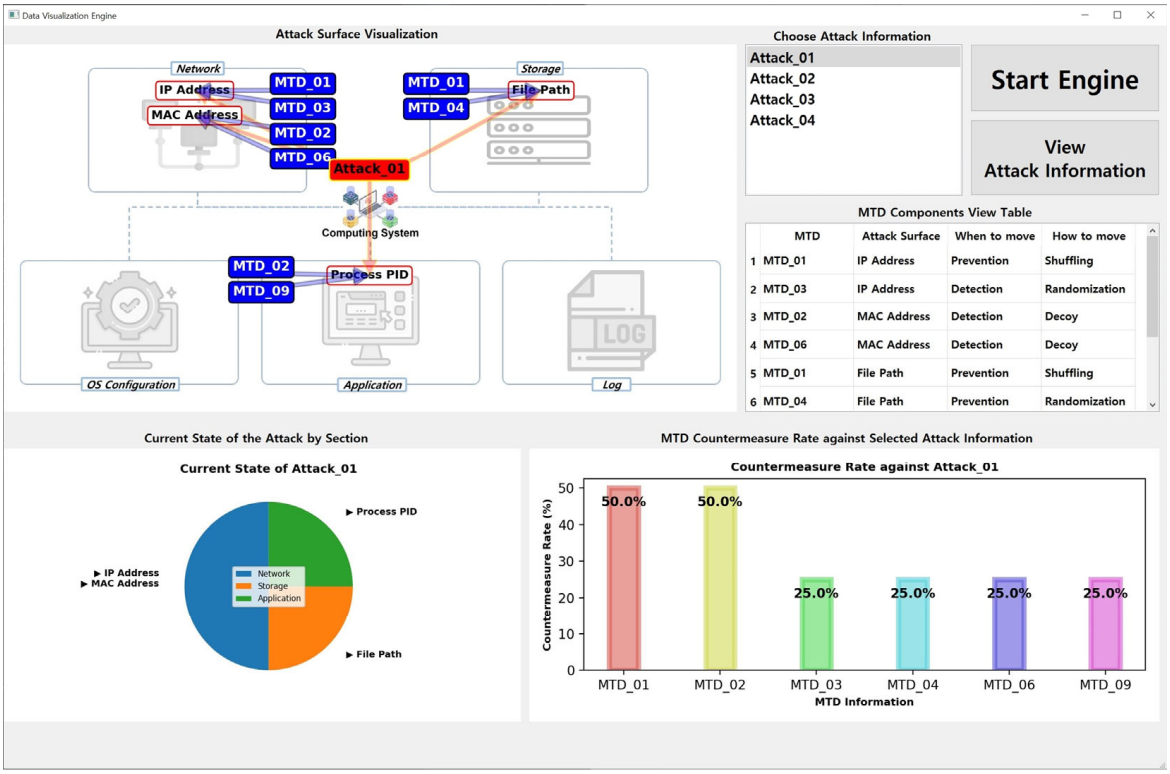

Figure 6. Result of the *MTD-Diorama* operation (when choosing Attack_01).

The explanation for Figure 6 is as follows. There are four attack surfaces (IP Address, MAC Address, File Path, and Process PID) in the selected Attack_01 information, and there are six MTD strategies (MTD_01, MTD_02, MTD_03, MTD_04, MTD_06, and MTD_09) that can respond to them. In this case, MTD_01 and MTD_02 can effectively respond to Attack_01 each with a 50% probability.

The results shown in Figure 6 were obtained from an engine run using the datasets listed in Tables 3 and 4. As a data visualization engine, *MTD-Diorama* provides various visual information and can easily confirm connection points between the MTD strategy and cyberattack information. Therefore, existing MTD strategy researchers can use it as an indicator of the practicality of their studied MTD strategy. The results when selecting different attack information are shown in Figures 7 and 8, and the accuracy and usability of these results increase depending on the size of the datasets.

