applied sciences

MDPI

Article

Location-Aware Measurement for Cyber Mimic Defense: You Cannot Improve What You Cannot Measure

Zhe Huang [1], Yali Yuan [1,2,*], Jiale Fu [3], Jiajun He [3], Hongyu Zhu [1] and Guang Cheng [1,4]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China; chengguang@seu.edu.cn (G.C.)
[2] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
[3] School of Mathematics, Southeast University, Nanjing 211189, China
[4] Jiangsu Province Engineering Research Center of Security for Ubiquitous Network, Nanjing 211189, China
* Correspondence: yaliyuan@seu.edu.cn

Abstract: Cyber mimic defense is designed to ensure endogenous security, effectively countering unknown vulnerabilities and backdoors, thereby addressing a significant challenge in cyberspace. However, the immense scale of real-world networks and their intricate topology pose challenges for measuring the efficacy of cyber mimic defense. To capture and quantify defense performance within specific segments of these expansive networks, we embrace a partitioning approach that subdivides large networks into smaller regions. Metrics are then established within an objective space constructed on these smaller regions. This approach enables the establishment of several fine-grained metrics that offer a more nuanced measurement of cyber mimic defense deployed in complex networks. For example, the common-mode index is introduced to highlight shared vulnerabilities among diverse nodes, the transfer probability computes the likelihood of risk propagation among nodes, and the failure risk assesses the likelihood of cyber mimic defense technology failure within individual nodes or entire communities. Furthermore, we provide proof of the convergence of the transfer probability. A multitude of simulations are conducted to validate the reliability and applicability of the proposed metrics.

Keywords: cyber mimic defense; complex network measurement; metrics

check for updates

Citation: Huang, Z.; Yuan, Y.; Fu, J.; He, J.; Zhu, H.; Cheng, G. Location-Aware Measurement for Cyber Mimic Defense: You Cannot Improve What You Cannot Measure. *Appl. Sci.* **2023**, *13*, 9213. https://doi.org/10.3390/app13169213

Academic Editors: Peter R.J. Trim and Yang-Im Lee

Received: 30 June 2023
Revised: 9 August 2023
Accepted: 9 August 2023
Published: 13 August 2023

1. Introduction

The increasingly widespread application of the Internet in various social and economic sectors is leading to an increasingly severe challenge for cyberspace. Network security threats are becoming more diverse, complex, frequent, and widespread. In the current online environment, there exists a significant asymmetry between network attacks and defenses [1], often favoring the attackers. From the defensive perspective, it is generally difficult to anticipate when and how attacks will occur, making it challenging to deploy targeted defense strategies.

Traditional defense techniques, such as firewalls and intrusion detection techniques [2,3], typically rely on known attack signatures to identify and match target behaviors, leaving them at a disadvantage against unknown vulnerabilities and backdoors in cyber warfare. A series of novel proactive defense technologies are proposed to address this issue, such as honeypots [4] and Moving Target Defense (MTD) [5–7]. These methods effectively improved the situation and significantly increased the difficulty and cost for attackers to launch their attacks. However, they still have limitations: honeypot technology requires a significant amount of prior knowledge from attackers [8], and MTD possesses time sensitivity and uncontrollability, and the high-frequency variability, particularly, leads to a decline in system performance [7].

In fact, there is no defense strategy that can achieve absolute security. Due to the stage-specific nature of technological development and the level of awareness, vulnerabilities or backdoor issues in software and hardware design cannot be completely avoided. In the absence of the ability to eliminate inherent flaws and lack of prior knowledge, addressing the threat of unknown vulnerabilities and backdoors remains a significant challenge in cybersecurity. The emergence of the Cyber Mimic Defense (CMD) theory [9] provided a new idea and paradigm to tackle this problem and demonstrated effective defense capabilities in areas such as Software-Defined Networking (SDN) [10], cloud computing [11], distributed systems [12], etc.

The core framework of Cyber Mimic Defense (CMD) is "Dynamic Heterogeneous Redundancy" (DHR) [9], which is characterized by the following: (1) Dynamic: selecting a set of functional executors based on scheduling policies at the current moment and continuously changing this set to conceal the internal structure. (2) Heterogeneous: utilizing multiple heterogeneous executors with significantly different implementation methods to achieve the same functionality. (3) Redundancy: employing multiple executors and using an adjudication mechanism to determine the final system output.

Through its structural effects, CMD achieves endogenous defense effects that are independent of attack characteristics, effectively countering various attacks and unknown threats. However, there is currently a lack of universal metrics to directly measure the effectiveness of cyber mimic defense technology when applied to modern networks. We cannot improve what we cannot measure [13], and this principle applies to cyber mimic defense technology as well. The vast scale of real-world networks and the complexity of their topology pose challenges for evaluating the effectiveness of the cyber mimic defense. Therefore, there is an urgent need to develop general quantitative evaluation metrics for cyber mimic defense systems.

It is widely recognized that no single metric is powerful enough to fully reflect the impact of all relevant behaviors and defense strategies on the network. Therefore, we establish multi-dimensional evaluation metrics to assess the effectiveness of cyber mimic defense technology from various perspectives. We also found that most existing security strategies are typically evaluated based on the entire network. However, in many cases, even the best defense strategy may not necessarily extend security uniformly across the entire network, especially in large networks with hundreds or thousands of nodes. If we use security metrics based on the overall network assessment and observe improved security, it could be misleading as the security improvement may be limited to certain parts of the network. As mentioned earlier, asymmetry in network attacks and defenses exists, particularly in large-scale networks. The location from which attackers launch their attacks is difficult to predict, and the scope of protection provided by defense strategies is often limited. In such cases, global metrics fail to clearly reflect the defensive performance. In other words, metrics can reflect the overall defensive performance but cannot pinpoint the exact location of changes in defensive information. Therefore, it is necessary to adopt a location-aware method.

The main contributions of our work are summarized as follows. Firstly, to capture variations in attack-defense performance within specific local networks, we utilize a network partitioning method (i.e., Louvain algorithm) to segment the extensive network into smaller segments and perform correlation metrics on these segments to achieve a more fine-grained assessment. Subsequently, we establish relevant security metrics within the partitioned objective space for cyber mimic defense. These metrics encompass various quantitative measures such as the common-mode index and failure risk, which are tailored to accurately reflect the effectiveness of cyber mimic defense systems. This results in the creation of an innovative approach for effectively assessing cyber mimic defense deployed in complex networks. Finally, simulated attacks are carried out to verify the applicability and effectiveness of the proposed metrics.

