Challenges in Moving Target Defense Techniques

Thesis submitted in partial fulfillment of the requirements
for the degree of
Master of Technology in Computer Technology
of
Computer Science and Engineering Department
of
Jadavpur University

by

Surojit Mondal

Registration No. - 160074 of 2021-2022
Examination Roll No. - M6TCT24002

under the supervision of

Dr. Mridul Sankar Barik
Assistant Professor

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University
2024

Certificate from the Supervisor

This is to certify that the work embodied in this thesis entitled **"Challenges in Moving Target Defense Techniques"** has been satisfactorily completed by **Surojit Mondal** (Registration Number 160074 of $2021-2022$; Class Roll No. 002110504003; Examination Roll No. $M6TCT24002$). It is a bona-fide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata for partial fulfilment of the requirements for the awarding of the **Master of Technology in Computer Technology** degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year $2023-2024$.

Dr. Mridul Sankar Barik,
Assistant Professor,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Forwarded By:

Prof. Nandini Mukhopadhyay,
Head,
Department of Computer Science and Engineering,
Jadavpur University.

Prof. Dipak Laha,
DEAN,
Faculty of Engineering & Technology,
Jadavpur University.

Department of Computer Science and Engineering
Faculty of Engineering And Technology
Jadavpur University, Kolkata - 700 032

Certificate of Approval

This is to certify that the thesis entitled **"Challenges in Moving Target Defense Techniques"** is a bona-fide record of work carried out by **Surojit Mondal** (Registration Number 160074 of 2021 − 2022; Class Roll No. 002110504003; Examination Roll No. $M6TCT24002$) in partial fulfilment of the requirements for the award of the degree of **Master of Technology in Computer Technology** in the **Department of Computer Science and Engineering, Jadavpur University**, during the period of July 2023 to May 2024. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose of which it has been submitted.

Examiners:

(Signature of The Examiner)

(Signature of The Supervisor)

Department of Computer Science and Engineering
Faculty of Engineering And Technology
Jadavpur University, Kolkata - 700 032

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that the thesis entitled **"Challenges in Moving Target Defense Techniques"** contains literature survey and original research work by the undersigned candidate, as a part of his degree of **Master of Technology in Computer Technology** in the **Department of Computer Science and Engineering, Jadavpur University**. All information have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Surojit Mondal

Examination Roll No.: M6TCT24002

Registration No.: 160074 of 2021 – 2022

Thesis Title: Challenges in Moving Target Defense Techniques

Signature of the Candidate:

ACKNOWLEDGEMENT

I am pleased to express my gratitude and regards towards my Project Guide **Dr. Mridul Sankar Barik**, Assistant Professor, Department of Computer Science and Engineering, Jadavpur University, without whose valuable guidance, inspiration and attention towards me, pursuing my project would have been impossible.

Last but not the least, I express my regards towards my friends and family for bearing with me and for being a source of constant motivation during the entire term of the work.

Surojit Mondal
MTCT Final Year
Exam Roll No. - M6TCT24002
Regn. No. - 160074 of 2021 – 2022
Department of Computer Science and Engineering,
Jadavpur University.

Contents

Certificate from Supervisor **I**

Certificate of Approval **II**

Declaration of Originality **III**

Acknowledgement **IV**

1 Introduction **1**
1.1 Background . . . 1
1.2 Research Statement . . . 2
1.3 Contribution of the Thesis . . . 2
1.4 Outline of the Thesis . . . 2

2 Related Works **3**
2.1 Moving Target Defense . . . 3
2.1.1 Overview . . . 3
2.1.2 History . . . 3
2.1.3 Definition . . . 4
2.1.4 MTD Classification for Network . . . 4
2.1.5 Classification of MTD . . . 5
2.1.6 Elements of an MTD Strategy . . . 6
2.1.7 Timeliness-Based MTD . . . 8
2.1.8 Operation-Based MTD . . . 8
2.2 Moving Target Defense for Internet of Things . . . 14
2.2.1 Overview . . . 14
2.2.2 History . . . 15
2.2.3 Definition . . . 15
2.2.4 MTD in Various IoT Industries . . . 15
2.2.5 Security Issues in MTD for IoT . . . 15
2.3 Software-Defined Networking(SDN) Based MTD . . . 16
2.3.1 Overview . . . 16
2.3.2 History . . . 17
2.3.3 Definition . . . 17
2.3.4 Key Components . . . 17