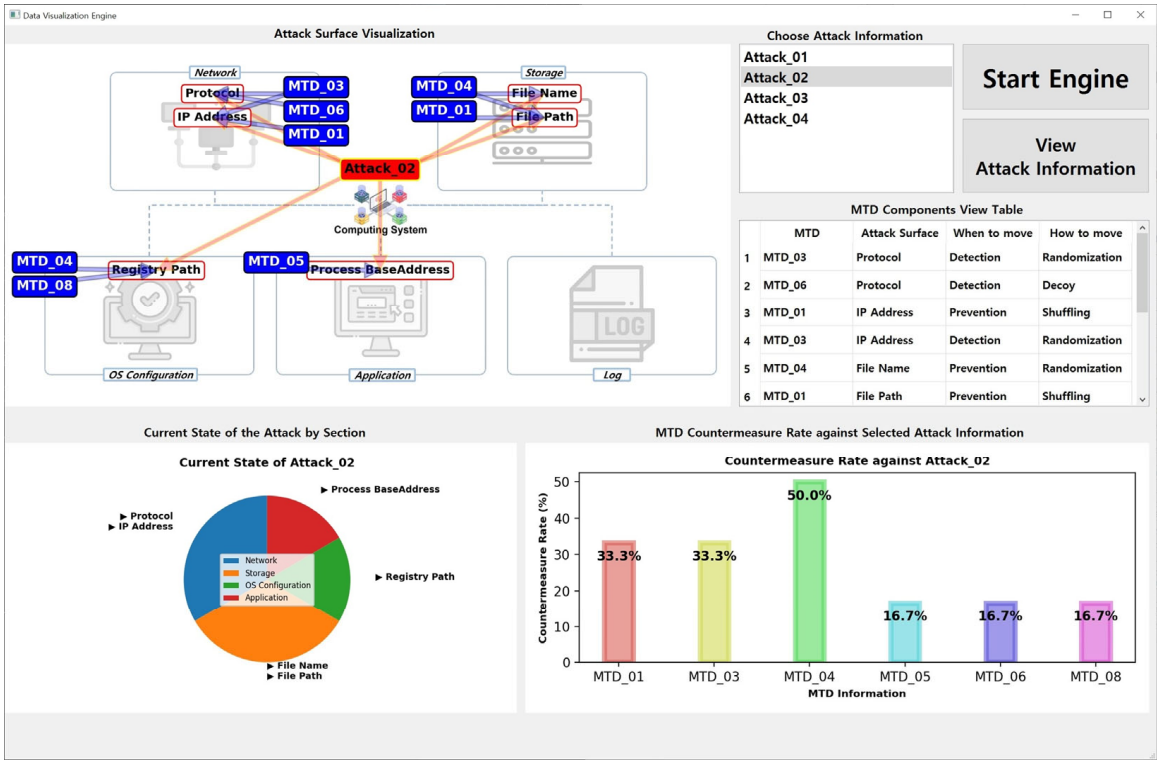

Figure 7. Result of the *MTD-Diorama* operation (when choosing Attack_02).