This paper is organized in the following way. In Section 2, we introduce the preliminaries; in Section 3, we give the network partitioning method; in Section 4, we define metrics in the constructed objective space; in Section 5, we perform simulation experiments and give results; in Section 6, we list related work; and in Section 7, we make a conclusion of our work and propose a vision for future development.

2. Preliminaries

In this section, we provide a concise overview of the CMD framework, including its key concepts, and illustrate the actual topology of modern networks.

2.1. CMD Framework

As depicted in Figure 1, the cyber mimic defense system [9] primarily comprises six components: input agent, online set of executors, back-up executor pool, arbiter, scheduler, and output agent. During runtime, the input agent acquires input data and duplicates them to distribute among the heterogeneous executors in the online executor set. Each executor independently processes the data and produces an output. The arbiter then adjudicates the outputs of each executor based on a predefined algorithm to determine the final output. Additionally, the arbiter provides feedback on the adjudication to the scheduler, which uses this information to dynamically update the online executor set using specific strategies. The functional details of each component are described as follows.

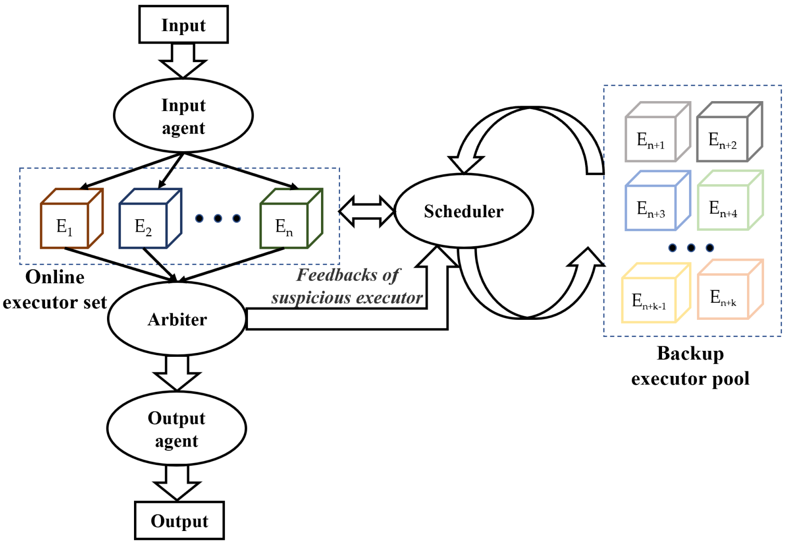


Figure 1. The Framework for Cyber Mimic Defense.

Input Agent: The input agent obtains the input, and copies and distributes it to each online executor.

Online executor set: Each online executor possesses an equivalent function and operates independently to process input. It is crucial to ensure a high degree of heterogeneity among the executors to mitigate common-mode vulnerabilities effectively. Once the calculations are completed, the results are transmitted to the arbiter for further processing.

Backup executor pool: The backup of online executors. The scheduler periodically or selectively chooses an instance to replace the currently active online executor set.

Arbiter: The arbiter adjudicates the output results of each executor based on a predefined algorithm and provides feedback regarding any suspicious executor to the scheduler.

Scheduler: The scheduler dynamically dispatches executors based on the operational status of online executors, handling tasks such as offline cleaning of suspicious executors and periodic replacement of executors.

Output Agent: The output agent obtains the voting results from the arbiter, formats it if necessary, and then outputs the final result.

With its endogenous security mechanism rooted in dynamic, heterogeneous, and redundancy strategies, cyber mimic defense establishes a spatiotemporal inconsistency scenario, preventing attackers from replicating past successes. It enhances the concealability and camouflage of the target defense scenario and behavior. Even in the event of an attack, the attacker cannot simultaneously breach all the actuators (exponential difficulty) [9], ensuring that the functions protected by the imitation system remain undisturbed and achievable. As a result, cyber mimic defense gains a more robust advantage when dealing with persistent, stealthy, and high-intensity offensive and defensive scenarios, especially in the presence of uncertain threats including unknown vulnerabilities, backdoors, and viruses.

2.2. Network Topology

Contrary to popular belief, most real-world networks do not exhibit random structures. Instead, they often follow a scale-free network concept [14], where a small number of nodes have a large number of connections, while the majority have only a few connections. Scale-free networks are complex networks characterized by a degree distribution that closely follows a power-law distribution. In such networks, the probability of a node having k connections (i.e., degree k) follows a power-law distribution, denoted as $P(k) \sim k^{-\gamma}$. The power exponent γ represents the structural properties of the network. The scale-free network exhibits significant heterogeneity, and the distribution of connections among its nodes is remarkably uneven, effectively simulating real-world network conditions.

In the era of the Internet of Everything, network-connected devices are diverse and encompass not only computers but also switches, sensors, smart home devices, and more. These devices can be regarded as nodes within the network. Let N represent the graph that illustrates the physical topology of the network, and we use a binary group of nodes and their connections to represent the network, denoted as $N = \langle V, E \rangle$. Here, V refers to the devices in the network, collectively known as hosts, and E represents the undirected edge connections between them. Subsequent studies are based on the proposed network topology as described above.

3. Network Partitioning

In this section, we discuss the necessity of performing network partitioning on complex networks for evaluating CMD and present how to utilize the Louvain algorithm for network partitioning.

Due to the immense scale of modern networks, conducting an evaluation of CMD technology from a global perspective in complex network environments is often imprecise and challenging. The location of attacker intrusions is random and unpredictable, and the effective coverage of CMD technology typically cannot encompass the entire large-scale network. This means that attacks conducted outside the effective range of CMD may remain largely unaffected. If an overall improvement in security is observed from global evaluation metrics, it could lead to misjudgments about the effectiveness of CMD technology. Therefore, it is necessary to adopt a divide-and-conquer approach, evaluating the effectiveness of CMD technology within smaller regions to enhance the accuracy and applicability of the metrics.

As mentioned in Section 2.2, there is significant heterogeneity in the connections between nodes in real-world networks. This often leads to the aggregation of nodes, forming communities within the network. Community structure is one of the essential features of complex networks [15]. Each module or community is composed of closely connected individuals due to similar structural characteristics and positions. Community detection methods are specifically designed to partition the internal structure of complex networks with the goal of grouping network nodes into tightly connected communities. Compared to other traditional methods, community detection methods pay more attention to the patterns of connections between nodes, allowing for better capture of the local structural characteristics of the network. We utilize a typical community detection algorithm, the Louvain algorithm, to partition complex networks, which is described in detail below.

