

Proactive Defense Mechanism: Enhancing IoT Security through Diversity-based Moving Target Defense and Cyber Deception*

Ms Zubaida Rehman^{a,*1}, Iqbal Gondal^{b,2}, Mengmeng Ge^{c,3}, Hai Dong^{d,4}, Mark A Gregory^{e,5} and Zahir Tari^{f,6}

^aRMIT University, Australia

^bRMIT University, Australia

^cUniversity of Canterbury, New Zealand

^dRMIT University, Australia

^eRMIT University, Australia

^fRMIT University, Australia

ARTICLE INFO

Keywords:

Internet of Things

Moving Target Defense

Graphical Security Models

Diversity


ABSTRACT


The Internet of Things (IoT) has become increasingly prevalent in various aspects of our lives, enabling billions of devices to connect and communicate seamlessly. However, the intricate nature of IoT connections and device vulnerabilities exposes the devices to security threats. To address the security challenges, we propose a proactive defense framework that leverages a model-based approach for security analysis and facilitates the defense strategies. Our proposed approach incorporates proactive defense mechanisms that combine Moving Target Defense techniques with cyber deception. The proposed approach involves the use of a decoy nodes as a deception technique and operating system based diversity as a moving target defense strategy to change the attack surface area of IoT networks. Additionally, we introduce a technique known as Important Measure-based Operating System Diversity to reduce defense cost. The effectiveness of the defense mechanisms was evaluated by using a graphical security model in a Software Defined Networking-based IoT network. Simulation results demonstrate the effectiveness of our approach in mitigating the impact of attacks while maintaining high performance levels in IoT networks.

1. Introduction

The Internet of Things (IoT) is an emerging paradigm due to its ability to provide intelligent connectivity and applications across various domains. It enables the connection of numerous objects with different service requirements in distributed networks, allowing for ubiquitous connectivity. However, this scalability also leads to increased complexity in management and poses challenges in ensuring cybersecurity [58]. To address these challenges, Software Defined Networking (SDN) presents a novel approach by decoupling the control plane and data plane in IoT networks [33]. This provides the SDN controller with a global network view, allowing for flexible traffic engineering and improved IoT security. Despite SDN's ability to manage IoT networks flexibly and enhance network and data security, its architecture also introduces vulnerabilities that increases the overall risk in the environment [24]. One such vulnerability arises from the fact that legacy and limited resource devices cannot be managed by the SDN controller,

*This document is the results of the research project funded by the CloudTech.

 s3929941@student.rmit.edu.au (Z. Rehman); iqbal.gondal@rmit.edu.au (I. Gondal); mge43@uclive.ac.nz (M. Ge); hai.dong@rmit.edu.au (H. Dong); mark.gregory@rmit.edu.au (M.A. Gregory); zahir.tari@rmit.edu.au (Z. Tari)

 rmit.edu.au (Z. Rehman); rmit.edu.au (I. Gondal); uclive.ac.nz (M. Ge); rmit.edu.au (H. Dong); rmit.edu.au (M.A. Gregory); rmit.edu.au (Z. Tari)

ORCID(s):

as they operate outside of the SDN architecture. Some IoT devices, especially those with limited processing power, memory, or energy resources, may not be suitable for direct management through SDN. SDN requires additional software components and overhead, which may be impractical or inefficient for resource-constrained devices. The nature of IoT devices means that data is constantly being transmitted, processed and collected in the cloud, often without any encryption. If a hacker was able to access a medical IoT device, they could use it to manipulate information and transmit false signals. If a healthcare practitioner acts on one of these signals then it could have a significant impact on the patient's treatment. Research conducted by the FDA [26] found that St Jude Medical's implantable cardiac devices have vulnerabilities. If hackers were able to gain access then they could deplete the battery or administer incorrect pacing or shocks. Another example given by the Chief Security Strategist at the PRPL Foundation was about the Owlet WiFi baby heart monitors transmitting unencrypted traffic to the base station without any authentication. This could leave the devices vulnerable to any eavesdropping attacks. The lack of necessary security protocols can make it easy for attackers to infect IoT devices, form botnets, and launch DDoS attacks. Moreover, network unavailability [57] can have catastrophic consequences in certain situations since numerous IoT applications rely on real-time inputs.

Traditional network configurations are often deterministic, static, and uniform, placing defenders in a passive position and making countermeasures costly and short-lived. In recent times, the Moving Target Defense (MTD) [60] cybersecurity technique has gained prominence because it can be used to reduce cyber attacks by introducing uncertainty and disrupting the Cyber Kill Chain (CKC) [43]. Cyber deception techniques [65] have proven effective by misleading attackers with false information, decoys and honeypots [21]. Existing solutions for IoT network security [55] have limitations; e.g., utilizing a single technology approach that is focused on intrusion detection [61] or MTD [52]. Combining defense mechanisms is often overlooked as a means to enhance cybersecurity and to limit risks associated with one of the defense mechanisms being compromised. Current approaches may not deter sophisticated attackers who can monitor traffic passively and pinpoint real attack targets. Frequent adaptations such as Internet Protocol (IP) address shuffling [69] or Virtual Machine (VM) migration [5] increase costs, response delay [64], and reduce service quality. Therefore, balancing proactive techniques for cybersecurity with performance and cost is essential.

This paper employs a diversity-based MTD technique that enhances security by periodically changing system components to confront attackers with evolving vulnerabilities. The novel and innovative technique introduces effective security metrics, including Attack Cost (AC), Return on Attack (RoA), and Risk (R), from both attacker and network defense perspectives. To address scalability associated with graphical security models (GSMs), we adopt the Hierarchical Attack Representation Model (HARM) [37]. The aim of the proposed technique is to proactively defend IoT network using a combination of cyber deception and MTD techniques within an SDN-based IoT framework. While MTD can be implemented traditionally using hardware-based middleboxes, this paper explores an approach tailored to SDN, offering programmability and controllability. The contributions of this paper include:

- Development of an integrated proactive cybersecurity technique for an IoT network by implementing the diversity-based MTD technique on a network that comprises both decoy and real nodes.
- Reduced cost to enhance IoT security.
- The application of Important Measures (IMs) on various Network Centrality Measure (NCM) properties to assess MTD technique effectiveness with a focus on IoT networks.
- Comprehensively analyzing security and performance metrics that include attack paths, AC, system risk, RoA, and Extra Operational Cost (EOC).
- Development of a framework and advanced GSM modeling technique that can be used to evaluate cybersecurity techniques.
- Testing and validation of the proposed cybersecurity technique under diverse scenarios, ensuring robustness and real-world applicability.

The rest of this paper is organized as follows. Section 2 provides a brief overview of the related work. Section 3 gives an overview of the proposed framework and network model. Section 4 includes the design of our proposed defense mechanism. Section 5 shows simulation results and analyses the results observed. Section 6 suggests future research directions and provides the conclusion.

2. Related Work

Due to the limited resources and heterogeneous nature of end-devices, as well as the emergence of novel protocols and networking technologies, providing security in IoT networks poses a significant challenge [17]. However, the literature shows that research in IoT security is rapidly progressing [35], with intrusion prevention, detection, and mitigation [14] being the primary solutions to defend against security attacks [19]. Proactive defense technologies, such as those employing SDN, are gaining more attention due to their flexibility and ability to mitigate attacks against IoT devices [14],[48]. Therefore, we provide a brief discussion of related techniques, including MTD and cyber deception, along with a review of diversity based MTD techniques for cloud and IoT security in this section.

Cyber Deception and MTD approaches for IoT. MTD is a proactive defense mechanism that aims to increase the dynamic nature of a system to deter attacks. It achieves this by employing techniques such as shuffling, diversity, and redundancy [4]. [Shuffling techniques \(e.g., changing IP addresses; migrating virtual machines\), are used to confuse the attackers during the reconnaissance phase. This increases the attack difficulty and efforts, and invalidates the previously acquired system intelligence \[21\].](#) The authors of [75] conducted a study on MTD strategies and analysed how these techniques affect the diversity, unpredictability, and vulnerability of a system when defending against cyber-attacks. They found that applying various MTD approaches could assist in reducing the risk of reconnaissance attacks, computer worm assaults, distributed denial-of-service attacks, and code injection attacks. The reason MTD is effective is because it introduces diversity and random changes to networks and systems, which might make the knowledge that attackers gathered during the reconnaissance phase invalid. [18] states that three MTD subfields were studied: theory, strategy, and evaluation. The selection of an MTD technique and how it will be applied depends on when it is appropriate to apply the technique but the cost factor was not taken into consideration. In [46] it was suggested the technique Micro One Time Address be used to anonymize packet flows for IoT devices. This method involves altering the structure of IPv4 packets, but it requires reconfiguration of all routers due to the change in IP header. In [54] a simpler approach was proposed for changing the addresses of IoT devices by utilizing network-wide address shuffling. This involves sending a multicast message to prompt devices to shuffle their addresses. Additionally, in [6] the authors introduced a model to prevent attackers from discovering device addresses in IoT networks. This method involves transmitting data through a dedicated MTD channel with the use of fake addresses. The MTD technique, developed by the authors of [41], is a method for pro-actively altering host IP addresses. [2] introduced the concept of Random Route Mutation (RRM) to determine an optimal randomized path between source and target. Meanwhile, [8] developed a technique called Shuffle, which employs IP randomization to counter Hit-List worm attacks. [71] suggested a hypervisor-level end-to-end defense mechanism to safeguard VMs in a cloud data center. The authors of [73] proposed a MTD approach to address the challenges associated with co-residency in a virtualized environment. The impact of MTD techniques on security was evaluated by [37] using a formal method and also analysed how each technique affected security. In [5],[3], a combination of two MTD techniques were proposed; shuffle and diversity, which minimized attacks on the cloud systems. Based on the results obtained, it was concluded that combining both shuffle and diversity can help to improve security metrics. In [4] the authors devised the combination of shuffling and a redundancy technique, where shuffling improves the security of cloud networks and systems and redundancy helps to improve the reliability of the cloud.

In [49], an SDN-based architecture was implemented, which enables cyber deception on legacy IP-based IoT devices. The authors achieved this by using SDN-enabled honeypots to collect attack information and communicating with SDN controllers to update flow rules on switches to reduce attack traffic. Although cyber deception has proven effective in disrupting early stages of the CKC, the complexity of honeypots and programming in large systems often leads to reduced attacker intelligence and attention span [68]. In [34] a honeypot-like approach using fake and real gateways was proposed that utilized sensors to identify cyber-attacks and to deceive attackers. [67] presented a game theoretic approach that minimized the extra operations required by Markov gaming while dominating the game based on a Zero-determinant (ZD) strategy. Other studies, such as [7] and [23], have used honeypots to mitigate DDoS attacks launched from IoT devices and to identify DDoS attacks and bot malware using a ZigBee honeypot. The high cost of implementing and maintaining high-interaction honeypots limits their scalability for gaining in-depth information about attacks and attacker behaviour [62]. To address these challenges, recent developments in IoT environments have seen the integration of visualization and automation technologies to facilitate decoy deployment and updates. However, attackers can still identify decoys based on longer response times, and it remains a challenge to develop deception techniques in production environments while ensuring continuous network monitoring and safe

deployment without compromising system integrity [40]. In contrast to traditional deception-based methods, the proposed approach involves flexible adjustments to network properties to disguise production systems and decoys, saving defense resources.