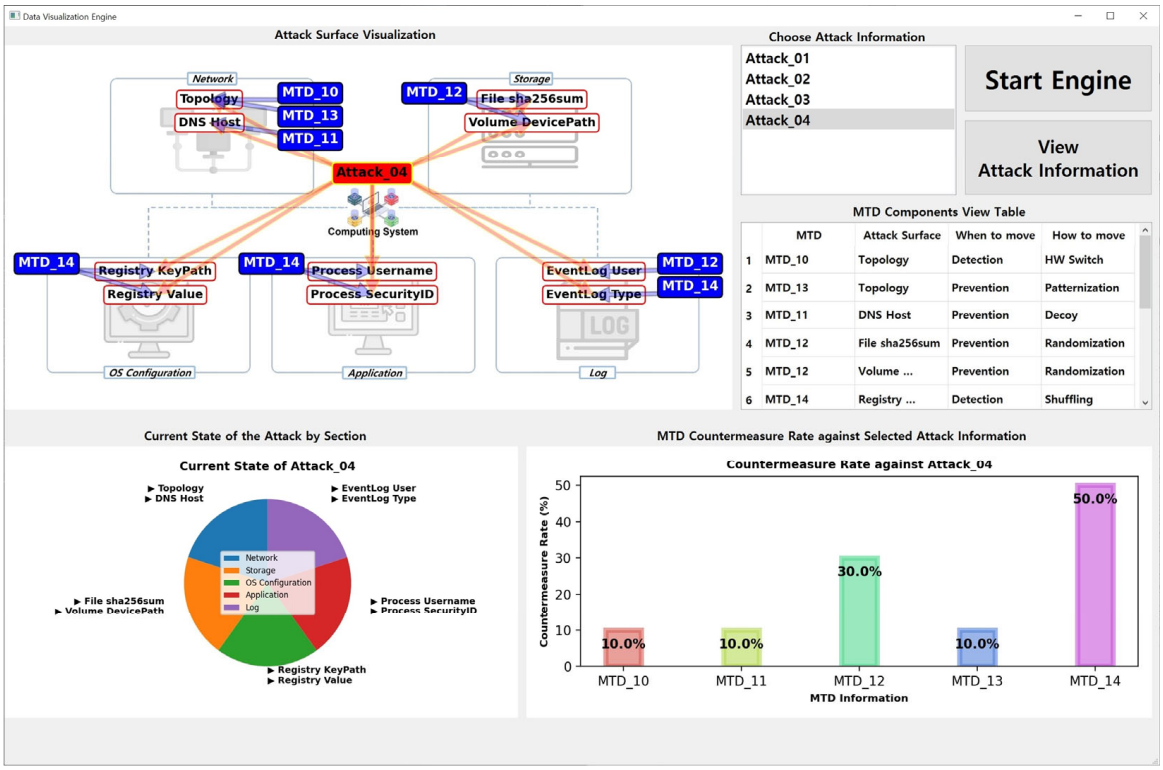

Figure 8. Result of the *MTD-Diorama* operation (when choosing Attack_04).

In addition, the *MTD-Diorama* can be used to configure new MTD strategies by combining the components of existing MTD strategy research results, as shown in Figure 9.