The Louvain algorithm [16] is grounded in multilevel optimization of modularity, offering the advantage of speed and accuracy in obtaining a hierarchical community structure with approximately linear time complexity. It takes a heuristic approach to maximize the local modularity of smaller communities, joining only if such aggregation leads to an increase in modularity. It is the preferred method for clustering (community detection) of complex networks [17] and was rated as one of the best community detection algorithms by [18]. Two key concepts are integral to the algorithm: modularity Q and modularity gain ΔQ [16], for which we provide the relevant formulas below.

- Modularity (Q)

$$Q = \sum_C [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2], \tag{1}$$

where m denotes the total number of edges in the graph, $\sum_{in}$ denotes the sum of the weights of the edges interconnected within community C, and $\sum_{tot}$ denotes the sum of the weights of the edges connected to the nodes of community C, including the edges inside the community as well as the edges outside the community.

- Modularity gain (ΔQ)

$$\begin{aligned}\Delta Q(i \to C) &= [\frac{\sum_{in} + k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2] - \\ &[\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2] \\ &= \frac{1}{2m}(k_{i,in} - \frac{\sum_{tot} k_i}{m}),\end{aligned} \tag{2}$$

where k_i denotes the sum of the weights of the edges connected to node i and $k_{i,in}$ denotes the sum of the weights of the edges of node i connected to the nodes in community C.

The main flow of the Louvain algorithm is as follows:

1. Initially, each node is regarded as a separate community;
2. For each node i, try to assign it to a neighbor community in turn and calculate the modularity gain ΔQ after assignment, find the assignment method with the maximum modularity gain and assign it if its $\Delta Q > 0$, otherwise leave it unchanged;
3. Repeat the steps in 2 until the communities in which all nodes are located no longer change;
4. Compress a community into a new node, convert the weights of edges interconnected by nodes within the community to the weights of the ring of the new node, and convert the weights of edges between communities to the weights of the edges between the new nodes;
5. Repeat the above steps until the results converge.

Through the Louvain algorithm, we distinguish the network structure hierarchically with a high degree of association between hosts within the community. Next, we use the delineated communities as the objective space to develop the definition of metrics for cyber mimic defense.

4. Metrics in the Objective Space

In this section, based on the objective space constructed by network partitioning, we develop multidimensional evaluation metrics to measure the effectiveness of cyber mimic defense technology.

4.1. Single Node

Definition 1 (Network topology)**.** *We represent the network community as a binary group $NC_i = (N, E)$, where N is the set of all nodes (hosts) in the network, including switches, routers, firewalls, etc., and E is the set of connection relationships between these nodes.*

Definition 2 (Vulnerability set)**.** *A collection of all possible vulnerabilities, especially zero-day vulnerabilities.*

$$VUL = VUL_1 \cup VUL_2 \cup \cdots \cup VUL_n,$$

where VUL_i represents the set of all vulnerabilities on node Ni (suppose n nodes in the community) and $VUL_i = \{vul_j | vul_j$ is a certain vulnerability on node $V_i\}$.

In CMD systems, the adjudication algorithm generally follows the "majority voting" principle. Consequently, when more than half of the executors possess the same symbiotic vulnerabilities and are successfully exploited by an attacker, they can potentially deceive the arbiter and allow the attack to evade detection, similar to an environment where CMD is not deployed. As a result, we propose the following hypothesis.

Assumption 1. *For a CMD system with (2l + 1) online executors, when there exists an (l + 1) order symbiotic vulnerability vul_t, it is considered that the node where the CMD system is deployed has vulnerability vul_t.*

Definition 3 (Vulnerability vector)**.** *Construct vulnerability vector V_i for each node.*

$$\mathbf{V_i} = (v_1, v_2, \ldots, v_m)^T, m = |VUL|,$$

$$v_k = \begin{cases} 1 & vul_k \in VUL_i \\ 0 & vul_k \notin VUL_i \end{cases}, \; k = 1, 2, \cdots, m.$$

Definition 4 (Node–vulnerability matrix)**.** *The indication of the corresponding relationship between nodes and vulnerabilities.*

$$\mathbf{NV_{n*m}} = (\mathbf{V_1}, \mathbf{V_2}, \ldots \mathbf{V_n})^T.$$

Definition 5 (CVSS vector)**.** *Construct vulnerability score vector CVSS for each vulnerability.*

$$\mathbf{CVSS} = (cvss_1, cvss_2, \ldots, cvss_m)^T,$$

where $cvss_i(i = 1, 2, \ldots, m)$ represents one-tenth (For normalization) of the Common Vulnerability Scoring System (CVSS) score for the corresponding vulnerability.

Definition 6 (Importance vector)**.** *Construct importance vector IM for each node considering the centrality (location of nodes in the community) and value (value of resources owned by nodes).*

$$\mathbf{IM} = (im_1, im_2, \ldots, im_n)^T,$$

$$im_k = w_1 \times centrality + w_2 \times value, \; k = 1, 2, \ldots, n.$$

where $w_i(i = 1, 2)$ represents the weight of the corresponding factors, $\sum w_i = 1$. Here, the weights and factors can be appropriately adjusted according to the actual situation.

In Definitions 2–5, the vulnerability set and CVSS score can be obtained from the open CVE vulnerability database. In Definition 6, the value depends on the property and resources owned by the nodes, and the centrality calculation method needs to be selected according to the actual network situation from degree centrality, betweenness centrality, closeness centrality, etc., and all of these concepts are defined.

Independent failure risk.

In CMD systems, the heterogeneity among different executors within the redundant structure is crucial and directly impacts the overall performance of the model. When multiple executors share the same vulnerability, it can lead to attacks escaping detection, rendering the cyber mimic defense strategy ineffective. From the perspective of individual nodes, we define the independent failure risk based on vulnerabilities, represented as an n-dimensional vector, where the i-th component represents the independent failure risk of node N_i. In the formula, the importance vector **IM** represents the likelihood of an attacker choosing to target a node, while **NV** × **CVSS** represents the likelihood of successfully compromising a node if targeted in an attack.

$$\mathbf{RI} = \mathbf{IM}^T \times \mathbf{NV} \times \mathbf{CVSS}. \tag{3}$$

4.2. Relationship between Nodes

Definition 7 (Executor set)**.** *The set of executors in the CMD system, each capable of independently implementing service functions, is denoted as* $A = \{A_1, A_2, \ldots\}$*, where* A_i *represents a specific executor.*

Definition 8 (Higher-order symbiotic vulnerability)**.** *Exploitable vulnerabilities that can achieve the same attack effect for m executors in executor set A (*$m \geqslant 3$*).*

Definition 9 (Adjacency matrix)**.** *The adjacency matrix represents the adjacency between nodes (We assume undirected edges in the community, so it is a symmetric matrix).*

$$\mathbf{Adj} = (a_{ij})_{n\times n},$$

$$a_{ij} = \begin{cases} 1 & edge_{ij} \in E \\ 0 & edge_{ij} \notin E \end{cases}.$$

Common-mode index.