2.3.5 Advantages of SDN-Based Network Over Traditional Network . . . 18
2.3.6 SDN Environment on MTD Techniques . . . 19
2.4 SDN-Based MTD for IoT . . . 20
2.4.1 Overview . . . 20
2.4.2 Advantages of SDN-Based MTD for IoT: . . . 20
2.4.3 Challenges of SDN-Based MTD for IoT: . . . 21
2.5 Attack Modeling Techniques . . . 21
2.5.1 Advanced Persistent Threats . . . 22
2.5.2 Reconnaissance . . . 23
2.5.3 Denial of Service (DoS) . . . 24
2.5.4 Zero-Day . . . 24
2.6 Defense Methods . . . 24
2.6.1 Proxy Filtering . . . 24
2.6.2 Intrusion Detection System . . . 24
2.6.3 Web Analytics . . . 25
2.6.4 Intrusion Prevention System . . . 25
2.6.5 Data Execution Prevention . . . 25
2.6.6 Access Control List . . . 25
2.6.7 Tarpit . . . 26
2.6.8 DNS Redirect . . . 26
2.6.9 Honey-Pot . . . 26
2.6.10 Anti-Virus . . . 26

3 Classifying Moving Target Defenses 27
3.1 What to Move? . . . 27
3.1.1 Address and Port Mutation . . . 28
3.1.2 Route Mutation . . . 32
3.1.3 Host Mutation . . . 34
3.1.4 Dynamic Resource Allocation via MTD . . . 38
3.1.5 Architectures Utilizing Multiple MTD Methods . . . 39
3.1.6 Other . . . 41
3.2 When to Move? . . . 42
3.2.1 Constant Period Switching . . . 42
3.2.2 Variable Period Switching (VPS) . . . 43
3.3 How to Move? . . . 44
3.3.1 Randomness . . . 44
3.3.2 Pre-Shared Key Derivation . . . 45
3.3.3 Current State-Based Calculations . . . 45
3.3.4 Game Theory Modeling . . . 46
3.4 SDN-Based Shuffling MTD . . . 47

4 Attack Graphs 51
4.1 Attack Graph . . . 51
4.2 Categories of Attack Graphs . . . 52
4.2.1 Based on Purpose . . . 52
4.2.2 Based on Representation . . . 52

4.2.3 Based on Scope . . . 52
4.2.4 Based on Techniques . . . 52
4.2.5 Based on Attributes . . . 53
4.2.6 Based on Source of Data . . . 53
4.2.7 Based on Time Horizon . . . 53
4.3 Security Vulnerabilities . . . 53
4.4 Network Topology . . . 54
4.5 Three-Tier Attack Graph (TAG) . . . 56
4.6 Advantages of Three-Tier Attack Graph (TAG) in the Moving Target Defense (MTD) 59

5 Challenges of Moving Target Defense 61
5.1 What To Move . . . 61
5.2 When To Move . . . 61
5.3 How To Move . . . 61

6 Experiments 63
6.1 SDN Network Configuration and Routing Experiment using Open vSwitch and Ryu Controller . . . 64

7 Conclusion and Future Work 68
7.1 Conclusion . . . 68
7.2 Future Work . . . 69

List of Figures

2.1 MTD Classification for Network 4
2.2 Types of MTD 6
2.3 Elements of an MTD Strategy 6
2.4 Shuffling Technique 9
2.5 Diversity Technique 11
2.6 Redundancy Technique 12
2.7 Hybrid Technique 14
2.8 Traditional Network V/s SDN-based Network 18
2.9 Venn Diagram of Three Technologies: IoT, MTD, and SDN 20
2.10 Different Phases of An APT 23

3.1 MAC Address Shuffling Sequence to Send Data from Host A to Host B. 48
3.2 IP Address Shuffling Sequence to Send Data From Host A to Host B. 50