Diversity-based approaches. Diversity-based MTD approaches have been applied to traditional networks, cyber physical systems, and the cloud. According to [21], attack complexity can be increased by the use of diversity-based MTD techniques. This is achieved through the incorporation of multiple system components (such as software and Operating Systems (OS)) that offer equivalent functionalities. Another study in [39] presented a Diversity MTD approach for virtual servers to improve network and service resilience. They achieved this by modifying the OS, visualization components, web servers, and application software. The researchers then assessed the effectiveness of their technique by analysing the probability of successful attacks. In [11], the authors developed a different Diversity MTD approach that involves randomly altering program variants during runtime. Their proposed approach divides a large program into smaller sections (cells or tasks) that can be executed using multiple variants with the same functionality. Different randomization techniques have been used to automatically generate diversity [63]. Those techniques can be applied to improve the network diversity measured using their metric model. The metrics identified provides a quantitative evaluation method. The formalization of diversity among redundant subsystems in smart grids is presented in [25]. Apart from design and generated diversity, opportunistic diversity, which already exists among various software systems, has also been utilized in recent research. For instance, [28] evaluates the feasibility of using OS diversity for intrusion tolerance. In [72] the authors have adapted biodiversity metrics to networks and extended the diversity metrics to the network level. While these diversity techniques serve as the basis for their research, they do not offer a systematic solution for enhancing network diversity. By modifying system components, diversity MTD techniques can complicate attacks and make them more challenging. This is because changing a component can introduce a fresh set of vulnerabilities, rendering the attacker's existing knowledge of vulnerabilities obsolete. As a result, attackers may need to invest more time, effort, and money to develop new techniques for exploiting the newly introduced vulnerabilities. Diversity-based techniques provide functionally equivalent applications with different implementations, such as code diversification [50] and instruction set randomization [44]. Redundancy-based techniques create replicas of applications [74] or services [42] with the same functionality to increase resilience to attacks. [15] used the network diversity optimization for resilience against unknown attacks in cloud environment and [20] used formal modeling network diversity (as a security metric for evaluating networks robustness against zero day attack) and [16] used OS diversity for intrusion detection in SCADA environment. Hybrid techniques combine multiple MTD mechanisms to work in cooperation. However, frequent MTD mutations can negatively impact system performance, and additional deployments may not be suitable for IoT devices due to their limited computational capabilities. Nevertheless, a proposed lightweight framework can be easily deployed on top of an SDN controller and avoid unnecessary defense costs while decreasing overhead. [Table 1 provides a comparative summary of well-known cybersecurity techniques in the literature. The cybersecurity techniques are compared based on their application of Moving Target Defense \(MTD\), Deception, and Intrusion Detection/Prevention techniques.](#)

SDN for IoT. SDN technologies have attracted attention in the context of managing IoT traffic flows due to its ability to be easily programmed and to optimize network performance. By employing an SDN-based architecture and network function virtualization, challenges faced in IoT environments can be effectively addressed, and devices can be made interoperable. In one specific study [47], the focus was on countering man-in-the-middle attacks that target the OpenFlow control channels, which are used in SDN. The proposed solution utilized a Bloom filter, a probabilistic data structure, to detect and mitigate attacks. Additionally, by implementing SDN-based networking in an IoT environment, traffic routing and energy consumption can be optimized, leading to more efficient network operation. Another study [66] introduced a routing protocol for SDN-based sensor networks that enables multihop communication. This protocol utilizes a centralized control approach, where an SDN controller manages the network. This centralized control allows for efficient multitasking in the sensor networks, facilitating the execution of multiple sensing tasks simultaneously [70]. However, a remaining challenge in this context is the design of an optimal management strategy for the sensor nodes. SDN solutions [27] have been used in various IoT networks for different purposes, including managing data flow between IoT devices, reducing data exchange in wireless sensor networks, managing wireless access networks, optimizing mobile networks [13], enabling smart urban sensing, and aiding in topology reconfiguration decision making in wireless sensor networks [22]. [12] conducted a comprehensive survey to categorize the different techniques for detecting and mitigating DDoS attacks using SDN. They also presented ProDefense, a proactive DDoS Framework based on SDN, that can detect and mitigate DDoS attacks in a large-scale network. [76] proposed an SDN-enabled

defense framework by combining MTD and cyber deception to create fake information to confuse attackers. Qin et al. [56] proposed a software-defined approach for IoT that dynamically achieves varying levels of quality in a diverse wireless network. [45] examined the possibility of gradually implementing SDN-based solutions alongside the current BGP-based Internet infrastructure.

Graphical Security Modelling for IoT. A graphical security model used to represent and analyze system vulnerabilities in a graphical format. HARM is a two-layer model that combines Attack Graphs (AG) and Attack Trees (AT) to capture network reachability and vulnerability information. According to [38], graphical security models provide an efficient way to evaluate system vulnerabilities by applying defense strategies. An AG displays possible attack sequences that can reach the target based on the vulnerability information and links between devices. However, AG scalability is limited as the network size grows. An AT is another graphical security model that systematically presents potential attacks in the network, but it also suffers from scalability issues. To address this problem [37] introduced a two-layer HARM that combines graphical security models on different layers. The upper layer (AGs) captures the network reachability information, while the lower layer (ATs) represents the vulnerability information of each node in the network. The HARM layers were constructed independently of each other, which reduced the computational complexity of calculating and evaluating the model compared to single-layered graphical security models. [37] evaluated MTD techniques in a virtualized system based on HARM by using a risk metric. [31] developed a framework using HARM to automate security analysis of IoT networks, by evaluating defense mechanisms at both device and network levels based on cost and impact metrics. Several studies have utilized a risk-based security approach to evaluate the effectiveness of defense mechanisms in IoT environments. The authors of [31] studied the effectiveness of address space layout randomization (ASLR) and assessed its performance through the utilization of HARM. In [59] a framework was proposed that adopts a game-theoretic approach and context-aware techniques to assess the expected risk and potential benefits in eHealth IoT domains and also an adaptive security management scheme considers security metrics to address the challenges of securing eHealth IoT environments. [58] proposes a method to estimate risk metrics from an economic perspective and devises an optimal security resource allocation plan for an IoT network composed of mobile nodes.

The studies discussed earlier focused on either MTD or cyber deception. [30] proposes using both shuffling based MTD and cyber deception, but none of the MTD-based approaches for IoT considered OS diversity-based techniques integrated with a defense model to effectively halt attacks that use compromised IoT devices as steppingstones. Additionally, a review of the literature was unable to identify a proposal to develop an integrated defense system that combines both MTD and defensive deception techniques. When decoys are deployed, MTD using network diversity not only confuses attackers by changing the OS among IoT devices but also makes the network more complicated and steers attackers away from actual IoT devices. This can increase the cost and effort required for an attack while decreasing the likelihood of genuine IoT devices being compromised. Therefore, we propose a proactive defense system that integrates both cyber deception and MTD techniques to prevent intrusions and effectively mitigate the negative impact of attackers before they can infiltrate an IoT system. There is work on combining honeypots with diversity, however, when compared with [10], the novelty of our proposal is that we use recently developed deception technologies and OS diversity and aim to explore the effectiveness of this approach and trade-off between the operational cost and diversity of both decoys and real devices for IoT networks. [Table 2 summarizes state-of-the-art, outlining methodologies proposed, simulation scenarios, attack models, and evaluation metrics.](#)

3. Proposed Framework and System Model

In this section, we describe the proposed framework, the network model with an example SDN-based IoT environment, the attacker's goals and abilities, the strategies to deploy our defense mechanism and the security parameters used for our research to evaluate performance.

3.1. Comparison with related works

In [30], the authors proposed a cyber security defense mechanism that was based on shuffling MTD and cyber deception. This work provided motivation to explore a proactive defense technique that proposes use of OS diversity. The earlier work reported in [10] includes honeypots with diversity and the outcomes of the research highlight the benefits of combining cyber security techniques. In [16], the authors employed diverse operating systems to create an intrusion detection system for SCADA systems. The proposed cyber security framework is novel and innovative

because we propose to utilize the combination of OS diversity and deception techniques to implement a more effective defense mechanism against cyber attacks for IoT networks. The combination of cyber security defense techniques utilized in the proposed model is new and best of our knowledge, such a combination of techniques has not been found in the literature.

Tables 1 and 2 provides a comparison of related works found in the literature. The research works explore the use of a combination of MTD and deception applications that can be used to carry out intrusion detection and prevention. The simulation scenarios used in these works include SCADA environments [16], cloud networking devices and IoT networks. A range of evaluation metrics have been employed to measure the effectiveness of the approach being proposed. The novel approach presented here provides enhanced intrusion prevention by employing OS diversity (MTD) and dynamic decoys (Deception) in a smart SDN enabled defense framework to confuse the attackers by presenting fake information.

Regarding the evaluation metrics, [16] used Detection Rate, False Positive Rate, System Overhead and focused on the performance of a security system in terms of detection accuracy, the reduction of false positives, and the impact on system resources. [31] employed ASP, MTTC, Attack Impact to address the broader cyber security concerns related to the management of attack surfaces, response times, and the severity of security incidents. [10] used Deception Rate, Detection Rate, Cost Per Honeypot Type which are specific to deception-based security strategies, such as honeypots, and evaluates the effectiveness, visibility, and cost-effectiveness of these strategies. [15] employed D1 (Gain Based on Number of Exploits), D2 (Gain Based on Shortest Path) and D3 (Gain Based on Number of variable) and focused on reducing the number of successful exploits to improve the security while reducing the number of variables or factors that can be manipulated by the attackers. [20] used Worm propagation and attack success rate to assess the speed and reach of an attack, and measure the effectiveness of individual attacks. [76] used Survival rate and Packet loss rate and focused on the percentage of successful deliveries, while the packet loss rate quantifies the extent of data loss. [30] MTTC focuses on attack response times, MTTSF assesses the reliability of security controls, and defense cost measures the financial investment in security measures. In our paper, we have used Attack Cost and System Risk that focus on assessing the exploitation efforts and vulnerability of an organization's security posture. ROA is a financial metric used to evaluate the probability and profitability of the attack. PDR is a performance metric used to measure the successful delivery of data packets in a network, reflecting the quality and reliability of data transmission.

Table 1
Comparison Among State-of-the-Art Based on MTD and IDS/IPS.

Paper	Applications of MTD	Applications of Deception	Intrusion Detection/Prevention
[1]	N/A	Decoy farm	Both
[9]	N/A	Honeypots	Both
[32]	N/A	N/A	Intrusion prevention
[15]	Network diversity	N/A	N/A
[20]	Dynamic network diversity	N/A	Intrusion detection
[31]	N/A	N/A	Intrusion detection
[30]	Network topology shuffling, IP address shuffling	Decoy	Both
[76]	Hybrid	Honeypot	Intrusion detection
[16]	OS diversity	N/A	Intrusion detection
[10]	Software Diversity	Honeypot	Intrusion detection
Our	OS diversity	Decoys	Intrusion prevention

3.2. Proposed Framework

The proposed framework presented in Figure 1 comprises of five stages to implement and evaluate the effectiveness of the operating system diversity technique:

Table 2
Comparison Among State-of-the-Art Based on Methodology and Attack Model.