When multiple nodes share similar vulnerabilities, attackers can rapidly exploit these vulnerabilities on one node, potentially affecting multiple nodes in a similar or identical manner. This leads to the rapid horizontal spread of the attack's impact, resulting in irreversible consequences. Nodes within the same network community are closely interconnected and often exhibit similar or identical component structures in the real environment, such as accessory modules purchased from the same batch of manufacturers. Therefore, we introduce the concept of the common-mode index to measure the similarity within a community, thereby revealing potential security risks.

We define the common-mode index between node N_s and node N_t ($N_s, N_t \in N$) as follows.

$$I_{(N_s,N_t)} = \frac{\sum_{vul_k \in (VUL_s \cap VUL_t)} CVSS_{vul_k}}{\sum_{vul_k \in (VUL_s \cup VUL_t)} CVSS_{vul_k}}, \tag{4}$$

where $CVSS_{vul_k}$ indicates the CVSS score of the vulnerability vul_k to characterize the magnitude of the vulnerability's harm.

Considering the particularity of the CMD system, we need to make another consideration for the node where CMD is deployed. In CMD, multiple redundant executors independently run the output results and obtain the final results through adjudication, and the online executors are in a state of dynamic transformation. Due to these characteristics, the cognition of the vulnerability set of the node where CMD is deployed needs to be changed. After deploying CMD on the node, its vulnerability set is in a dynamic state. Accordingly, we give the definition of the common-mode index between the node N_{cmd} where CMD is deployed and the ordinary node N_x.

$$I_{(N_{CMD},N_x)} = \sum I_{(N_{CMD},N_x),t_i} \times \frac{t_i}{T}, \tag{5}$$

where $I_{(N_{cmd},N_x),t_i}$ denotes the common-mode index in a period t_i for a dynamically updated CMD vulnerability set and T denotes a period as long as possible to show the possible states of the executor set.

Based on the above definition, we integrate the common-mode index between nodes into matrix form, as shown below.

$$\mathbf{CM} = (c_{ij})_{n\times n},$$

$$c_{ij} = I_{(N_i,N_j)}. \tag{6}$$

Transfer probability.

When an adversary breaks through a host, they often use this host as a base and then launch attacks on other hosts to expand the control range and spread worms and viruses. Typically, the attack spreads from the compromised host to neighboring hosts, gradually infecting the entire network. Considering the attack transfer between neighboring nodes, we propose the concept of transfer probability.

We construct the transfer matrix to represent the single-step transfer probability.

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}, \tag{7}$$

where n represents the number of nodes in the community and p_{ij} denotes the probability of an attacker moving from node i to node j in a single step, which is defined as follows.

$$p_{ij} = \begin{cases} 0 & a_{ij} = 0 \\ \frac{p_1}{d_i} & a_{ij} = 1 \; and \; t_{ij} = 0, \\ \frac{p_2}{d_i} & a_{ij} = 1 \; and \; t_{ij} = 1 \end{cases} \tag{8}$$

where p_1, and p_2, respectively, represent the success rate of transfer from node i to node j with or without common-mode vulnerability (prior knowledge is needed for machine learning in practical application), d_i denotes the degree of the i-th node, and a_{ij} is an element in the adjacency matrix **A**. If $a_{ij} = 1$ then it means there is an edge between node i and node j, $a_{ij} = 0$ means there is no edge between node i and node j, and t_{ij} is an element in $\mathbf{T_{n*n}}$, which is defined as follows.

$$\mathbf{T} = (t_{ij})_{n*n},$$

$$t_{ij} = \begin{cases} 1 & VUL_s \cap VUL_t \neq \varnothing \\ 0 & VUL_s \cap VUL_t = \varnothing \end{cases}.$$

From the above definition, we can know:

1. $0 \leqslant p_1 \leqslant p_2 \leqslant 1$
2. $\mathbf{P} = \mathbf{P^T}$

Before starting the definition of transfer probability, a related lemma and a theorem are given.

Lemma 1. *$\|P\|_1 < 1$, where $\|\cdot\|_1$ is the 1-norm of the matrix.*

We know that $\|P\|_1 = \max\limits_{1 \leqslant i \leqslant n} \sum_{j=1}^{n} |p_{ij}|$, so to prove $\|P\|_1 < 1$, we have to prove $\max\limits_{1 \leqslant i \leqslant n} \sum_{j=1}^{n} |p_{ij}| < 1$, that is, to prove $\forall i$, $\sum_{j=1}^{n} |p_{ij}| < 1$.

$\forall i$, we can prove that $\sum_{j=1}^{n} |p_{ij}| = \sum_{j=1}^{n} p_{ij} = \sum_{j=1}^{n} a_{ij} p_{ij} \leqslant \sum_{j=1}^{n} a_{ij} \frac{p_2}{d_i} = \frac{p_2}{d_i} \sum_{j=1}^{n} a_{ij} = \frac{p_2}{d_i} d_i = p_2$.

Therefore, $\|P\|_1 \leqslant p_2 < 1$. The lemma is proved.

Theorem 1. *If $\|P\| < 1$, then $I + P + P^2 + \cdots P^n + \cdots$ converges, and $I + P + P^2 + \cdots P^n + \cdots = (I - P)^{-1}$.*

Since $\|P\| < 1$, then $\|I\| + \|P\| + \|P\|^2 + \cdots + \|P\|^n + \cdots$ converges. And since the completeness of $(P_{n*n}, \|\cdot\|)$, then $I + P + P^2 + \cdots P^n + \cdots$ converges.

$$\begin{aligned}&(I-P)\left(I+P+P^2+\cdots P^n+\cdots\right)\\=&\left(I+P+P^2+\cdots P^n+\cdots\right)\\&-\left(P+P^2+\cdots P^n+\cdots\right)\\=&I.\end{aligned}$$

Therefore, $I+P+P^2+\cdots P^n+\cdots=(I-P)^{-1}$*. The theorem is proved.*

Combining the basic transfer factor ε_0 and the multi-step transfer case, we define the transfer probability matrix as follows.

$$\begin{aligned}\mathbf{TP}&=\varepsilon_0\mathbf{Adj}+(\mathbf{P})+(\mathbf{P})^2+(\mathbf{P})^3+\cdots\\&=\varepsilon_0\mathbf{Adj}+(\mathbf{I}-\mathbf{P})^{-1}-\mathbf{I}.\end{aligned}\tag{9}$$

4.3. Entire Community

Comprehensive failure risk.