4.1 An Example of Network Topology 54
4.2 Three Tier Attack Graph 58

6.1 Network Topology with IP Shuffling Technique 66
6.2 Network Topology Logs 67

List of Tables

3.1 A Taxonomy of Surveyed MTD Architectures, Covering the Address and Port Mutation 32
3.2 A Taxonomy of Surveyed MTD Architectures, Covering Route Mutation 34
3.3 A Taxonomy of Surveyed MTD Architectures, Host Mutation 37
3.4 A taxonomy of surveyed MTD architectures, Dynamic Resource Allocation via MTD 39
3.5 A Taxonomy of Surveyed MTD Architectures, Architectures Utilizing Multiple MTD Methods 41
3.6 A taxonomy of surveyed MTD architectures, Others 42

4.1 Vulnerabilities of an Example Network 55
4.2 Potential Attack Path of All Host 59

Abstract

Moving Target Defense (MTD) is a cybersecurity strategy designed to enhance system resilience against cyber threats by continually altering the system's attack surface. This approach counters the limitations of traditional static security measures, which often fall short against dynamic cyber threats. MTD techniques, such as IP shuffling, packet header randomization, and network topology shuffling, create uncertainty and confusion for attackers, making it harder for them to exploit vulnerabilities.

The static nature of many currently used network systems offers several practical benefits, such as cost optimization and ease of deployment. However, this static nature also makes these systems vulnerable to attackers who can silently observe and gain insights before launching a significant attack on the infrastructure. Moving Target Defense (MTD) is an emerging approach that promises to counter this type of threat by continuously altering system parameters and changing the attack surface of protected systems.

This survey aims to understand the effectiveness of MTD by analyzing existing research in this field. It will identify current shortcomings and suggest potential improvements that need to be addressed in future proposals to make MTD a viable solution for mitigating current cybersecurity threats in real-world scenarios.

As part of our research explores the effectiveness of attack graph-based security analytics in implementing MTD strategies. It introduces the existing model Three-Tier Attack Graph (TAG) model, which separates network topology and vulnerability information into distinct layers: connectivity, remote vulnerabilities, and local vulnerabilities. This structured approach simplifies the representation of attack paths, enhances the manageability of attack graphs, and provides a comprehensive framework for security analysis.

Our research makes several contributions: it demonstrates the benefits of the TAG model for MTD, reviews recent MTD methods and their effectiveness against various threats, and identifies challenges with key MTD components. The findings suggest that combining MTD with traditional security measures significantly improves the robustness of network defenses, particularly in the face of increasing cloud infrastructure usage. The TAG model's ability to predict attack paths and streamline security analysis offers valuable insights for proactive threat mitigation and defense strategy optimization. Finally, the configuration of an SDN environment using Open vSwitch and the Ryu controller focuses on network communication, routing, and security features. The setup involves the rest_router application within Ryu, dynamic IP management via DHCP, and static route configuration, verified through topology logs.

Chapter 1

Introduction

1.1 Background

Moving Target Defense is a form of cybersecurity strategy that aims to increase the resilience of systems from cyber threats through continuous change or variation of the aspects of the system's attack surface. It is a reflection of the presumed or hostility model in which an attacker seeks the attack surface of a network system to identify a potential weakness to exploit.

In recent years, there has been a significant increase in the usage and complexity of both network and cloud infrastructures. Predictions from Gartner[37] suggest that by 2025, around 80 percent of the entire IT infrastructure, including various applications, technologies, and services, will be hosted in the cloud. While efforts have been made to enhance performance aspects such as storage capacity, networking efficiency, and hardware capabilities to meet the evolving demands of businesses, security measures for cloud infrastructure have predominantly relied on conventional

However, considering the potential repercussions of security breaches, such as erosion of customer trust and harm to business reputation, it becomes imperative to assess the effectiveness of these traditional security approaches. Are practices like monitoring network traffic for anomalies, regularly updating software to address known vulnerabilities, and deploying defensive tools such as Firewalls, Intrusion Detection Systems (IDS), and Anti-malware software enough to adequately detect and prevent attacks?