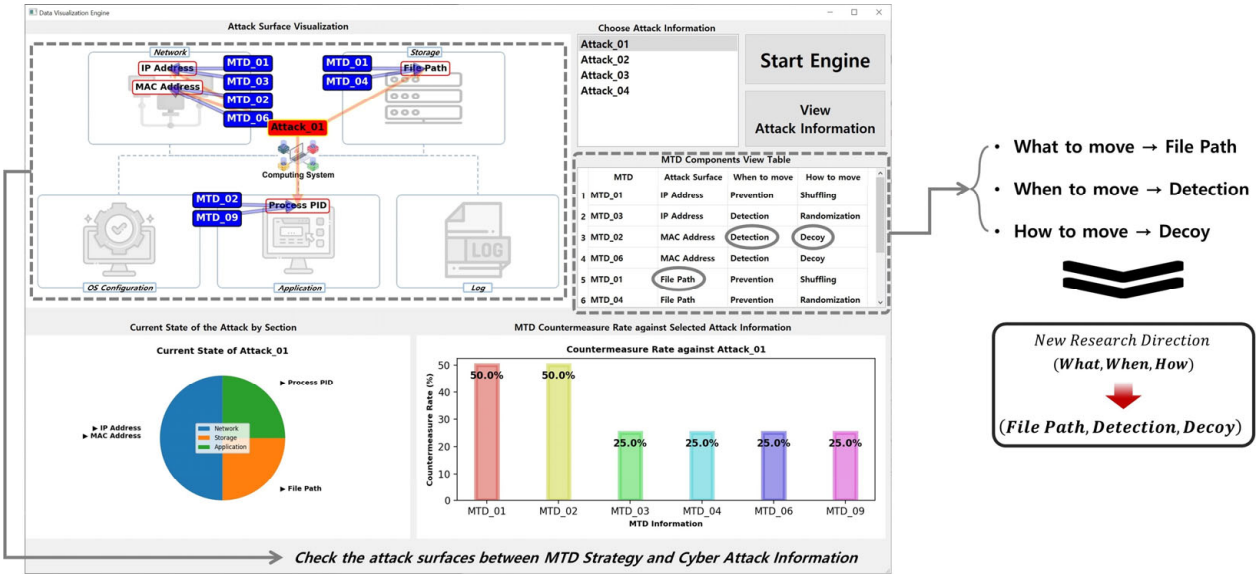

Figure 9. Method of deriving a new MTD strategy research direction using *MTD-Diorama*.

Moreover, the visual information displayed in *MTD-Diorama* can be used as an indicator to determine the research or development direction when configuring new MTD strategies.

When a new MTD strategy or cyberattack information is identified, the data visualization engine can immediately add it to the dataset and utilize it. An example is shown in Figure 10. Even if new information is identified, users can add the desired information at any time based on the dataset configuration described in Section 3. The added data are immediately reflected in the data visualization engine. Accordingly, users can continuously use it as an indicator to check the connection between the existing and new MTD strategy and cyberattack information.

As can be seen from the research results of the MTD strategy surveyed and analyzed in Section 3.1, the field in which the MTD strategy has most widely been used to date is the network field. Using *MTD-Diorama* as a visualization engine in a network field, especially in a software-defined network (SDN) environment [42–44], can suggest new strategic possibilities for network security. By combining the visualization and component dataset-understanding capabilities of *MTD-Diorama* with the central control and flexible network configuration capabilities of SDN, a more efficient and effective cybersecurity defense system can be built. This provides great advantages in implementing dynamic defense strategies and responding to real-time threats, especially in complex network environments.