In the previous sections, we established quantitative evaluation metrics for individual nodes and between nodes. By combining these metrics, we defined the comprehensive community failure risk, which assesses the local effectiveness of deploying the cyber mimic defense strategy. We formulated it as a quadratic expression as shown below, where **RI** represents the individual node failure risk, and $\mathbf{CM}\odot\mathbf{TP}$ represents the probability of attack spread.

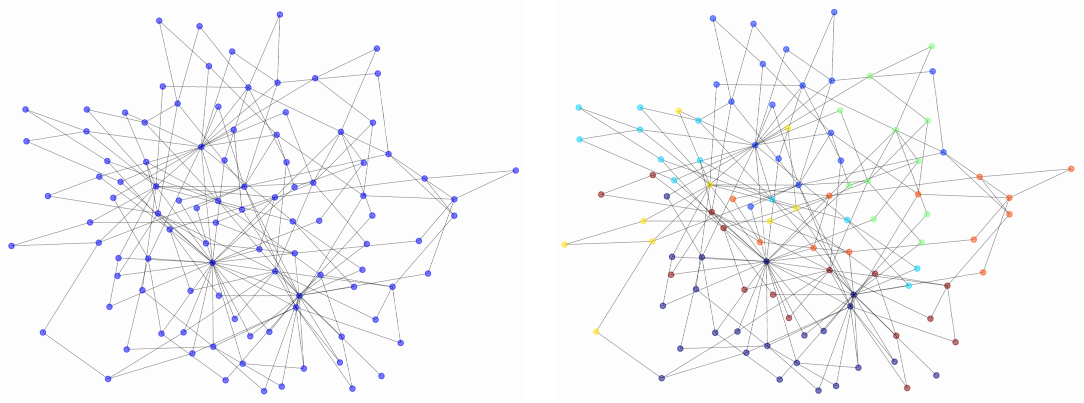
$$RC=\mathbf{RI}^{\mathbf{T}}\times(\mathbf{CM}\odot\mathbf{TP})\times\mathbf{RI},\tag{10}$$

where $\odot$ denotes the Hadamard product, i.e., the multiplication of corresponding elements.

5. Simulation

In this section, we conduct simulation experiments and comparative analysis to validate the effectiveness and rationality of the above metrics.

Firstly, we use the NetworkX package in Python to generate a scale-free network structure with a large number of nodes and apply the Louvain algorithm to partition the network. After a limited number of iterations, the finite number of communities was successfully divided. We color different communities separately for visualization, and the example effect is shown in Figure 2.



(**a**) Before division (**b**) After division

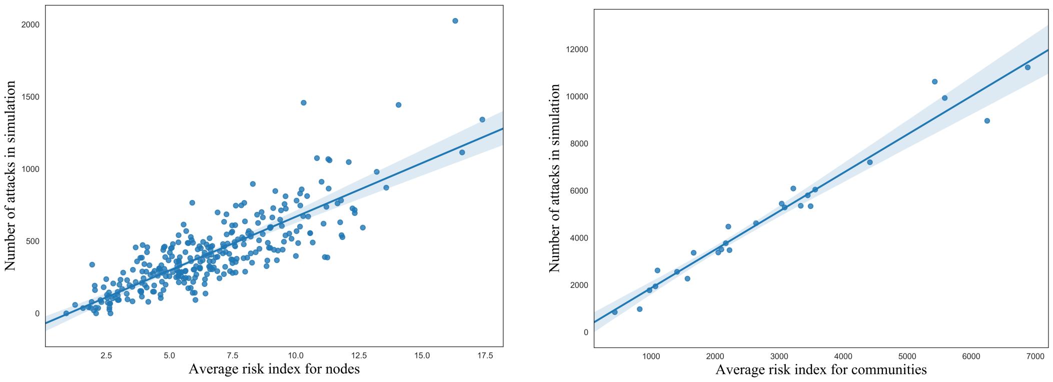
Figure 2. The example effect of community division.

Secondly, we construct the vulnerability set by selecting n vulnerabilities from the open CVE database and then generate each component of the vulnerability vector corresponding to each node with probability p. For nodes with CMD deployed, we add up the vulnerability vectors of each executor in the online executor set (assuming a total of $2l + 1$ online executors), and consider a component in the resulting sum vector as indicating the presence of a specific vulnerability if it is greater than l.

After completing the community partitioning and selecting the vulnerability set, we utilize the network topology structure, vulnerability information, and pre-defined asset values as inputs to calculate the metric values according to the formulas provided in Section 4. To validate the reasonableness of the proposed metrics, we conduct simulated attacks and compared the computed metric values with the results of the simulated attacks

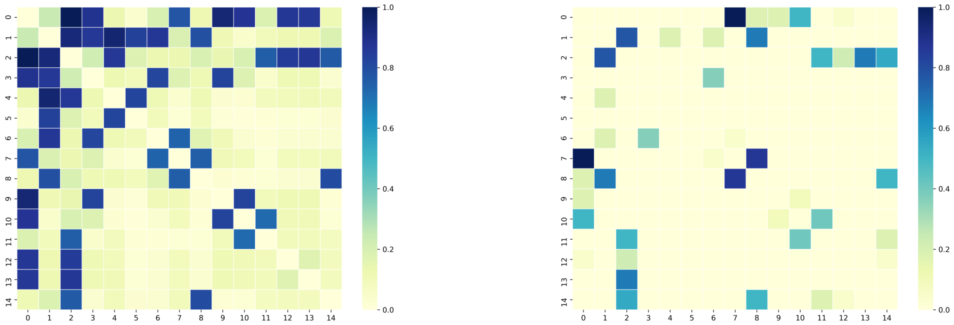
Since the location of the attack initiation in the network is generally unknown, it can be regarded as a random event [19]. We simulate the attacks from a probabilistic perspective, where each simulation involves multiple attacks, and each attack is considered independent. A simulated attack can be divided into the following three phases: initial attack, horizontal spread, and clearance. (1) Initial attack: The attacker randomly selects a node and a vulnerability v from the vulnerability set to attack. If the chosen node possesses vulnerability v, the attack is considered successful; otherwise, it fails. (2) Horizontal spread: If the attacker successfully infiltrates the network in the initial attack, at each time step, it can attempt to spread to neighboring nodes. If a neighbor node lacks vulnerability v, the success rate of spreading to it is denoted by p_1; otherwise, it is set to p_2 (where $p_1 < p_2$). (3) Clearance: Considering that both regular nodes and CMD nodes have their own checking and clearing mechanisms, we assume that at each time step, there is a certain probability of the attacker being detected and cleared by the node's protection mechanism. For regular nodes, there is a probability of $p_r ec$ to find and clear exploitable vulnerabilities at each time step ($p_r ec$ is set based on the actual probability). For CMD nodes, if fewer than half of the executors have vulnerability v in the dynamic scheduling of each time step, the attacker will not achieve their goal under the CMD's ruling mechanism. This scenario is equivalent to the vulnerability being cleared. The simulated attack continues until it no longer spreads, and then this round of simulated attack is concluded.