Moving Target Defense (MTD) is a proactive strategy used to deter potential attackers from exploiting system vulnerabilities by constantly changing the system's attack surface. This technique is part of a broader set of network obfuscation methods aimed at misleading attackers. MTD achieves this by altering system configurations, such as IP addresses, packet headers, or network topology, through techniques like IP shuffling[48], packet header randomization[115], or network topology shuffling [3]. Additionally, MTD involves migrating system platforms[36] to further confuse attackers.

The inspiration behind Moving Target Defense (MTD) lies in the recognition of the limitations of traditional static security[43] measures in the face of dynamic cyber threats. MTD draws upon the concept of continuously changing the attack surface to increase the complexity and uncertainty for adversaries, making it harder for them to successfully exploit vulnerabilities.

Many network systems currently in use are static, meaning they're configured once and remain unchanged for extended periods[82]. While this approach is logical for meeting functional require-

ments and cost optimization[52], it poses a significant security risk. The immutability of these networks provides potential attackers with an advantage, as they can monitor the network over time, gathering data on endpoints and traffic patterns without detection.

A compilation of 98 data breach statistics has been presented in a blog post, consolidating information on various aspects including types of breaches, industry-specific statistics, risks, costs, defense strategies, and prevention resources [98]. This compilation underscores the severity of the situation, highlighting it as a significant threat to the security of modern networks [61].

1.2 Research Statement

The aim of this thesis is: *To explore the suitability of attack graph-based security analytics for implementing moving target defense strategies.*

1.3 Contribution of the Thesis

The contributions of the thesis are as follows:

1. The three-tier attack graph (TAG) concept, which is an existing attack graph model. This model is designed to calculate the likelihood of a host being compromised based on the host's vulnerability data. The benefits of the Three-Tier Attack Graph (TAG) in Moving Target Defense (MTD) are discovered.

2. We have reviewed the most recent MTD methods, paying particular attention to how they operate, what kinds of threats they defend against, and whether or not they use threat models.

3. We've also identified the challenges with three important MTD components.

4. The proposed work SDN Network Configuration and Routing Experiment using Open vSwitch and Ryu Controller.

1.4 Outline of the Thesis

The structure of these begins with an abstract, followed by an Introduction in Chapter 1, which includes the overview, Inspirations behind Moving Target Defense, Research Statement, and Contribution of the Thesis. Chapter 2 briefly covers related works. Chapter 3 presents Classifying Moving Target Defenses, while Chapter 4 covers Attack Graphs with Three Tier Attack Graph. In Chapter 5, Challenges of Moving Target Defense (What to Move, When To Move How To Move). Chapter 6 includes details on the experiments conducted in preparation for this thesis. Lastly, Chapter 7 presents the conclusion of the work completed in this thesis and upcoming Future Works.

Chapter 2

Related Works

2.1 Moving Target Defense

2.1.1 Overview

Moving Target Defence, often known as MTD, is a cybersecurity tactic that uses a dynamic and moving defense system to shield a target from prospective threats. This technology is designed to improve network security by making it more challenging for attackers to find and use vulnerabilities. In 2009 [79], MTD emerged as a cyber-defense instance. This strategy constantly modifies the system's configuration and attack surface to make it more challenging for attackers to get past its defenses. The fundamental idea underlying MTD is to make the environment more complicated and uncertain, effectively turning the system into a moving target that attackers find difficult to hit precisely. MTD works to resist potential cyber-attacks by routinely changing attack surfaces, configurations, software implementations, or user access patterns. Typical static defenses used in traditional cybersecurity strategies include firewalls, intrusion detection systems, and access controls. Despite their value, these measures have several drawbacks.

2.1.2 History

The idea behind Moving Target Defense (MTD) draws from the military's use of maneuverability, which involves regularly changing the position and makeup of troops on the battlefield to make them challenging targets for enemy forces. Similarly, in MTD, the rules and location of a system are constantly modified to make it more difficult for adversaries to observe and comprehend its patterns. The application of this concept in cyber security was aimed at enhancing the resilience of networks and systems against cyber attacks, particularly with the growth of networks and cyber threats in the 1990s and 2000s [124].In Moving Target Defense (MTD), "shell games" involve constantly altering the location or appearance of a system's vulnerabilities or sensitive information to make it more difficult for attackers to locate and exploit them. This entails modifying the system's components and creating fake systems to divert and confuse attackers [125]. The phrase "Moving Target Defense" (MTD) was first introduced and proposed as a cyber security defense paradigm by the Networking and Information Technology Research and Development (NITRD) program during the U.S. National Cyber-Defense Summit in 2009 [124].