Paper	Proposed Methodology	Simulation Scenario	Attack Model	Evaluation Metrics
[1]	Induction of threat hunting in SCADA	SCADA environment	Hatman Malware, HTTP Dos, Headless HTTP Server	Threat Detection, validation
[9]	Game theory based honeypot allocation algorithm over an attack graph	Virtualized network	Attack graph	Game-theoretic metrics
[32]	SQL Database Architecture design based on OS vulnerabilities	SQL database	Statistical analysis	Attack vector, access complexity, authentication, vulnerability integrity
[15]	Optimization for resilience against unknown attacks by using heuristic and optimization algorithm	Cisco cloud data centre and Openstack	Resources attack graph	D1, D2, D3
[20]	Formal modeling network diversity as a security metric for evaluating networks robustness against zero day attacks	Stuxnet and SCADA	Zero day attack	Worm propagation, success rate of attack
[31]	Graphical security model SHARPE based security assessment	Zigbee based Smart home scenario	N/A	ASP, MTTC, Attack Impact
[30]	An SDN-based network topology shuffling and optimal honeypot placement defense strategy	Smart hospital	APT	MTTC, MTTSF, Defense Cost
[76]	An SDN-Enabled defense framework by combining MTD and cyber deception to create fake information to confuse attackers	N/A	DDoS	Survival rate, average packet loss rate
[16]	Use of diverse Operating Systems to create a host-based intrusion detection model for SCADA systems	Simulated SCADA environment	Unknown threats	Detection rate, False Positive rate, system overhead
[10]	Utilizing game theory to optimally allocate and diversify honeypots to deceive attackers by considering the network topology as well and importance of the nodes	Node Tree	Game Model	Deception rate, detection rate, cost per honeypot type
Our	An SDN-Enabled defense framework by combining OS-based MTD and cyber deception to create fake information to confuse attackers	Smart hospital	Reconnaissance, data exfiltration	Attack cost, System risk, ROA, PDR

Stage 1. The User provides relevant system information, such as initial network topology and node vulnerability data, to the IoT Generator in order to create the IoT network architecture. The output is then passed to the Diversity Deployment Module.

Stage 2. In this stage, the initial deployment of the decoy-based IoT network associated with the real IoT network is developed. Initially, it is proposed to deploy one decoy node for each of the real nodes, in the VLANs. Table 3 shows the different types of the nodes used in the network. We consider four attribute categories: a node is either real ($n_i.r=0$) or a decoy ($n_i.d=1$), compromised or not, critical with essential information or not with range 0 and 1 respectively.

Stage 3. The Diversity Deployment Module generates the (real and decoy based) IoT network. The IoT network changes the OSs for the nodes as per proposed technique along with updating vulnerability information in random and IMS-based OS-diversity fashions. In the case of random diversity, the Diversity Deployment Module applies OS-diversity on randomly selected nodes based on a percentage of all nodes and in the case of IMS-based OS-diversity, the Diversity Deployment Module selects critical nodes first and then applies OS-diversity and feeds the IoT network with an updated OS and passes their vulnerabilities to the Security Model Generator for further processing.

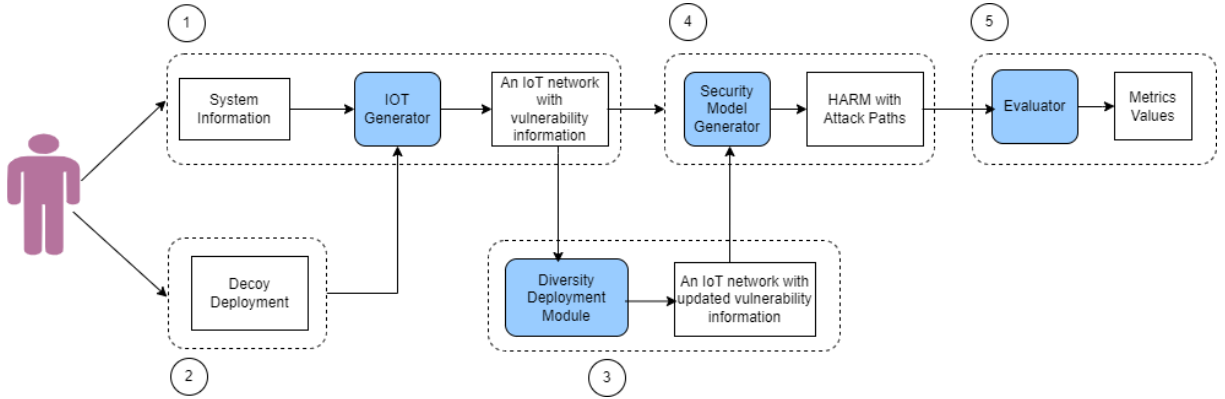


Figure 1: The Proposed Framework

Stage 4. The Security Model Generator automatically generates a HARM value based on both the initial IoT network and the updated OS, which depicts all of the attack paths. Our HARM model comprises two layers. The upper layer represents the reachability information, while the lower layer represents the vulnerability information of each node.

Stage 5. The Evaluator uses the HARM model output, which is a graphical security model to calculate the results for the given evaluation metrics.

3.3. System Model

In this section we describe our system model including the network model, attack model, and defense model in an IoT environment.

Network Model. In Stage 1, the network model having details about network architecture, components, and infrastructure in the SDN-based IoT environment is used to create an IoT network. In this paper, we consider a smart hospital as the sample IoT network, containing N real nodes composed of servers and IoT nodes. The IoT nodes collect data and transmit it periodically to servers through one or multiple hops. The path from an IoT node to a server node is a sequence of nodes, consisting of intermediate nodes. In the network, IoT nodes of distinct functions and servers are situated across various VLANs. The study assumes that SDN technology [27, 22] is utilized to manage and regulate traffic flows effectively among the nodes. The IoT network used an SDN controller located on a remote server, which interacts with SDN enabled switches and manages the traffic flows between the IoT nodes and servers connected to switches through communication channels such as cables or wireless signals. In this paper, we use the network model of a Smart hospital but as medical devices are expensive it is not yet realistic to apply diversity to all medical IoT devices. For that reason, in this work, we assume that all the devices are compatible with OS diversity.

Table 3
Node Types

Node Type	Attributes	Value Range
Real Node	$n_i.r$	0, 1
Decoy Node	$n_i.d$	0, 1
Compromised Node	$n_i.c$	0, 1
Critical Node	$n_i.r$	0, 1

Table 3 presents the node types and the attributes used in the proposed model. We consider four attribute categories: a node is either real or a decoy, compromised or not, critical with essential information or not and a list of vulnerabilities is attached to each node.

Attacker Model. In order to compute attack paths using a Security Model Generator in Stage 4, we provide the attack model as an input. The attack model typically includes information about potential vulnerabilities, threats, and attack

vectors that could be exploited by an attacker. The attack model is presented in Table 4. To describe attackers, we have made assumptions about their behaviors including:

- Attackers have limited knowledge about decoy nodes, which are fake nodes designed to mimic real nodes. The attacker's ability to detect decoys depends on the difference between the deployed decoys and the real nodes. We measure the attacker's capabilities in detecting decoys by how often they interact with the decoy node.
- Attackers will terminate interactions with decoys immediately if they realize their existence has been detected and they will try to find new targets to attack.
- The attacker's ultimate goal is to leak confidential information to unauthorized entities outside the IoT network by compromising servers.
- Attackers can identify unpatched vulnerabilities and exploit them using scanning and exploitation tools.
- Attackers are unlikely to compromise servers directly and will instead use vulnerable IoT nodes as entry points to move laterally within the network and eventually compromise servers. Servers are typically better protected with multiple security measures such as firewalls, intrusion detection systems, and regular security updates, under the defense in depth strategy. It can be more challenging for attackers to gain direct access to a well-protected server. Compromising a server directly may trigger immediate alerts and security measures, making it riskier for the attackers to remain undetected. IoT devices often have weaker security protection mechanisms and more vulnerabilities for exploitation. Attackers may prefer compromising these devices to find their way to the servers. Attackers often seek to maintain their anonymity. Using a compromised IoT node as an intermediate step can help obscure their identity and location.
- The SDN controller [29] is well-protected, and traffic between the controller and the SDN-enabled switches is secure.

Table 4
Attacker Model

Attack types	Details
Reconnaissance attacks	Attackers perform scans (i.e., packet sniffing, port scanning) to identify vulnerable targets and gain access to the network, compromising system integrity.
Data exfiltration attacks	Attackers performed unauthorized copying, transfer, or retrieval of data from servers or individual computers. These attacks pose a considerable threat to network with valuable data, regardless of whether the perpetrator is an external threat actor or a trusted insider.

Defense Model. In the context of a security analysis framework, the diversity module in Stage 3 typically aims to incorporate diversity into the defense model. The defense model represents the security measures and countermeasures implemented to protect the network against potential attacks. The primary objective of the defender system is to defensively deceive attackers by setting up a decoy nodes that is distinct from the IoT network, with the intention of enticing the attackers to engage with the decoy nodes. The purpose is to capture and analyse malicious behavior and reveal attackers' intentions. Legitimate users are unaware of the decoy system, and alerts are generated for the defender when an attacker intrudes into the decoy system. In this paper, we consider two types of decoys for IoT networks: emulation-based and full OS-based. Both can be independently created to fit into the existing infrastructure. A combination of various decoys with different interactive capabilities can increase the chances of attackers connecting them. The intelligence centre performs tasks such as creating, deploying, and updating the decoy system, providing automated attack analysis, and integrating the decoy system with other prevention systems. To monitor outgoing traffic and mislead attackers, the intelligence centre connects to specific ports (SPAN and Trunk port for in and out traffic). The design parameter P_d indicates the probability of an attacker interacting with an individual decoy node.

OS change is the process in which backup OS is used instead of default OS on each node in the network. We have used OS diversification as the MTD technique in the proposed framework. We assume that the possibility of failure in launching a new OS is insignificant. OS diversification makes the network more complicated for the attackers as every time a new OS is launched it presents a new set of vulnerabilities for the attacker to research and exploit. Alternatives to OS diversification include adjusting the applications and services running on the nodes, as well as adopting different

programming languages. Nevertheless, in this paper, we restrict our focus to OS diversification as a means of achieving diversity. The combination of cyber deception and MTD alters the attack surface of the IoT network that consists of real and fake nodes.

Additionally, some researchers have presented MTD algorithms that apply continuous modifications to a targeted system, without taking into account the existence of threats or identifying potential attack points and risk indicators within the system. As a result, this approach can cause excessive power consumption and increase service downtime in the network, leading to delays and higher service costs for both users and providers. In our approach, we consider the OS diversity as a proactive approach to minimize the possibility of service downtime.