In the simulated attacks mentioned above, the number of simulated attacks on nodes or communities can reflect their actual vulnerability to some extent. Combining the calculated failure risk of nodes and communities (i.e., independent and comprehensive failure risk) with the number of attacks, we draw a scatter plot on the two-dimensional coordinate system, as shown in Figure 3. After fitting the scatter points, it can be seen that the number of attacks is roughly proportional to the calculated failure risk. We also performed a comparative analysis of the transfer probabilities between nodes, part of which is visualized in Figure 4. The squares within the i-th row and j-th column of the figure represent the transfer probability from node i to node j, with color intensity denoting the probability magnitude Comparing the calculated and experimental results, we observe a close alignment. All of these indicate that our proposed metrics have excellent practical application value.



(**a**) Risk for nodes (**b**) Risk for communities

Figure 3. The relationship between failure risk and number of attacks.



(**a**) Calculated values (**b**) Experimental values

Figure 4. Transfer probability between nodes.

In addition, considering the scale of the network and the estimated deviation of p_1 and p_2, we calculated the coefficients of correlation between the theoretical vulnerability and the actual vulnerability and summarized them in Tables 1–3. As shown in Table 1, we tested the effectiveness of evaluation metrics for nodes and communities under different network scales, and we can see that when the network scale gradually expands, our indicators fit well with the actual situation, especially for our community-based research ideas. In Tables 2 and 3, we consider the effect of the metrics when the estimated values p_1 and p_2 in the transition probability mentioned above exhibit some deviation. The majority of data in these tables exceed 0.7, signifying a robust correlation coefficient. This implies that despite potential estimation deviation, our metrics retain substantial error tolerance.

Table 1. Average coefficient of correlation for nodes and communities under different variables.

Number of Trials	Number of Nodes	Number of Simulated Attacks	Average Coefficient of Correlation for Nodes	Average Coefficient of Correlation for Communities
100	100	10,000	0.73	0.91
100	200	20,000	0.71	0.94
50	500	50,000	0.71	0.97
20	1000	100,000	0.70	0.97
20	2000	200,000	0.69	0.98

Finally, two comparative experiments are given, and the experimental data are shown in Table 4. We compared with two related models [13,20] to further substantiate our model's performance. The outcomes demonstrate that our method yields favorable results for calculating the correlation coefficients of nodes and communities within the intricate network environment. Specifically, our approach outperforms other methods in terms of nodes, and as the number of nodes progressively increases, the superiority of our model becomes particularly pronounced within the community context.

Table 2. Coefficient of correlation for nodes considering the estimated deviation of p_1 and p_2.

Coefficient of Correlation Estimated Deviation of p_2 Estimated Deviation of p_1	−5%	−4%	−3%	−2%	−1%	0%	1%	2%	3%	4%	5%
−5%	0.92	0.98	0.56	0.99	0.95	0.99	0.83	0.94	0.99	0.91	0.86
−4%	0.95	0.96	0.97	0.96	0.96	0.97	0.98	0.96	0.67	0.96	0.99
−3%	0.92	0.95	0.95	0.97	0.74	0.99	0.96	0.98	0.97	0.74	0.96
−2%	0.98	0.89	0.99	0.83	0.92	0.96	0.93	0.85	0.92	0.96	0.94
−1%	0.78	0.95	0.88	0.95	0.71	0.99	0.94	0.92	0.98	0.93	0.78
0%	0.96	0.95	0.87	0.98	0.88	0.99	0.84	0.90	0.97	0.91	0.97

Table 2. *Cont.*

Coefficient of Correlation / Estimated Deviation of p_2 / Estimated Deviation of p_1	−5%	−4%	−3%	−2%	−1%	0%	1%	2%	3%	4%	5%
1%	0.95	0.91	0.93	0.93	0.93	0.95	0.99	0.92	0.91	0.96	0.91
2%	0.96	0.90	0.88	0.93	0.95	0.96	0.92	0.90	0.94	0.98	0.98
3%	0.92	0.98	0.97	0.83	0.97	0.96	0.89	0.96	0.87	0.95	0.98
4%	0.92	0.90	0.98	0.94	0.93	0.82	0.54	0.90	0.84	0.96	0.86
5%	0.94	0.93	0.94	0.92	0.82	0.95	0.97	0.96	0.97	0.95	0.96

Table 3. Coefficient of correlation for communities considering the estimated deviation of p_1 and p_2.

Coefficient of Correlation / Estimated Deviation of p_2 / Estimated Deviation of p_1	−5%	−4%	−3%	−2%	−1%	0%	1%	2%	3%	4%	5%
−5%	0.66	0.78	0.57	0.77	0.74	0.82	0.72	0.75	0.77	0.73	0.82
−4%	0.72	0.72	0.80	0.77	0.76	0.78	0.81	0.72	0.73	0.80	0.77
−3%	0.79	0.84	0.76	0.76	0.80	0.79	0.69	0.65	0.74	0.74	0.75
−2%	0.77	0.61	0.79	0.78	0.70	0.73	0.68	0.71	0.69	0.70	0.75
−1%	0.71	0.74	0.62	0.73	0.64	0.80	0.72	0.78	0.72	0.74	0.66
0%	0.78	0.76	0.74	0.74	0.75	0.85	0.74	0.63	0.74	0.79	0.76
1%	0.74	0.73	0.77	0.74	0.71	0.70	0.78	0.74	0.78	0.76	0.76
2%	0.70	0.69	0.74	0.74	0.74	0.67	0.72	0.63	0.72	0.74	0.80
3%	0.83	0.67	0.77	0.73	0.79	0.70	0.77	0.76	0.76	0.76	0.84
4%	0.80	0.76	0.74	0.78	0.81	0.77	0.70	0.79	0.68	0.74	0.64
5%	0.78	0.77	0.72	0.49	0.65	0.81	0.79	0.80	0.63	0.72	0.78

Table 4. Average coefficient of correlation for nodes and communities for different models.