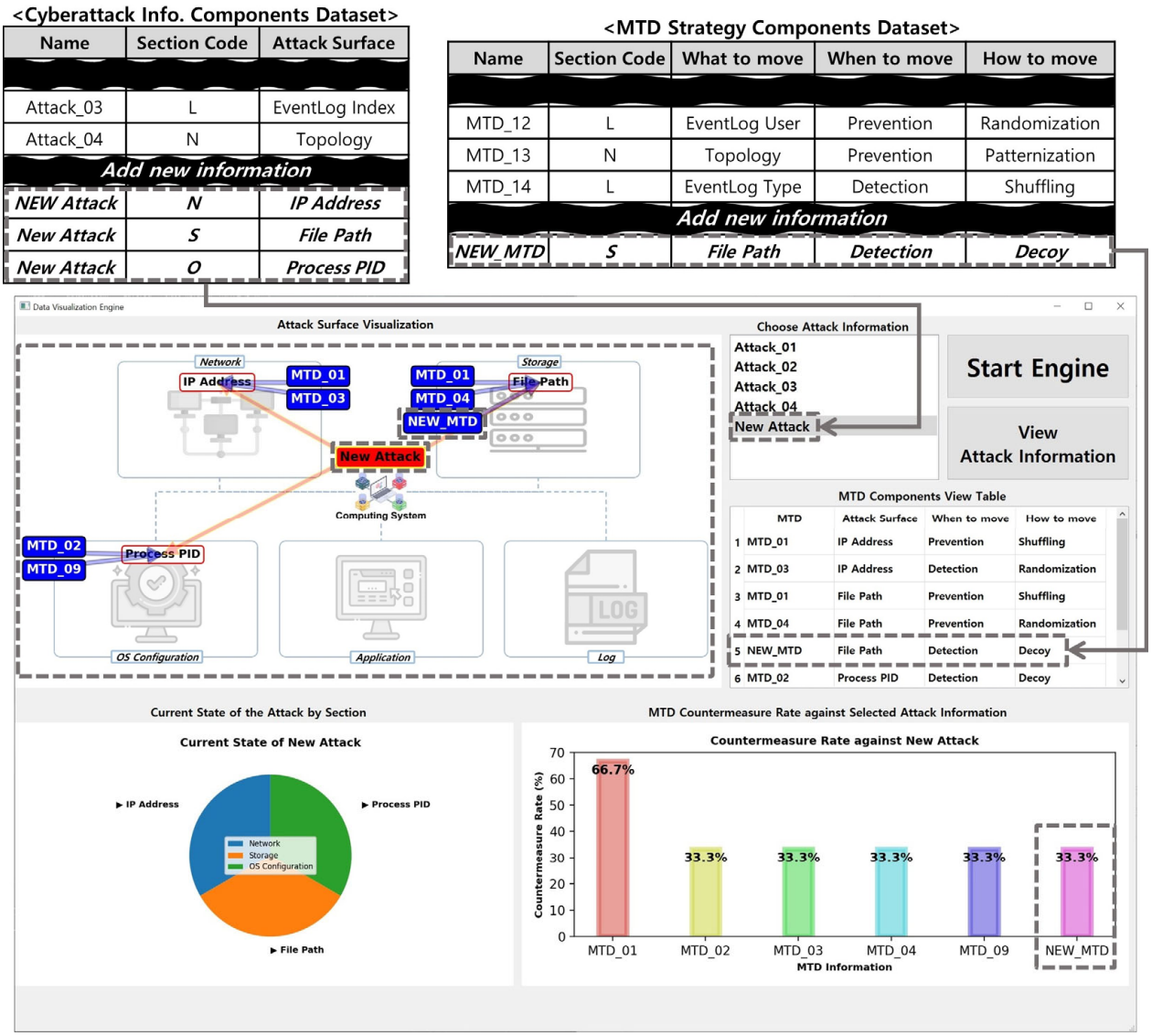

Figure 10. Example of *MTD-Diorama* when adding new information in the dataset.

6. Conclusions

With the development of ICT, cyberattacks are also developing and increasing. The MTD strategy helps configure a preemptive defense strategy for a mission-critical system and respond to these cyberattacks. However, with the increasing diversity of computing systems, there is a need to formulate a systematic MTD strategy that can be utilized from the perspective of a general computing system. Therefore, an indicator that can be used as a reference is required.

This study proposes a data visualization engine visually demonstrating the connection between the MTD strategy and cyberattack information. Using the proposed engine, users can check the components of the MTD strategy and cyberattack information and use them as indicators when configuring a new security strategy.

However, there are shortcomings in directly proving how systematic the newly configured MTD strategy using *MTD-Diorama* is. To achieve this, when a newly configured MTD strategy technology operates in a specific system, a method is needed to measure the overhead of that system and show the actual internal system configuration. In addition, the visualization method using the current component dataset has a shortcoming in that it is difficult to derive results related to the status information of the actual computing system.

To achieve this, a more detailed component classification method is needed, such as the usage data range of each component, data type, actual data values, etc.

In the future, we plan to study an extended and specified component classification scheme and develop an advanced *MTD-Diorama* data visualization engine based on it to show how not only the components in a large-scale concept but also the detailed data of the components interact in a computing system. Moreover, we plan to study ways to identify what happens within a mission-critical system in real time when responding to a cyberattack by simulating how the MTD strategy operates in a computing system.

Finally, we plan to build a testbed environment based on digital twins [45] for simulation verification of the advanced *MTD-Diorama*. This will be used to compare and analyze the real-time cyberattack response results of the MTD strategy in a virtually implemented computing system with the response results in an actual computing system and derive an interactive relationship. To achieve this, we plan to build, test, and verify this using the Software-In-The-Loop Simulation (SITL) method and the Hardware-In-The-Loop Simulation (HILS) method [46,47].