Security Conditions. In Stage 5 of the proposed framework, security conditions are utilized to evaluate the security posture of the system and determine whether it meets the desired security objectives. These conditions serve as criteria or requirements that the system must satisfy in order to be considered secure. Security conditions used in Stage 5 include:

- Loss of integrity: if the attackers can compromise a specific number (i.e., one third) of legitimate nodes via reconnaissance or other attacks.
- Loss of data confidentiality: if the confidential information is leaked to unauthorized entities by inside or outsider attackers through data exfiltration attacks.

4. Proposed Defense Mechanisms

In this section, we describe the proposed defense mechanisms to deploy decoy nodes in the real network and to apply diversity on both real and decoy nodes to achieve our goals, and also describe the metrics used in our research.

4.1. Decoy Deployment

In Stage 2 of our proposed framework, we identify the initial deployment of the IoT network with real and decoy devices and servers. The network consists of multiple VLANs as shown in Figure 2, initially we deploy decoy nodes for the real IoT devices in each of the VLANs. We increased the number of decoys with different types of real nodes in each VLAN to enhance deception. To understand the attacker's intentions, at least one decoy server was deployed to capture the attacker's interaction with the network. After deployment of decoy nodes in the VLANs, we establish connections between real IoT nodes and decoy nodes; and redirect the traffic from real to decoy nodes and use a script to generate simulated traffic and place fake credentials on real IoT nodes to redirect attackers towards the decoy nodes. The traffic flows from real nodes to decoy nodes or from decoy node to decoy node are managed by an SDN controller that updates the flow tables in the SDN-enabled switches. In our study, flows are not allowed from decoy nodes to real nodes, as the sole purpose of decoy nodes is to divert attackers from the actual system. Once an attacker is trapped into the decoy system, they will be further redirected to other decoys within the system. The behaviour of the attacker is continuously monitored by the intelligence centre. If an attacker identifies a decoy node, they will likely terminate their interaction with the decoy node and search for a new target.

Decoy capabilities: Our decoy network has following features and capabilities:

- **One-Way Connection:** The decoy network has a one-way connection with real IoT nodes. In such a configuration, fake traffic can be directed from real IoT nodes to decoy nodes to lure the attackers into the decoy network. Once attackers are lured into the decoy network, then they can move laterally within the decoy network and cannot access the real network. This design choice prevents the real IoT nodes from being compromised by the potential attackers utilizing decoy nodes.
- **Data Collection and Transmission:** The decoy network can be equipped to collect data from potential attackers. This data could include information about the attack traffic (e.g., the IP address of attack packets), attackers' activities (e.g., stealing log file data, device configuration files), and tactics. The collected data is then transmitted to an intelligence center for further analysis. This capability is essential for understanding and monitoring potential threats to the network.
- **Interaction Capability:** The decoy network is designed to be heterogeneous that resembles the real network. Some decoys are emulated software to mimic IoT devices, while others are full OS-based and highly interactive,

potentially simulating workstations or other devices. This design approach accounts for the diversity of devices on the network, making it more realistic and challenging for attackers to identify and exploit vulnerabilities across the entire network.

Deployment steps:

1. Decoy System Creation and Deployment:

- Deception Platform: deploy a comprehensive deception platform (e.g., Attivo Networks Threat Defend platform) to create and manage decoys. The deception platform may offer centralized control for ease of management. Decoys are virtualized systems that offer different levels of interaction capabilities. Ensure that the intelligence center has access to the trunk port of the core switch in each VLAN. This allows defenders to mislead attackers into believing that decoys are present on those VLANs, providing a more comprehensive deception strategy.
- Decoy Distribution: Deploy decoys across various network segments like Zscaler [77], including VLANs, to maximize their coverage and effectiveness.
- Decoy Updates: Regularly update the decoys to maintain their realism and stay ahead of attackers' tactics.

2. Automated Attack Analysis, Vulnerability Assessment, and Forensic Reporting:

- Implement automated analysis tools to monitor and analyze incoming and outgoing traffic to detect potential threats and attacks.
- Conduct vulnerability assessments to identify weaknesses in the network that attackers might exploit.
- Generate forensic reports to understand the nature of the attacks, their origins, and the impact on your network.

3. Integration with Other Prevention Systems:

- Connect the intelligence center to your existing security infrastructure, including
 - Security Incident and Event Management (SIEM): Integrate with a SIEM system to correlate and analyze data from the decoy system, allowing more comprehensive threat detection and response.
 - Firewalls and Intrusion Detection/Prevention Systems: Configure rules and policies to block the attackers or malicious traffic based on intelligence from the decoy system.

4. Monitoring Network Traffic:

- Connect to the SPAN (Switched Port Analyzer) or TAP (Test Access Point) port at the Internet egress point to monitor outgoing traffic. This is crucial for identifying malicious activities or data exfiltration by malware.

5. Security Policy Updates:

- Continuously update and adapt security policies based on the intelligence gathered from the decoy system and automated analysis. This may involve altering firewall rules, blocking malicious IPs, and implementing new security measures.

6. Response Strategy:

- Develop a strategy for responding to detected threats. Depending on the severity and nature of the attacks, actions may range from segmentation of the network to threat intelligence sharing.

Purpose of decoy: When an attacker initiates interactions with a decoy, it triggers an alert to the defenders. But in the meanwhile, the security system permits further engagement with additional decoy nodes to uncover the true intentions and assess the behaviors of the attackers. As for the decoy target node, it's worth noting that the decoy network closely mirrors the actual network environment. Consequently, the decoy target node is frequently a decoy server strategically positioned within a decoy network subnet, having a high capacity for interaction with potential attackers.

4.2. Deployment of OS Diversity

Changing OS on each node (either real or decoy) by applying OS diversity is proposed to increase the network complexity. When we run different operating systems for the node, the kernel and applications on the operating system will also change. Exploitable vulnerabilities can exist in the operating system, kernel or applications. We use the term OS diversity to describe our approach as we change the operating system for the node, which implicitly implies the change of kernel and applications. The reason is that a persistent attacker may discover the network design. Diversity in network architecture, configurations, and defense can add to the operational cost incurred by the network operator but results in enhancing network security. By deploying a diverse set of technologies, protocols, and configurations

across the network, an attacker's task becomes more challenging as they would need to have a wide range of tools and techniques to exploit potential vulnerabilities. One example of diversity in network security is through the use of decoys, which are designed to lure attackers. By deploying different types of decoys with various vulnerabilities, configurations, and behaviours, an attacker could be misled thus wasting their time and resources trying to exploit decoy vulnerabilities. This diversionary tactic can buy time for network admins to detect and respond to the attacks.

Furthermore, diversity in network security can also prevent a single point of failure. If all systems and configurations are homogeneous, a single vulnerability or misconfiguration could potentially expose the entire network to an attack. However, by diversifying the network, such as by using different OS, firewalls, and security solutions, an attacker's attempt to exploit a single vulnerability may be limited to a specific segment of the network, reducing the overall impact of the attack. Changes to the traffic flows in an updated system, can impact the regular transmission of data from IoT nodes to servers for service delivery. This can result in increased energy consumption by IoT nodes to accommodate the additional traffic flows, potentially leading to delays in sending packets to the server. We utilize Packet Delivery Ratio (PDR) as a metric to measure the availability of the service.

Our work entails three strategies (zero, random, IMs) aimed at addressing the implementation of OS Diversity, with the primary goal of increasing the likelihood of attackers targeting decoy nodes. This approach effectively deters or prevents security attacks on real nodes. To reach a target node, attackers typically exploit a node, as an entry point and then compromise other nodes to ultimately reach the target. Attackers may discover multiple attack paths through one or more entry points, which are sequences of nodes that can be compromised to reach the target node. In our approach, we define two sets of attack paths, namely AP_r and AP_d . AP_r represents attack paths with real nodes as targets, while AP_d denotes attack paths with decoy nodes as targets. AP_r exclusively includes real nodes, whereas AP_d encompasses both real and decoy nodes. For instance, if an attacker identifies a real node as the entry point and compromises other real nodes until reaching a real target node, this is categorized as an attack path in AP_r . However, the attacker may be redirected to a decoy node within the decoy system. Once the attacker falls into the trap of the decoy system, the attacker may be further redirected to other decoy nodes within the system. If the attacker successfully reaches a decoy target node, then this is counted as an attack path in AP_d . However, if the attacker identifies the decoy node and terminates its interaction, it is not considered an attack path, as the attacker does not reach the decoy target node. Decoy nodes may be regularly reconfigured if deemed as compromised by the network intelligence center. In such cases, the attacker will not be able to recognize the previously visited decoy node during subsequent attacks. We consider three strategies in this paper:

Zero OS Diversity. This is the initial deployment of the real and decoy network and their connections, and the calculation of the security and performance metrics without any MTD technique applied.

Random OS Diversity. Applying random OS diversity on the nodes involves the implementation of a diverse range of OS across different computing nodes within a network. By randomly deploying different OS, organizations can enhance infrastructure security and resilience. This approach mitigates the risks of widespread attacks and reducing the chance of exploiting only a specific vulnerability in a single OS. Random OS diversity ensures that even if one OS is compromised, others remain intact, thereby reducing the overall impact of potential security breaches, and the calculation of the performance metrics to determine its effectiveness.

IMs-Based OS Diversity. To investigate scalability issues when conducting security analysis with the use of GSM, particularly when using an Evolutionary Strategy (ES) to find the optimal solution; we leverage significant Network Centrality Measures (NCM), such as Closeness, to identify the most critical nodes in the network and apply MTD techniques (OS diversity) to these nodes only, without resorting to ES. NCMs aid in identifying important nodes in the HARM model. The formula for computing Closeness centrality, as shown in Equation 1, involves determining the shortest distance between nodes n_i and n_j (in our work we assume number of hops rather than the number of edges), where g represents the total number of nodes in a closed graph with no disjoint components. Overall, NCMs should identify the most critical nodes in the network and apply MTD techniques to essential nodes as an alternative to ES to improve scalability and to examine how the diversity impacts the network's security and performance in the GSM analysis. In this work, we identify the five most important nodes and apply OS diversity on the five nodes.

$$Closeness_{ni} = (g - 1) * \left[\sum_{j=1} d(n_i, n_j) \right]^{-1} \quad (1)$$

4.3. Metrics

Our objective is to assess the effectiveness of proactive defense mechanisms by evaluating their impact on security, performance, and service availability. Specifically, we analyze the extent to which diversion from the real system to decoy targets can deter and mislead attackers, which can be quantified by measuring the number of attack paths towards decoy targets that result in decoys being compromised. We will also investigate how MTD techniques affect the lifespan of the system, based on security failure conditions, and the delivery of services and potential packet dropping. Additionally, we evaluate the efforts of an attacker to attack a network and calculate system risk and the cost associated with MTD operations.

Number of attack paths toward decoy targets. This metric quantifies the extent to which deception tactics are used to redirect or mislead an attacker away from the real system with the use of $|AP_d|$ to sum up attack paths toward the decoy targets.