Number of Trials	Number of Nodes	Number of Simulated Attacks	Models	Average Coefficient of Correlation for Nodes	Average Coefficient of Correlation for Communities
100	100	10,000	Our model	0.73	0.91
			Model 1	0.35	0.66
			Model 2	0.65	0.98
100	200	20,000	Our model	0.71	0.94
			Model 1	0.39	0.72
			Model 2	0.60	0.98
50	500	50,000	Our model	0.71	0.97
			Model 1	0.48	0.82
			Model 2	0.52	0.97
20	1000	100,000	Our model	0.70	0.97
			Model 1	0.56	0.88
			Model 2	0.58	0.95
20	2000	200,000	Our model	0.69	0.98
			Model 1	0.55	0.92
			Model 2	0.52	0.96

6. Related Works

Security Strategy Measurement. The importance of evaluation metrics in evaluating the effectiveness of new security strategies that are constantly emerging cannot be overstated. Lingyu Wang et al. proposed a theoretical model based on zero-day security, combining the value of target assets and the shortest attack sequence to obtain k-zero-day security metrics [21]. Jin B. Hong et al. classified and proposed a series of performance metric definitions based on different characteristics of attack and defense behaviors, including attack cost, attack path exposure time, defense deployment cost, and downtime [22]. Jin B. Hong et al. also used the hierarchical attack representation model and the importance measure to evaluate the effectiveness and scalability of MTD technology [23]. Hai Jin et al. proposed a security framework that automatically senses and updates in container-based cloud environments and builds a multidimensional attack graph model to analyze attack behavior [11]. Warren Connell et al. proposed a maximizing utility function approach to capture the trade off between security and performance [24]. Luis Muñoz-González et al. modeled attack graphs and used Bayesian inference to perform static and dynamic analysis [25]. Luis Muñoz-González et al. also proposed a Bayesian-based probabilistic graphical model to estimate the vulnerability and interconnection of system components and calculate the attack probability of target nodes to determine security [26]. Mengyuan Zhang et al. evaluated the network diversity based on the effective quantity of different resources, and the minimum and average attack effort, respectively [27]. These studies typically conduct evaluations from a global perspective. However, due to the significant asymmetry in network attacks and defenses, especially in complex networks, they are unable to identify specific regions where security benefits can be obtained.

Cyber Mimic Defense Measurement. As research on cyber mimic defense unfolds, how to evaluate the effectiveness of CMD deployment becomes a key issue. Fei Yu et al. conducted a series of experiments on basic, common-mode, and differential-mode attacks to obtain the defense success rate and analyzed the delay and throughput to reflect their performance loss [28]. Congqi Shen et al. proposed a decentralized multi-adjudicator arbiter approach to determine the defense effectiveness using the consistent convergence of subarbiters after data injection attacks [29]. Quan Ren et al. analyzed the applicability of cyber mimic defense in a software-defined network from the aspects of availability, response time, compromise tolerance, and performance [30]. Haiyang Yu et al. studied the effect of cyber mimic defense in a distributed system from the aspects of data reliability, fault repair, and security [31]. Chen Yu et al. analyzed the security and effectiveness of mimic DAA scheme [32]. Wei Liu et al. evaluated the mimic defense strategy in terms of storage limitation, throughput, and algorithm speed [33]. Yufeng Zhao et al. constructed a security quantification model from multiple angles, analyzed the different characteristics of cyber mimic defense architecture, and achieved a relatively complete security quantification method [34]. These studies primarily focus on measuring the security of cyber mimic defense system itself, and there is currently a lack of research on evaluating the effectiveness of deploying cyber mimic defense in large-scale networks.

7. Conclusions

In this paper, we propose a series of cyber mimic defense evaluation metrics by partitioning the complex network with the idea of the Louvain algorithm and mapping it to the objective space for finer-grained evaluation, incorporating common-mode index, transfer probability, and failure risk. Numerous simulation results demonstrate that our proposed metrics are highly reliable and can accurately reflect the effectiveness of cyber mimic defense technology deployed in complex networks. In future research, we will further refine the metrics for cyber mimic defense and integrate them with real-world scenarios. We believe that this work will inspire researchers in related fields and contribute to the improvement of the cyber mimic defense measurement.