Author Contributions: Conceptualization, S.-H.L. and K.-W.P.; methodology, S.-H.L.; formal analysis, S.-H.L.; investigation, S.-H.L.; data curation, S.-H.L.; writing—original draft preparation, S.-H.L.; writing—review and editing, K.-W.P.; visualization, S.-H.L.; supervision, K.-W.P.; project administration, K.-W.P.; funding acquisition, K.K., Y.K. and K.-W.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Future Challenge Defense Technology Research and Development Project (9150921) hosted by the Agency for Defense Development Institute in 2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data are contained within this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nord, J.H.; Koohang, A.; Paliszkiewicz, J. The Internet of Things: Review and theoretical framework. *Expert Syst. Appl.* **2019**, *133*, 97–108. [CrossRef]
2. Noor, M.M.; Hassan, W.H. Current research on Internet of Things (IoT) security: A survey. *Comput. Netw.* **2019**, *148*, 283–294. [CrossRef]
3. Koutras, D.; Stergiopoulos, G.; Dasaklis, T.; Kotzanikolaou, P.; Glynos, D.; Douligeris, C. Security in IoMT Communications: A Survey. *Sensors* **2020**, *20*, 4828. [CrossRef]
4. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]
5. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]
6. Cho, E.; Park, S.; Kang, N. Hacking Attacks and Countermeasures using Vulnerabilities of Lightweight IP Camera in Internet of Things. *J. Digit. Contents Soc.* **2019**, *20*, 1069–1077. [CrossRef]
7. Kim, H.; Lee, H.; Lee, Y. A Survey Analysis of Internet of Things Security Issues and Combined Service. *J. Korea Soc. Comput. Inf.* **2020**, *25*, 73–79.
8. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. Internet of Things: Security and Solutions Survey. *Sensors* **2022**, *22*, 7433. [CrossRef]
9. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the Mirai Botnet. In Proceedings of the 26th USENIX Security Symposium, Vancouver, BC, Canada, 16–18 August 2017.
10. Mun, H.; Choi, G.; Hwang, Y. Countermeasure to Underlying Security Threats in IoT communication. *J. Converg. Inf. Technol.* **2016**, *6*, 37–44.
11. Grammatikis, P.I.R.; Sarigiannidis, P.G.; Moscholios, I.D. Securing the Internet of Things: Challenges, threats and solutions. *Internet Things* **2019**, *5*, 41–70. [CrossRef]
12. Cho, J.; Sharma, D.P.; Alavizadeh, H.; Yoon, S.; Ben-Asher, N.; Moore, T.J.; Kim, D.S.; Lim, H.; Nelson, F.F. Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 709–745. [CrossRef]