Mean Time To Compromise (MTTC). This measure quantifies the duration it takes for an attacker to exploit a series of vulnerabilities or weaknesses in different parts of a network, with the ultimate goal of compromising the entire system. MTTC is calculated using Equation 2.

$$MTTC = \sum_{i \in S} S_i \int_{t=0}^{\infty} P_i(t) dt \quad (2)$$

where S refers to a set of network states and S_i is “1”; when in state i the network does not reach the given level. $P_i(t)$ is the probability of the system being in state i at time t .

Packet Delivery Ratio. PDR is based on the concept of attack paths AP_r , which are routes that attackers may take to compromise nodes within the network. The diversity of OS on these nodes can affect the success of such attacks. When a node along an attack path is compromised, the attacker may drop or change packets that pass through the node. However, the attacker may choose not to drop or change packets to avoid detection by an Intrusion Detection System (IDS). The metric calculates the PDR for attack paths, which is the ratio of attack paths that can successfully deliver packets to all of the attack paths $|AP_r|$. In other words, it measures the proportion of attack paths that are able to deliver packets without being affected by packet loss caused by the attacks. The PDR is calculated at each operation, and the mean PDR is calculated over all of the operations until the system reaches one of the predefined security conditions. The focus of this metric is to analyze the impact of attacks on service availability, specifically in terms of packet loss caused by attacks. It assumes that packet losses due to collisions or errors will be handled by data link layer and network layer protocols, and therefore will not affect the service availability represented by the PDR metric. This metric helps assess the resilience of the system against attacks that may result in packet loss and potentially disrupt service availability. By monitoring and analyzing the PDR for attack paths, organizations can gain insights into the effectiveness of their security measures in mitigating such attacks and maintaining service availability in the face of potential disruptions.

System Risk. The System Risk determines the overall risk of a network by assessing the vulnerabilities present in each VM and is measured at both HARM layers. The construction of the HARM for the network was based on the vulnerabilities listed in Table 5 & 6. The probability of a successful attack on a specific node is represented as P_n , while the impact of the attack, if successful, is represented as I_n . The risk value of a node is calculated using Equation 3, the risk value for an attack path can be computed using Equation 4. Finally, the overall risk value of the system can be calculated using Equation 5. In this work, we assume that system risk is the maximum risk value of all attack paths.

$$Risk_n = P_n * I_n \quad (3)$$

$$Risk_{ap} = \sum_{n_i \in ap} R_{ni} \quad (4)$$

$$SystemRisk = \sum_{n_i \in ap} R_{ap} \quad (5)$$

Attack Cost. The cost associated with exploiting vulnerabilities on a node by an attacker is referred to as the AC and can be used to calculate the overall AC of a system using Equation 8. The upper HARM layer can be used to determine the overall network AC. Table 5 and Table 6 provide the costs associated with exploiting a node through vulnerabilities AC_n . The overall AC value of a networked system can be calculated using Equation 8.

$$AC_{ni} = 10 - (Basescore_{ni}) \quad (6)$$

$$AC_{ap} = \sum_{n_i \in ap} AC_{ni} \quad (7)$$

$$SystemAC = \sum_{n_i \in ap} AC_{ap} \quad (8)$$

Return On Attack. ROA shows the readiness of the attacker to use the same nodes, attack paths, and vulnerabilities to penetrate the network. The higher the ROA value, the higher the probability the attacker will exploit the vulnerabilities. The ROA is defined as the ratio of the network risk and the AC, as shown in Equation 9. The value of the total ROA on a single node is the sum of all the ROAs deployed on the same host or passing through the same attack path.

$$ROA_i = (E_{ni} * I_{ni}) / AC_{ni} \quad (9)$$

Extra Operational Cost. The EOC refers to the additional cost that attackers would need to bear in order to successfully exploit the same nodes and vulnerabilities along the attack path to breach a network. It quantifies the increased effort, resources, or cost required by the attackers to achieve their objectives. The EOC metric helps to assess the effectiveness of the network's defensive measures by evaluating the impact they have on increasing the cost and difficulty for the attackers. A higher EOC indicates that the network's security measures are effective in raising the barriers for attackers and making their exploitation attempts more costly. Equation 10 shows the extra cost requires by the attackers to exploit the same nodes, and vulnerabilities on the attack path to penetrate the network.

$$EOC_{ap} = AC_{ap} - AC_{ni} \quad (10)$$

$$SystemEOC = \sum_{n_i \in ap} EOC_{ap} \quad (11)$$

Benefit Cost Ratio (BCR). The BCR in an IoT network with the use of system risk is a financial metric that takes into account the potential benefits and costs of implementing IoT solutions while considering the associated risks. In

the evaluation of BCR, the benefits and costs should be analyzed in the context of the associated system risks. This involves quantifying the potential impact of system risks on the benefits and costs and adjusting them accordingly. For instance, the costs may include investments in robust security measures and risk mitigation strategies to minimize the impact of cybersecurity threats. In our case, we calculate the BCR value using Equation 12, which is the average system risk divided by the average AC.

$$BCR = Avg.SR / Avg.AC \quad (12)$$

5. Simulation and Analysis

In this section, we discuss our simulation design and implementation details along with the simulation outcomes.

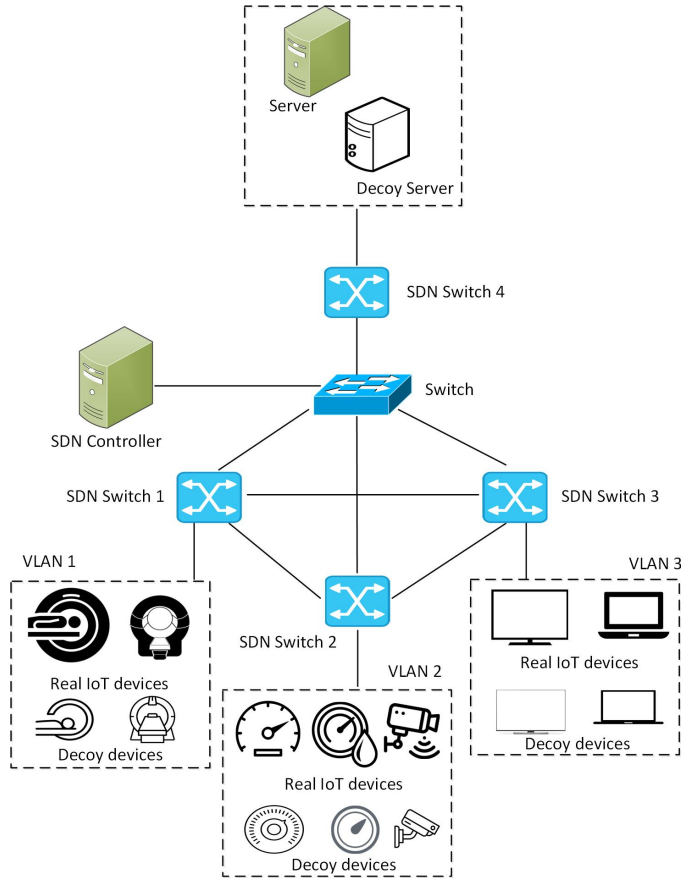


Figure 2: IoT Network Model

5.1. Simulation Setup

The IoT network shown in Figure 2 simulates a smart hospital scenario but as discussed before, we assume that all of the medical devices are compatible with OS diversity [51], consisting of four VLANs with specific devices in each VLAN as shown in Table 5. VLAN1 represents the medical examination room and includes an MRI and a CT Scan machine as IoT devices. VLAN2 represents medical care unit and contains a smart thermostat, a smart meter, and a smart camera, which are equipped with microprocessors, i.e., ARM and have varying memory and storage capacities. For example, a smart meter has limited memory and storage, while a thermostat or a camera can have higher capacities. Smart medical devices like MRI and CT Scan machines have increased capabilities with enhanced CPUs and larger storage capacities. VLAN3 represents staff offices with a smart TV and a laptop, while VLAN4 represents the server

room that contains a server. The VLAN2 is connected to VLAN3 as all of the devices in VLAN2 transmit data (control sensors and video images) to the smart TV and laptop. Initially, VLAN4 is connected to the other three VLANs to enable IoT devices to send information to the server for further processing. Similarly, VLAN2 is connected to VLAN3 to facilitate communication between laptop applications, smart sensors, and the smart camera. The IoT network is configured using SDN-enabled switches for efficient network management and dynamic reconfiguration of connections between different VLANs and devices. This smart hospital scenario highlights the utilization of IoT devices in different VLANs for diverse healthcare applications, such as medical examinations, patient monitoring, and office operations.

HARM [31] is utilized in this research to assess the security posture of an IoT network with multiple vulnerabilities. HARM employs a graphical representation, using AND gates to capture the exploitation of multiple vulnerabilities and OR gates to capture the exploitation of any of the vulnerabilities, in order to represent the network node vulnerability information. The vulnerabilities used in this research are sourced from publicly disclosed vulnerabilities in real-world devices, as documented in the Common Vulnerabilities and Exposures (CVE) and National Vulnerability Database (NVD) [53]. It is assumed that the IoT network nodes have at least one vulnerability that could potentially be exploited by an attacker to gain root privilege. However, the HARM model allows for the inclusion of additional vulnerabilities for network nodes, and the gates in HARM can be combined to capture the complex relationship among vulnerabilities.

Table 5
Real Network with Vulnerability information

Real Node	VLAN	CVE ID	Affected OS/Vendor/Component	BS	I	E	CR	AC
MRI	VLAN1	CVE-2018-8308	Win10	6.6	5.9	0.7	0.006	3.4
CT Scan	VLAN1	CVE-2018-8308	Win10	6.6	5.9	0.7	0.006	3.4
Smart Thermostat	VLAN2	CVE-2018-11315	CT80	6.5	3.6	2.8	0.006	3.5
Smart Meter	VLAN2	CVE-2017-9944	Siemens 7KT	9.8	5.9	3.9	0.042	0.2
Smart Camera	VLAN2	CVE-2018-10660	Axis IP	9.8	5.9	3.9	0.042	0.2
Smart TV	VLAN3	CVE-2018-4094	Mac TvOS	7.8	5.9	1.8	0.012	2.2
Laptop	VLAN3	CVE-2018-8345	Win10	7.5	5.9	1.6	0.012	2.5
Server	VLAN4	CVE-2018-8273	SQL Server	9.8	5.9	3.9	0.042	0.2

The vulnerability information for real network devices and decoy network devices are presented in Table 5 and Table 6 respectively with detailed information about CVE ID and their severity level, i.e., Base Score. In addition, we have considered the likelihood of each vulnerability being exploited by an attacker to gain root privileges, which is expressed as the compromise rate per unit of time, such as per hour. The mean vulnerability exploitation time is estimated using the base score metric from the CVSS, and its inverse is used to calculate the compromise rate. The compromise rate is mentioned in the Table 7 which is dependent on the severity level of the vulnerabilities. Using this compromise rate, we calculate the MTTC using the HARM model. In our simulation, we assume that a compromised node may attempt to disrupt service availability by dropping or corrupting the packets, with corresponding probabilities of P_d and P_{ma} , respectively. However, in practice, detecting such attacks may be difficult for network IDS, as a compromised node may not always drop or manipulate packets that pass through it. We deploy one decoy node against each real device in the corresponding VLAN with three sets of backup OS in which each OS has multiple vulnerabilities. The attacker could exploit any of the vulnerabilities to get root privileges for the node.