Author Contributions: Conceptualization, Z.H. and Y.Y.; Methodology, Z.H., Y.Y., J.F. and H.Z.; Software, Z.H. and J.F.; Validation, Z.H., J.F. and H.Z.; Formal analysis, Z.H. and Y.Y.; Investigation, Z.H. and H.Z.; Resources, Z.H., Y.Y., J.F. and J.H.; Data curation, Z.H., Y.Y. and H.Z.; Writing—original draft, Z.H., Y.Y. and H.Z.; Writing—review & editing, Z.H., Y.Y. and J.H.; Visualization, Z.H.; Supervision, Y.Y.; Project administration, G.C.; Funding acquisition, G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (Grant No. 2020YFB1804604) and the Special Funds for Basic Scientific Research Operations of Central Universities (Grant No. 2242023k30053).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Zheng, Y.; Li, Z.; Xu, X.; Zhao, Q. Dynamic defenses in cyber security: Techniques, methods and challenges. *Digit. Commun. Netw.* **2022**, *8*, 422–435. [CrossRef]
2. Yang, J.; Chen, X.; Chen, S.; Jiang, X.; Tan, X. Conditional variational auto-encoder and extreme value theory aided two-stage learning approach for intelligent fine-grained known/unknown intrusion detection. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 3538–3553. [CrossRef]
3. Yousef, W.A.; Traoré, I.; Briguglio, W. UN-AVOIDS: Unsupervised and Nonparametric Approach for Visualizing Outliers and Invariant Detection Scoring. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 5195–5210. [CrossRef]
4. Tian, W.; Du, M.; Ji, X.; Liu, G.; Dai, Y.; Han, Z. Honeypot detection strategy against advanced persistent threats in industrial internet of things: A prospect theoretic game. *IEEE Internet Things J.* **2021**, *8*, 17372–17381. [CrossRef]
5. Giraldo, J.; El Hariri, M.; Parvania, M. Decentralized Moving Target Defense for Microgrid Protection against False-Data Injection Attacks. *IEEE Trans. Smart Grid* **2022**, *13*, 3700–3710. [CrossRef]
6. Hu, Y.; Xun, P.; Zhu, P.; Xiong, Y.; Zhu, Y.; Shi, W.; Hu, C. Network-based multidimensional moving target defense against false data injection attack in power system. *Comput. Secur.* **2021**, *107*, 102283. [CrossRef]
7. Sengupta, S.; Chowdhary, A.; Sabur, A.; Alshamrani, A.; Huang, D.; Kambhampati, S. A survey of moving target defenses for network security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1909–1941. [CrossRef]
8. Negi, P.S.; Garg, A.; Lal, R. Intrusion detection and prevention using honeypot network for cloud security. In Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 29–31 January 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 129–132.
9. Wu, J. *Cyberspace Mimic Defense*; Springer: Berlin/Heidelberg, Germany, 2020.
10. Zheng, J.; Wu, G.; Wen, B.; Lu, Y.; Liang, R. Research on SDN-based mimic server defense technology. In Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science, Wuhan, China, 12–13 July 2019; pp. 163–169.
11. Jin, H.; Li, Z.; Zou, D.; Yuan, B. Dseom: A framework for dynamic security evaluation and optimization of mtd in container-based cloud. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 1125–1136. [CrossRef]
12. Li, H.; Hu, J.; Ma, H.; Huang, T. The architecture of distributed storage system under mimic defense theory. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2658–2663.
13. Picek, S.; Hemberg, E.; O'Reilly, U.M. If you can't measure it, you can't improve it: Moving target defense metrics. In Proceedings of the 2017 Workshop on Moving Target Defense, Dallas, TX, USA, 30 October 2017; pp. 115–118.
14. Barabási, A.L. Scale-free networks: A decade and beyond. *Science* **2009**, *325*, 412–413. [CrossRef] [PubMed]
15. Fortunato, S. Community detection in graphs. *Phys. Rep.* **2010**, *486*, 75–174.
16. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
17. Cohen-Addad, V.; Kosowski, A.; Mallmann-Trenn, F.; Saulpic, D. On the power of louvain in the stochastic block model. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 4055–4066.
18. Fortunato, S.; Lancichinetti, A. Community detection algorithms: A comparative analysis: Invited presentation, extended abstract. In Proceedings of the 4th International ICST Conference on Performance Evaluation Methodologies and Tools, Pisa, Italy, 20–22 October 2009.
19. Shameli-Sendi, A.; Louafi, H.; He, W.; Cheriet, M. Dynamic optimal countermeasure selection for intrusion response system. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 755–770. [CrossRef]
20. Yang, S.; Chen, W.; Zhang, X.; Liang, C.; Wang, H.; Cui, W. A graph-based model for transmission network vulnerability analysis. *IEEE Syst. J.* **2019**, *14*, 1447–1456. [CrossRef]
21. Wang, L.; Jajodia, S.; Singhal, A.; Cheng, P.; Noel, S. k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. *IEEE Trans. Dependable Secur. Comput.* **2013**, *11*, 30–44. [CrossRef]
22. Hong, J.B.; Enoch, S.Y.; Kim, D.S.; Nhlabatsi, A.; Fetais, N.; Khan, K.M. Dynamic security metrics for measuring the effectiveness of moving target defense techniques. *Comput. Secur.* **2018**, *79*, 33–52. [CrossRef]
23. Hong, J.B.; Kim, D.S. Assessing the effectiveness of moving target defenses using security models. *IEEE Trans. Dependable Secur. Comput.* **2015**, *13*, 163–177. [CrossRef]
24. Connell, W.; Menasce, D.A.; Albanese, M. Performance modeling of moving target defenses with reconfiguration limits. *IEEE Trans. Dependable Secur. Comput.* **2018**, *18*, 205–219. [CrossRef]
25. Muñoz-González, L.; Sgandurra, D.; Barrère, M.; Lupu, E.C. Exact inference techniques for the analysis of Bayesian attack graphs. *IEEE Trans. Dependable Secur. Comput.* **2017**, *16*, 231–244. [CrossRef]

26. Muñoz-González, L.; Sgandurra, D.; Paudice, A.; Lupu, E.C. Efficient attack graph analysis through approximate inference. *arXiv* **2016**, arXiv:1606.07025.
27. Zhang, M.; Wang, L.; Jajodia, S.; Singhal, A.; Albanese, M. Network diversity: A security metric for evaluating the resilience of networks against zero-day attacks. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1071–1086. [CrossRef]
28. Yu, F.; Wei, Q.; Geng, Y.; Wang, Y. Research on Key Technology of Industrial Network Boundary Protection based on Endogenous Security. In Proceedings of the 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 18–20 June 2021; IEEE: Piscataway, NJ, USA, 2021; Volume 4, pp. 112–121.
29. Shen, C.; Chen, S.X.; Wu, C.M. A Decentralized Multi-ruling Arbiter for Cyberspace Mimicry Defense. In Proceedings of the 2019 International Symposium on Networks, Computers and Communications (ISNCC), Istanbul, Turkey, 18–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
30. Ren, Q.; Hu, T.; Wu, J.; Hu, Y.; He, L.; Lan, J. Multipath resilient routing for endogenous secure software defined networks. *Comput. Netw.* **2021**, *194*, 108134. [CrossRef]
31. Yu, H.; Li, H.; Yang, X.; Ma, H. On distributed object storage architecture based on mimic defense. *China Commun.* **2021**, *18*, 109–120. [CrossRef]
32. Yu, C.; Chen, L.; Lu, T. A Direct Anonymous Attestation Scheme Based on Mimic Defense Mechanism. In Proceedings of the 2020 International Conference on Internet of Things and Intelligent Applications (ITIA), Zhenjiang, China, 27–29 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
33. Liu, W.; Peng, Y.; Tian, Z.; Li, Y.; She, W. A Medical Blockchain Privacy Protection Model Based on Mimicry Defense. In Proceedings of the International Conference on Artificial Intelligence and Security, Hohhot, China, 17–20 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 581–592.
34. Zhao, Y.; Zhang, Z.; Tang, Y.; Ji, X. A Security Quantification Method for Mimic Defense Architecture. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; IEEE: Piscataway, NJ, USA, 2021; Volume 5, pp. 36–40.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.