13. Lee, S.; Park, K. A Method for Derivation of Software-Defined MTD Research Direction for secure IoT Device through Analysis of MTD Strategy Research Result. *JDCA* **2022**, *5*, 147–158.
14. Zhao, J.; Yan, Q.; Li, J.; Shao, M.; He, Z.; Li, B. TIMiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data. *Comput. Secur.* **2020**, *95*, 101867. [CrossRef]
15. Lee, S.; Alawami, M.A.; Park, K. Data Visualization Engine for systematic MTD Strategy Configuration linked to Cyber Attack Information. In Proceedings of the 9th International Conference on Next Generation Computing (ICNGC 2023), Da Nang, Vietnam, 20–23 December 2023.
16. Hong, J.B.; Kim, D.S. Assessing the Effectiveness of Moving Target Defenses Using Security Models. *IEEE Trans. Dependable Secur. Comput.* **2016**, *13*, 163–177. [CrossRef]
17. Zheng, J.; Namin, A.S. A Survey on the Moving Target Defense Strategies: An Architectural Perspective. *J. Comput. Sci. Technol.* **2019**, *34*, 207–233. [CrossRef]
18. Burow, N.; Burrow, R.; Khazan, R.; Shrobe, H.; Ward, B.C. Moving Target Defense Considerations in Real-Time Safety- and Mission-Critical Systems. In Proceedings of the 7th ACM Workshop on Moving Target Defense (MTD'20), Online, 9–13 November 2020.
19. Navas, R.E.; Cuppens, F.; Cuppens, N.B.; Toutain, L.; Papadopoulos, G.Z. MTD, Where Art Thou? A Systematic Review of Moving Target Defense Techniques for IoT. *IEEE Internet Things J.* **2021**, *8*, 7818–7832. [CrossRef]
20. Zhuang, R.; DeLoach, S.A.; Ou, X. Towards a Theory of Moving Target Defense. In Proceedings of the First ACM Workshop on Moving Target Defense (MTD'14), Scottsdale, AZ, USA, 3–7 November 2014.
21. Xu, J.; Guo, P.; Zhao, M.; Erbacher, R.F.; Zhu, M.; Liu, P. Comparing Different Moving Target Defense Techniques. In Proceedings of the First ACM Workshop on Moving Target Defense (MTD'14), Scottsdale, AZ, USA, 3–7 November 2014.
22. Cai, G.; Wang, B.; Hu, W.; Wang, T. Moving target defense: State of the art and characteristics. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 1122–1153. [CrossRef]
23. Cho, H. Analysis of Cyber Threat Level Based on Indicator of Compromise. Master's Thesis, Sungkyunkwan University, Seoul, Republic of Korea, August 2017.
24. Kim, S. A Method to Indicator Compromise Utilization for the Effective Infringement Accident Analysis. Master's Thesis, Konkuk University, Seoul, Republic of Korea, August 2015.
25. OpenIOC 1.1. Available online: https://github.com/fireeye/OpenIOC_1.1 (accessed on 1 May 2024).
26. Utilization of IOC, IOAF and SigBase. Available online: http://forensicinsight.org/wp-content/uploads/2013/05/F-INSIGHT-Utilization-of-IOC-IOAF-and-SigBase.pdf (accessed on 1 May 2024).
27. Shah, Y.; Sengupta, S. A survey on Classification of Cyber-attacks on IoT and IIoT devices. In Proceedings of the 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 28–31 October 2020.
28. Wang, Y.; Wang, Y.; Liu, J.; Huang, Z.; Xie, P. A Survey of Game Theoretic Methods for Cyber Security. In Proceedings of the 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), Changsha, China, 13–16 June 2016.
29. Zhang, L.; Wang, Z.; Zhang, H.; Min, M.; Wang, C.; Niyato, D.; Han, Z. Anti-Jamming Colonel Blotto Game for Underwater Acoustic Backscatter Communication. *IEEE Trans. Veh. Technol.* 2024, *early access*. [CrossRef]
30. Pirozmand, P.; Ghafary, M.A.; Siadat, S.; Ren, J. Intrusion Detection into Cloud-Fog-Based IoT Networks Using Game Theory. *Wirel. Commun. Mob. Comput.* **2020**, *1*, 8819545. [CrossRef]
31. Abdalzaher, M.S.; Muta, O. A Game-Theoretic Approach for Enhancing Security and Data Trustworthiness in IoT Applications. *IEEE Internet Things J.* **2020**, *7*, 11250–11261. [CrossRef]
32. Moon, S.Y. A Study on the Moving Target Defense Model for Advanced Persistent Threat Security. In Proceedings of the 2018 Korean Institute of Communications and Information Sciences (KICS) Summer Conference, Jeju, Republic of Korea, 20–22 June 2018.
33. Leem, S.; Lee, M.; Lim, J. MTD (Moving Target Detection) with Preposition Hash Table for Security of Drone Network. *J. Korea Inst. Inf. Commun. Eng.* **2019**, *23*, 477–485.
34. Park, T.; Park, K.; Moon, D. Attack Surface Expansion through Decoy Trap for Protected Servers in Moving Target Defense. *J. Korea Soc. Comput. Inform.* **2019**, *24*, 25–32.
35. Hong, J.B.; Yoon, S.; Lim, H.; Kim, D.S. Optimal Network Reconfiguration for Software Defined Networks Using Shuffle-Based Online MTD. In Proceedings of the IEEE 36th Symposium on Reliable Distributed Systems (SRDS), Hong Kong, China, 26–29 September 2017.
36. Narantuya, J.; Yoon, S.; Lim, H.; Cho, J.; Kim, D.S.; Moore, T.; Nelson, F. SDN-Based IP Shuffling Moving Target Defense with Multiple SDN Controllers. In Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks–Supplemental Volume (DSN-S), Portland, OR, USA, 24–27 June 2019.
37. Woo, S.; Moon, D.; Youn, T.; Lee, Y.; Kim, Y. CAN ID Shuffling Technique (CIST): Moving Target Defense Strategy for Protecting In-Vehicle CAN. *IEEE Access* **2019**, *7*, 15521–15536. [CrossRef]
38. Brown, R.; Marti, A.; Jenkins, C.; Shannigrahi, S. Dynamic Address Validation Array (DAVA): A Moving Target Defense Protocol for CAN bus. In Proceedings of the 7th ACM Workshop on Moving Target Defense (MTD'20), Online, 9–13 November 2020.
39. Park, J.; Lee, Y.; Kang, K.; Lee, S.; Park, K. Ghost-MTD: Moving Target Defense via Protocol Mutation for Mission-Critical Cloud Systems. *Energies* **2020**, *13*, 1883. [CrossRef]