Table 8 presents the model notations, their definitions, and default values that we utilized in our experiments. We employed identical weights for w_1 and w_2 , although their values can be altered depending on the significance of a particular component. Our baseline scenario involves a moderate level attack from a less persistent attacker. To exploit vulnerabilities, we used the HARM model, which calculates possible attack routes. In each simulation, the attacker randomly selects an entry point from an attack path and penetrates nodes along the route according to the attacker model's behaviors until one of the security criteria is met. To determine the attacker's behaviour when compromising a node, we verified the privilege level of the vulnerability and also checked if the attacker had a higher privilege level than required, the mean vulnerability exploitation time was added to MTTC. Once the intelligence centre detected the attacker's interaction with the decoy target, decoy nodes were eliminated from the system, and subsequent actions by the attacker would not recognize the same decoy node. In each simulation, decoy nodes were cleared by marking only the compromised real nodes as compromised in the attack paths. The nodes may be randomly selected using the RD strategy during an attack on a node. We assumed that lost connections would force the attacker to exit the network

Table 6
Decoy Network with Vulnerability information

Decoy Node	CVE ID	Affected OS/Vendor	BS	I	E	CR	AC
MRI/CT Scan	CVE-2019-11671	Philips	4.1	3.6	0.5	0.004	5.9
	CVE-2018-14789		6.7	5.9	0.8	0.006	3.3
	CVE-2018-4834	Siemens	9.8	5.9	3.9	0.042	0.2
	CVE-2016-5566		5.3	1.4	3.9	0.004	4.7
	CVE-2020-14886	GE Healthcare	6.0	4.0	1.5	0.006	4.0
	CVE-2019-10964		8.8	5.9	2.8	0.012	1.2
Smart Thermostat/Smart Meter	CVE-2019-20496	Siemens	5.5	3.6	1.8	0.006	4.5
	CVE-2021-21541		6.1	2.7	2.8	0.006	3.9
	CVE-2018-12889	Honeywell Inc	9.8	5.9	3.9	0.042	0.2
	CVE-2017-7659		7.5	3.6	3.9	0.012	2.5
	CVE-2017-9605	Johnson Control	5.5	3.6	1.8	0.012	4.5
	CVE-2019-15735		5.5	3.6	1.8	0.012	4.5
Smart Camera/Smart TV	CVE-2020-27403	TCL	6.5	3.6	2.8	0.006	3.5
	CVE-2020-28055		7.8	5.9	1.8	0.012	2.2
	CVE-2018-4094	Apple Inc	7.8	5.9	1.8	0.012	2.2
	CVE-2018-4095		7.8	5.9	1.8	0.012	2.2
	CVE-2022-44636	Samsung	4.6	2.5	2.1	0.004	5.4
	CVE-2015-5729		9.8	5.9	3.9	0.042	0.2
Laptop/Server	CVE-2017-8530	Win 10	5.4	2.5	2.8	0.004	4.6
	CVE-2017-8490		5.0	3.6	1.3	0.004	5.0
	CVE-2018-14633	Linux	7.0	4.7	2.2	0.012	3.0
	CVE-2017-15126		8.1	5.9	2.2	0.012	1.9
	CVE-2021-20254	Redhat	6.8	5.2	1.6	0.006	3.2
	CVE-2021-29921		9.8	5.9	3.9	0.042	0.2

Table 7
Vulnerability compromise rate according to severity level

Severity Level	Base Score	Compromise Rate (hour)	MTTC
Low	0.1-5.4	Once per 10 days	$1/240 = 0.004$
Medium	5.5-6.9	Once per week	$1/168 = 0.006$
High	7.0-8.9	Twice per week	$1/84 = 0.012$
Critical	9.0-10	Once per day	$1/24 = 0.042$

Table 8
Parameters and their description

Parameter	Description	Value
w1	A weight to consider the security vulnerability associated with failure condition 1	0.5
w2	A weight to consider the security vulnerability associated with failure condition 2	0.5
P_d^{em}	Interaction probability of an attacker with an emulated decoy	0.9
P_d^{os}	Interaction probability of an attacker with an OS-based decoy	1.0
P_a^d	Probability of a packet to be dropped	0.5
P_a^m	Probability of a packet to be manipulated	0.5

and attempt other intrusion methods. After diversification, the attacker is still able to recognize nodes that they have previously compromised and resume their attack on those nodes in subsequent attacks. To simulate different network diversification scenarios, a new HARM model is used to compute potential attack paths for each newly diversified network. The HARM model allows for the identification of potential attack paths that an attacker may take and helps to identify vulnerabilities that need to be addressed to increase network security.

During the simulations, various metric values were collected to assess the network's security performance. The metrics included the AC, the SR, ROA, EOC, and the PDR. To ensure statistical significance, the simulation was run 100 times with different random seeds. After completing the 100 simulations, the metric means were calculated for performance analysis. The simulation results were collected using Python and a computer equipped with an 11th Generation Intel(R) Core(TM) i5-1145G7 @ 2.60GHz 2.61 GHz and 8GB RAM.

5.2. Result Analysis

In this section, we present a comparative analysis of the simulation results by applying the proposed model and metrics to evaluate different use cases. For each metric value, we deploy one decoy node in each VLAN and calculate the metric values without applying the Diversity-based MTD technique, which is called our baseline scenario i.e., Zero-Diversity strategy. In a second scenario, i.e., Random Diversity Strategy, we consider scenarios based on node selection by percentage (i.e., 30%, 50%, 70%, 100%). For each use case, we select nodes randomly according to percentage from the real and decoy nodes and apply diversity and calculate the metric values. In a third Scenario, which is our optimal strategy, we used IMs [36], i.e., Closeness, to find the optimal solution.

The selection of nodes is provided in Figure 3. The figure shows the number of nodes selected in each scenario; a percentage for random approaches and the highest critical value, i.e. five, for the IMs.

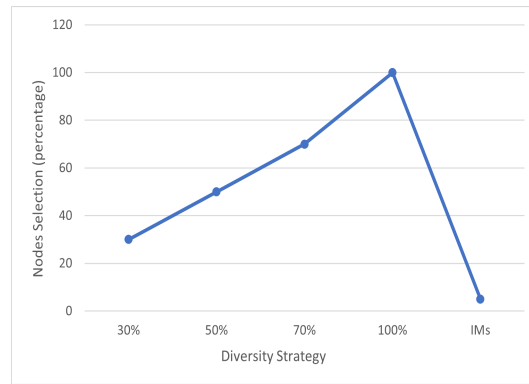


Figure 3: Selection of Nodes

The ROA metric results are provided in Figure 4a. ROA is used to measure the effectiveness of a cybersecurity defense strategy and the results show that a 50% random selection strategy result in a higher ROA value, i.e., higher probability for attacker to exploit vulnerabilities than an IMs-based strategy since the latter relies on a fixed number of nodes, which are critical and cannot be easily targeted by an attacker. However, it's interesting to note that the 100% diversity strategy shows the lowest ROA value. This may indicate that although OS-diversity can be effective in preventing attacks, it may not always be the best approach to achieve high ROA for specific networks.

The analysis of the average AC, provided in 4b, shows that OS-diversity can increase the cost of exploiting vulnerabilities on a node by an attacker. This is because the use of different OS with a different set of vulnerabilities on different nodes in the network creates more complexity, making it more difficult for attackers to successfully exploit vulnerabilities. As shown in Figure 4b, the use of OS-diversity on all nodes, both decoy and real devices, gives the highest AC, indicating the highest level of security. However, this approach may come with a higher implementation cost due to the need for a diverse set of OS across all nodes. In contrast, using OS-diversity on critical nodes only, as in the IMs scenario, gives slightly lower AC values, but with a lower implementation cost compared to random selection. This approach focuses on using OS-diversity on the most critical nodes in the network, which can significantly improve the security of the network at a lower cost.

Random OS-diversity can lead to higher system risk values compared to the IMs-based approach, as shown in Figure 5a. This is because random OS-diversity may not necessarily focus on critical nodes in the network, leading to a higher risk of system failure or security breaches. However, it's important to note that the selection of the OS-diversity approach depends on various factors, including the cost of implementation, the criticality of nodes, and the overall security goals of the network. While the IMs-based approach may provide better security, it may come with a higher

Proactive Defense Mechanism: Enhancing IoT Security

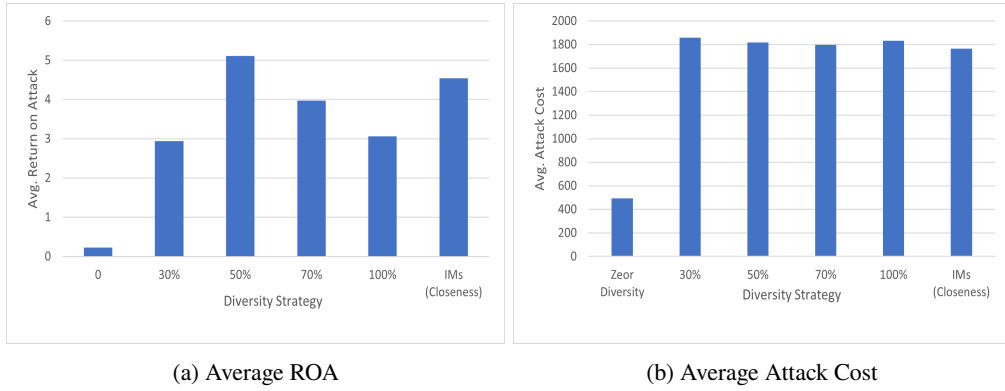


Figure 4: Attack Cost Analysis

implementation cost, and it may not be feasible for networks with a large number of nodes. The benefit-cost ratio, as shown in Figure 5b, is related to benefit cost ratio. In a network with OS-diversity, BCR should be calculated to check the feasibility of the network in the context of associated system risk; which is better in IMs-based scenario and slightly lower in random based scenario.

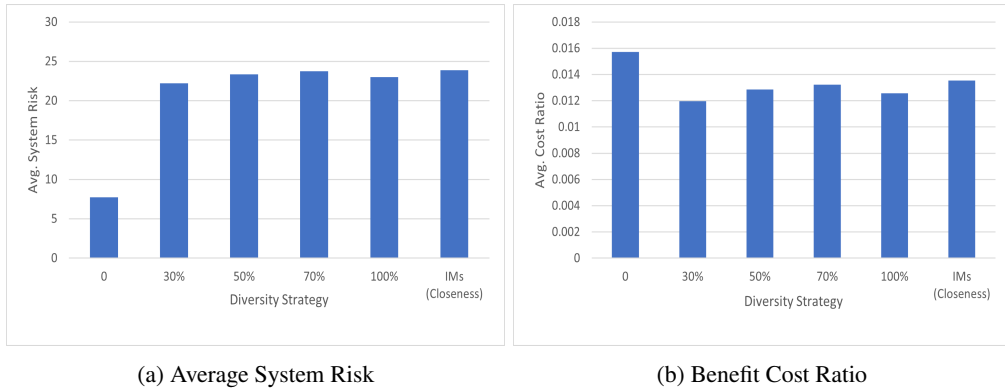


Figure 5: System Risk Analysis

Figure 6b shows the average PDR (service availability). In a network with OS-diversity, the PDR tends to be higher compared to a network where shuffling [30] is used. This is because OS-diversity produces a smaller number of attack paths towards decoy targets than shuffling as mentioned in Table 9. As a result, the attacker has a smaller chance to drop or manipulate packets passing through the attack paths, which leads to a higher PDR and better service availability. As in an OS-diversified network, different types of operating systems are used for various nodes within the network. This diversity results in a variety of configurations and software implementations across the network. Our studies suggest that this diversity makes it difficult for an attacker to exploit a vulnerability consistently across the entire network. As a result, it becomes challenging for an attacker to successfully launch an attack. This increased difficulty in launching successful attacks means that the Packet Delivery Ratio (PDR) tends to be higher in OS-diverse networks. A higher PDR indicates better service availability, as network packets are less likely to be dropped or manipulated by attackers.

In contrast, when the network employs a shuffling approach [30], nodes randomly change their positions within the network. The shuffling can create new paths or configurations within the network. This implies that these changes may also introduce new attack paths that an attacker can potentially exploit. The uncertainty of node positions can lead to a lower PDR compared to an OS-diverse network, as it might be easier for attackers to identify and target vulnerabilities in the shuffled network.

Table 9
PDR Comparison

Technique	PDR Ratio
[30]	0.8
OS Diversity	0.9

Figure 6a shows the EOC metric result, a measure of the additional cost or effort required for an attacker to successfully exploit vulnerabilities in a system that is protected by an OS-diversity defense strategy. According to Figure 6a, it appears that the IMs-based OS-diversity strategy outperformed the other random-based scenarios in terms of the EOC metric. This suggests that this strategy is more effective at raising the cost and effort required for attackers to successfully exploit vulnerabilities. Deploying OS-diversity can make it more difficult and costly for attackers to mount successful attacks against a network. This is because an attacker who is familiar with one type of operating system will need to learn how to exploit the vulnerabilities of a different type of OS, which requires additional time and effort on their part. However, it's important to note that while OS-diversity may increase the AC for an attacker, it does not necessarily change the overall system Risk.

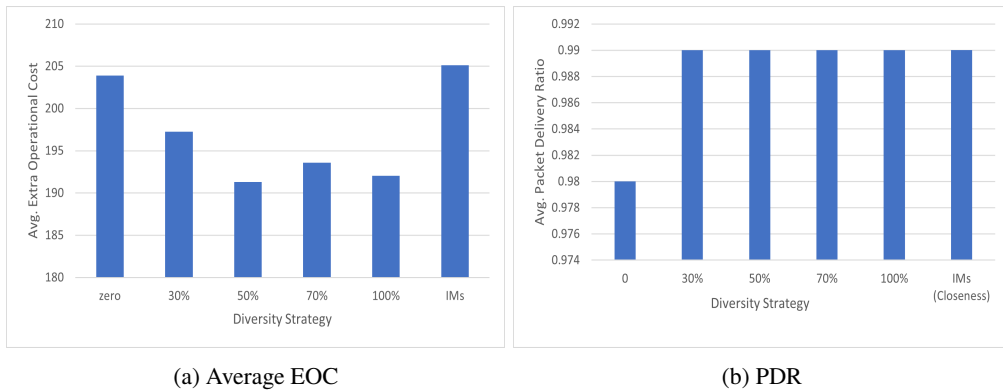


Figure 6: EOC and Service Availability Analysis

6. Conclusion

We have proposed a proactive defense framework that enables the optimal deployment of proactive defense in an SDN-based IoT environment. Our approach has leveraged the SDN architecture to facilitate flexible deployment and easy integration of defense mechanisms. By employing MTD techniques and cyber deception, we have established defense mechanisms that mislead the attackers, causing them to make incorrect decisions and deplete their resources with minimal impact on resources. We have introduced a diversity-based defense model to increase the attackers' efforts and costs. Additionally, we have presented strategy for selecting critical nodes, striking a balance between effectiveness and cost to optimise defense implementation. Evaluation results have demonstrated that our proposed method effectively mitigates attacks with lower implementation costs, while maintaining service availability in IoT networks. In our future work, we will focus on incorporating more security mechanisms in a flexible manner for real-world applications, performing scalability analysis for the proposed approach, and exploring proactive adaptation techniques to differentiate between malicious attackers from legitimate users. Through extensive experimentation, we aim to validate the effectiveness of our approach in mitigating attacks while ensuring high-performance levels in IoT networks.

References

- [1] AJMAL, A. B., ALAM, M., KHALIQ, A. A., KHAN, S., QADIR, Z., AND MAHMUD, M. P. Last line of defense: Reliability through inducing cyber threat hunting with deception in scada networks. *IEEE Access* 9 (2021), 126789–126800.
- [2] AL-SHAER, E. Toward network configuration randomization for moving target defense. In *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*. Springer, 2011, pp. 153–159.

- [3] ALAVIZADEH, H., JANG-JACCARD, J., AND KIM, D. S. Evaluation for combination of shuffle and diversity on moving target defense strategy for cloud computing. In *2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)* (2018), IEEE, pp. 573–578.
- [4] ALAVIZADEH, H., KIM, D. S., HONG, J. B., AND JANG-JACCARD, J. Effective security analysis for combinations of mtd techniques on cloud computing (short paper). In *International Conference on Information Security Practice and Experience* (2017), Springer, pp. 539–548.
- [5] ALAVIZADEH, H., KIM, D. S., AND JANG-JACCARD, J. Model-based evaluation of combinations of shuffle and diversity mtd techniques on the cloud. *Future Generation Computer Systems* 111 (2020), 507–522.
- [6] ALMOHAIMEED, A., GAMPA, S., AND SINGH, G. Privacy-preserving iot devices. In *2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* (2019), IEEE, pp. 1–5.
- [7] ANIRUDH, M., THILEEBAN, S. A., AND NALLATHAMBI, D. J. Use of honeypots for mitigating dos attacks targeted on iot networks. In *2017 International conference on computer, communication and signal processing (ICCCSP)* (2017), IEEE, pp. 1–4.
- [8] ANTONATOS, S., AKRITIDIS, P., MARKATOS, E. P., AND ANAGNOSTAKIS, K. G. Defending against hitlist worms using network address space randomization. In *Proceedings of the 2005 ACM workshop on Rapid malware* (2005), pp. 30–40.
- [9] ANWAR, A. H., KAMHOUA, C., AND LESLIE, N. Honeypot allocation over attack graphs in cyber deception games. In *2020 International Conference on Computing, Networking and Communications (ICNC)* (2020), IEEE, pp. 502–506.
- [10] ANWAR, A. H., AND KAMHOUA, C. A. Cyber deception using honeypot allocation and diversity: A game theoretic approach. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)* (2022), IEEE, pp. 543–549.
- [11] AZAB, M., HASSAN, R., AND ELTOWEISSY, M. Chameleonsoft: A moving target defense system. In *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)* (2011), IEEE, pp. 241–250.
- [12] BAWANY, N. Z., SHAMSI, J. A., AND SALAH, K. Ddos attack detection and mitigation using sdn: methods, practices, and solutions. *Arabian Journal for Science and Engineering* 42 (2017), 425–441.
- [13] BERNARDOS, C. J., DE LA OLIVA, A., SERRANO, P., BANCHS, A., CONTRERAS, L. M., JIN, H., AND ZÚÑIGA, J. C. An architecture for software defined wireless networking. *IEEE wireless communications* 21, 3 (2014), 52–61.
- [14] BHUNIA, S. S., AND GURUSAMY, M. Dynamic attack detection and mitigation in iot using sdn. In *2017 27th International telecommunication networks and applications conference (ITNAC)* (2017), IEEE, pp. 1–6.
- [15] BORBOR, D., WANG, L., JAJODIA, S., AND SINGHAL, A. Optimizing the network diversity to improve the resilience of networks against unknown attacks. *Computer Communications* 145 (2019), 96–112.
- [16] BULLE, B. B., SANTIN, A. O., VIEGAS, E. K., AND DOS SANTOS, R. R. A host-based intrusion detection model based on os diversity for scada. In *IECON 2020 The 46th annual conference of the IEEE industrial electronics society* (2020), IEEE, pp. 691–696.
- [17] BUTUN, I., ÖSTERBERG, P., AND SONG, H. Security of the internet of things: Vulnerabilities, attacks, and countermeasures. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 616–644.
- [18] CAI, G.-L., WANG, B.-S., HU, W., AND WANG, T.-Z. Moving target defense: state of the art and characteristics. *Frontiers of Information Technology & Electronic Engineering* 17, 11 (2016), 1122–1153.
- [19] CHAABOUNI, N., MOSBAH, M., ZEMMARI, A., SAUVIGNAC, C., AND FARUKI, P. Network intrusion detection for iot security based on learning techniques. *IEEE Communications Surveys & Tutorials* 21, 3 (2019), 2671–2701.
- [20] CHEN, H., CAM, H., AND XU, S. Quantifying cybersecurity effectiveness of dynamic network diversity. *IEEE Transactions on Dependable and Secure Computing* 19, 6 (2021), 3804–3821.
- [21] CHO, J.-H., SHARMA, D. P., ALAVIZADEH, H., YOON, S., BEN-ASHER, N., MOORE, T. J., KIM, D. S., LIM, H., AND NELSON, F. F. Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Communications Surveys & Tutorials* 22, 1 (2020), 709–745.
- [22] DE OLIVEIRA, B. T., GABRIEL, L. B., AND MARGI, C. B. Tinsydn: Enabling multiple controllers for software-defined wireless sensor networks. *IEEE Latin America Transactions* 13, 11 (2015), 3690–3696.
- [23] DOWLING, S., SCHUKAT, M., AND MELVIN, H. A zigbee honeypot to assess iot cyberattack behaviour. In *2017 28th Irish signals and systems conference (ISSC)* (2017), IEEE, pp. 1–6.
- [24] DU, M., AND WANG, K. An sdn-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial internet of things. *IEEE Transactions on Industrial Informatics* 16, 1 (2019), 648–657.
- [25] DUMAN, O., ZHANG, M., WANG, L., AND DEBBABI, M. Measuring the security posture of iec 61850 substations with redundancy against zero day attacks. In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2017), IEEE, pp. 108–114.
- [26] FDA. Fda, us food and drug administration, 2017.
- [27] GALLUCCIO, L., MILARDO, S., MORABITO, G., AND PALAZZO, S. Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks. In *2015 IEEE conference on computer communications (INFOCOM)* (2015), IEEE, pp. 513–521.
- [28] GARCIA, M., BESSANI, A., GASHI, I., NEVES, N., AND OBELHEIRO, R. Os diversity for intrusion tolerance: Myth or reality? In *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)* (2011), IEEE, pp. 383–394.
- [29] GÄRTNER, F. C. Byzantine failures and security: Arbitrary is not (always) random. *INFORMATIK 2003-Mit Sicherheit Informatik, Schwerpunkt "Sicherheit-Schutz und Zuverlässigkeit"* (2003).
- [30] GE, M., CHO, J.-H., KIM, D., DIXIT, G., AND CHEN, I.-R. Proactive defense for internet-of-things: Moving target defense with cyberdeception. *ACM Transactions on Internet Technology (TOIT)* 22, 1 (2021), 1–31.
- [31] GE, M., HONG, J. B., GUTTMANN, W., AND KIM, D. S. A framework for automating security analysis of the internet of things. *Journal of Network and Computer Applications* 83 (2017), 12–27.
- [32] GORBENKO, A., ROMANOVSKY, A., TARASYUK, O., AND BILOBORODOV, O. From analyzing operating system vulnerabilities to designing multiversion intrusion-tolerant architectures. *IEEE Transactions on Reliability* 69, 1 (2019), 22–39.
- [33] GUO, X., LIN, H., LI, Z., AND PENG, M. Deep-reinforcement-learning-based qos-aware secure routing for sdn-iot. *IEEE Internet of things journal* 7, 7 (2019), 6242–6251.
- [34] HAMADA, A. O., AZAB, M., AND MOKHTAR, A. Honeypot-like moving-target defense for secure iot operation. In *2018 IEEE 9th Annual*

- Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (2018), IEEE, pp. 971–977.
- [35] HASSIJA, V., CHAMOLA, V., SAXENA, V., JAIN, D., GOYAL, P., AND SIKDAR, B. A survey on iot security: application areas, security threats, and solution architectures. *IEEE Access* 7 (2019), 82721–82743.
- [36] HONG, J. B., AND KIM, D. S. Scalable security model generation and analysis using k-importance measures. In *Security and Privacy in Communication Networks: 9th International ICST Conference, SecureComm 2013, Sydney, NSW, Australia, September 25-28, 2013, Revised Selected Papers 9* (2013), Springer, pp. 270–287.
- [37] HONG, J. B., AND KIM, D. S. Assessing the effectiveness of moving target defenses using security models. *IEEE Transactions on Dependable and Secure Computing* 13, 2 (2015), 163–177.
- [38] HONG, J. B., KIM, D. S., CHUNG, C.-J., AND HUANG, D. A survey on the usability and practical applications of graphical security models. *Computer Science Review* 26 (2017), 1–16.
- [39] HUANG, Y., AND GHOSH, A. K. Introducing diversity and uncertainty to create moving attack surfaces for web services. In *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*. Springer, 2011, pp. 131–151.
- [40] ISLAM, M. M., AND AL-SHAER, E. Active deception framework: An extensible development environment for adaptive cyber deception. In *2020 IEEE Secure Development (SecDev)* (2020), IEEE, pp. 41–48.
- [41] JAFARIAN, J. H., AL-SHAER, E., AND DUAN, Q. Openflow random host mutation: transparent moving target defense using software defined networking. In *Proceedings of the first workshop on Hot topics in software defined networks* (2012), pp. 127–132.
- [42] KANELLOPOULOS, A., AND VAMVOUDAKIS, K. G. A moving target defense control framework for cyber-physical systems. *IEEE Transactions on Automatic Control* 65, 3 (2019), 1029–1043.
- [43] KHOSRAVI-FARMAD, M., RAMAKI, A. A., AND BAFGHI, A. G. Moving target defense against advanced persistent threats for cybersecurity enhancement. In *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)* (2018), IEEE, pp. 280–285.
- [44] KOO, H., CHEN, Y., LU, L., KEMERLIS, V. P., AND POLYCHRONAKIS, M. Compiler-assisted code randomization. In *2018 IEEE symposium on security and privacy (SP)* (2018), IEEE, pp. 461–477.
- [45] KOTRONIS, V., DIMITROPOULOS, X., AND AGER, B. Outsourcing the routing control logic: Better internet routing based on sdn principles. In *Proceedings of the 11th ACM workshop on hot topics in networks* (2012), pp. 55–60.
- [46] KOUACHI, A. I., SAHRAOUI, S., AND BACHIR, A. Per packet flow anonymization in 6lowpan iot networks. In *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)* (2018), IEEE, pp. 1–7.
- [47] LI, C., QIN, Z., NOVAK, E., AND LI, Q. Securing sdn infrastructure of iot-fog networks from mitm attacks. *IEEE Internet of Things Journal* 4, 5 (2017), 1156–1164.
- [48] LIU, W., GE, M., AND KIM, D. S. Integrated proactive defense for software defined internet of things under multi-target attacks. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)* (2020), IEEE, pp. 767–774.
- [49] LIU, Y., GRIGORYAN, G., KAMHOUA, C. A., AND NJILLA, L. L. Leverage sdn for cyber-security deception in internet of things. *Modeling and Design of Secure Internet of Things* (2020), 479–503.
- [50] MAHMOOD, K., AND SHILA, D. M. Moving target defense for internet of things using context aware code partitioning and code diversification. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)* (2016), IEEE, pp. 329–330.
- [51] MUDALIAR, M. D., AND SIVAKUMAR, N. Iot based real time energy monitoring system using raspberry pi. *Internet of Things* 12 (2020), 100292.
- [52] NAVAS, R. E., CUPPENS, F., CUPPENS, N. B., TOUTAIN, L., AND PAPADOPOULOS, G. Z. Mtd, where art thou? a systematic review of moving target defense techniques for iot. *IEEE internet of things journal* 8, 10 (2020), 7818–7832.
- [53] NIST. National vulnerability database. national institute of standards and technology, u.s. government, 2023.
- [54] NIZZI, F., PECORELLA, T., ESPOSITO, F., PIERUCCI, L., AND FANTACCI, R. Iot security via address shuffling: The easy way. *IEEE Internet of Things Journal* 6, 2 (2019), 3764–3774.
- [55] OO, W. K. K., AND KOIDE, H. A framework of moving target defenses for the internet of things. *Bulletin of Networking, Computing, Systems, and Software* 8, 2 (2019), 104–107.
- [56] QIN, Z., DENKER, G., GIANNELLI, C., BELLAVISTA, P., AND VENKATASUBRAMANIAN, N. A software defined networking architecture for the internet-of-things. In *2014 IEEE network operations and management symposium (NOMS)* (2014), IEEE, pp. 1–9.
- [57] RAVI, N., AND SHALINIE, S. M. Learning-driven detection and mitigation of ddos attack in iot via sdn-cloud architecture. *IEEE Internet of Things Journal* 7, 4 (2020), 3559–3570.
- [58] SALMAN, O., ELHAJ, I., CHEHAB, A., AND KAYSSI, A. Iot survey: An sdn and fog computing perspective. *Computer Networks* 143 (2018), 221–246.
- [59] SAVOLA, R. M., ABIE, H., AND SIHVONEN, M. Towards metrics-driven adaptive security management in e-health iot applications. In *BODYNETS* (2012), pp. 276–281.
- [60] SENGUPTA, S., CHOWDHARY, A., SABUR, A., ALSHAMRANI, A., HUANG, D., AND KAMBHAMPATI, S. A survey of moving target defenses for network security. *IEEE Communications Surveys & Tutorials* 22, 3 (2020), 1909–1941.
- [61] SMITH, J., JOHNSON, A., AND DAVIS, M. A comparative analysis of intrusion detection techniques for iot networks. In *IEEE International Conference on Internet of Things (iThings)* (2020), IEEE.
- [62] TAMBE, A., AUNG, Y. L., SRIDHARAN, R., OCHOA, M., TIPPENHAUER, N. O., SHABTAI, A., AND ELOVICI, Y. Detection of threats to iot devices using scalable vpn-forwarded honeypots. In *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy* (2019), pp. 85–96.
- [63] TEAM, P. Pax address space layout randomization. <http://pax.grsecurity.net/docs/aslr.txt> (2003).
- [64] TORQUATO, M., MACIEL, P., AND VIEIRA, M. Security and availability modeling of vm migration as moving target defense. In *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)* (2020), IEEE, pp. 50–59.
- [65] WANG, C., AND LU, Z. Cyber deception: Overview and the road ahead. *IEEE Security & Privacy* 16, 2 (2018), 80–85.
- [66] WANG, J., MIAO, Y., ZHOU, P., HOSSAIN, M. S., AND RAHMAN, S. M. M. A software defined network routing in wireless multihop network.

- Journal of Network and Computer Applications* 85 (2017), 76–83.
- [67] WANG, S., SHI, H., HU, Q., LIN, B., AND CHENG, X. Moving target defense for internet of things based on the zero-determinant theory. *IEEE Internet of Things Journal* 7, 1 (2019), 661–668.
- [68] WINN, M., RICE, M., DUNLAP, S., LOPEZ, J., AND MULLINS, B. Constructing cost-effective and targetable industrial control system honeypots for production networks. *International Journal of Critical Infrastructure Protection* 10 (2015), 47–58.
- [69] YAO, S., LI, Z., GUAN, J., AND LIU, Y. Stochastic cost minimization mechanism based on identifier network for iot security. *IEEE Internet of Things Journal* 7, 5 (2019), 3923–3934.
- [70] ZENG, D., LI, P., GUO, S., MIYAZAKI, T., HU, J., AND XIANG, Y. Energy minimization in multi-task software-defined sensor networks. *IEEE transactions on computers* 64, 11 (2015), 3128–3139.
- [71] ZHANG, L., SHETTY, S., LIU, P., AND JING, J. Rootkitdet: Practical end-to-end defense against kernel rootkits in a cloud environment. In *Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19* (2014), Springer, pp. 475–493.
- [72] ZHANG, M., WANG, L., JAJODIA, S., SINGHAL, A., AND ALBANESE, M. Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks. *IEEE Transactions on Information Forensics and Security* 11, 5 (2016), 1071–1086.
- [73] ZHANG, Y., LI, M., BAI, K., YU, M., AND ZANG, W. Incentive compatible moving target defense against vm-colocation attacks in clouds. In *Information Security and Privacy Research: 27th IFIP TC 11 Information Security and Privacy Conference, SEC 2012, Heraklion, Crete, Greece, June 4-6, 2012. Proceedings 27* (2012), Springer, pp. 388–399.
- [74] ZHANG, Z., NJILLA, L., KAMHOUA, C. A., AND YU, Q. Thwarting security threats from malicious fpga tools with novel fpga-oriented moving target defense. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27, 3 (2018), 665–678.
- [75] ZHENG, J., AND NAMIN, A. S. A survey on the moving target defense strategies: An architectural perspective. *Journal of Computer Science and Technology* 34 (2019), 207–233.
- [76] ZHOU, Y., CHENG, G., AND YU, S. An sdn-enabled proactive defense framework for ddos mitigation in iot networks. *IEEE Transactions on Information Forensics and Security* 16 (2021), 5366–5380.
- [77] ZSCALRE. Zscaler, deploying services, 2023.