40. Yoon, S.; Cho, J.; Kim, D.S.; Moore, T.J.; Free-Nelson, F.; Lim, H. Attack Graph-Based Moving Target Defense in Software-Defined Networks. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1653–1668. [CrossRef]
41. Groza, B.; Popa, L.; Murvay, P.; Elovici, Y.; Shabtai, A. CANARY—A reactive defense mechanism for Controller Area Networks based on Active RelaYs. In Proceedings of the 30th USENIX Security Symposium, Online, 11–13 August 2021.
42. Siddiqui, S.; Hameed, S.; Shah, S.A.; Ahmad, I.; Aneiba, A.; Draheim, D.; Dustdar, S. Toward Software-Defined Networking-Based IoT Frameworks: A Systematic Literature Review, Taxonomy, Open Challenges and Prospects. *IEEE Access* **2022**, *10*, 70850–70901. [CrossRef]
43. Gao, S.; Peng, Z.; Xiao, B.; Hu, A.; Song, Y.; Ren, K. Detection and Mitigation of DoS Attacks in Software Defined Networks. *IEEE/ACM Trans. Netw.* **2020**, *28*, 1419–1433. [CrossRef]
44. Shang, G.; Zhe, P.; Bin, X.; Aiqun, H.; Kui, R. FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks. In Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017.
45. Zhang, L.; Wang, H.; Xue, H.; Zhang, H.; Liu, Q.; Niyato, D.; Han, Z. Digital Twin-Assisted Edge Computation Offloading in Industrial Internet of Things with NOMA. *IEEE Trans. Veh. Technol.* **2023**, *72*, 11935–11950. [CrossRef]
46. Jain, S.; Pappachan, P.; Guajardo, J.; Trieflinger, S.; Raghupatruni, I.; Huber, T. CMP-SiL: Confidential Multi Party Software-in-the-Loop Simulation Frameworks. In Proceedings of the 2023 24th International Symposium on Quality Electronic Design (ISQED), San Francisco, CA, USA, 5–7 April 2023.
47. Ravikumar, G.; Hyder, B.; Govindarasu, M. Hardware-in-the-Loop CPS Security Architecture for DER Monitoring and Control Applications. In Proceedings of the 2020 IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 6–7 February 2020.