2.1.3 Definition

Zhuang et al. [114] Briefly define MTD as “constantly changing a system to reduce or move the attack surface available for exploitation by attackers.” Continuously altering a system’s configuration, architecture, or design to lessen the potential points of entry or vulnerabilities that attackers could exploit is referred to as reducing or shifting the attack surface. The term ”attack surface” refers to all the potential points of entry, manipulation, and system disruption for an attacker.

2.1.4 MTD Classification for Network

I present a classification for MTD Techniques based on the system layer. It was first proposed by Okhravi et. al [116]. Figure 2.1 shows the proposed MTD for the IOT Classification for the Network. Classifications are the following:

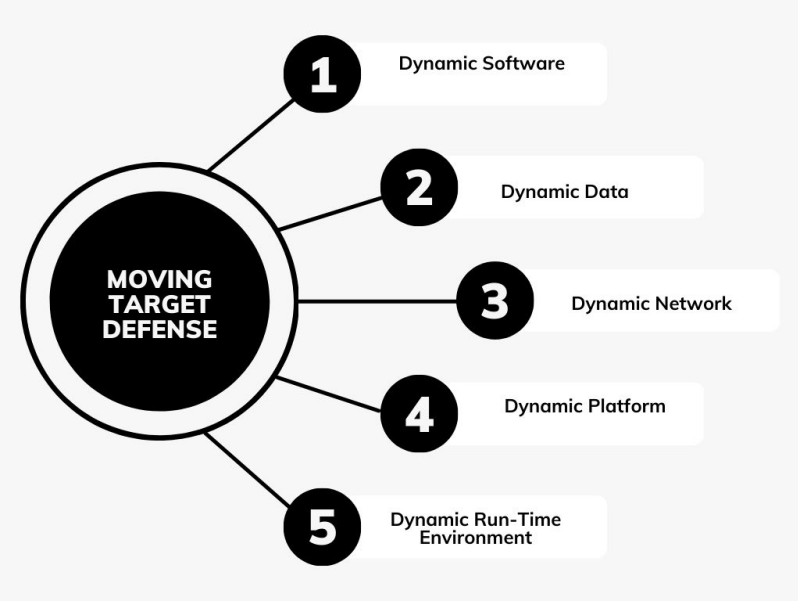


Figure 2.1: MTD Classification for Network

1. **Dynamic Software**: Moving target defense (MTD) uses dynamic software, which is the practice of regularly updating and upgrading software components, programs, and configurations to improve cybersecurity and make it harder for attackers to exploit vulnerabilities. e.g., binary object shuffling and application diversification.

2. **Dynamic Data**: As a cybersecurity tactic to protect against various cyber threats, dynamic data in moving target defense refers to the use of continually changing or evolving data and configurations. In moving target defense (MTD), the attack surface is purposefully made more unpredictable and challenging for adversaries to exploit. MTD tries to lessen the potency of cyberattacks by routinely changing the attack surface, which includes the network, system configurations, and data properties. By bringing diversity and unpredictability into the system, dynamic data plays a vital part in MTD by making it difficult for attackers to find and exploit vulnerabilities. i.e., the same semantics with different syntax.

3. **Dynamic Network**: In the context of moving target defense (MTD), a dynamic network is a network infrastructure that has been purposefully created and is routinely changed to introduce unpredictability and variability. In MTD, a dynamic network is intended to make it more challenging for attackers to identify weaknesses, carry out reconnaissance, and execute successful cyberattacks. e.g., protocols, addresses, and topology.

4. **Dynamic Platform**: A dynamic platform is the underlying environment, system, or infrastructure in which the moving target defense (MTD) tactics and strategies are deployed. This platform is made to continuously alter, vary, and be unpredictable in order to make it harder for attackers to find weaknesses and carry out successful cyberattacks. It is challenging for adversaries to obtain reliable information about the system and its components in MTD because the dynamic platform plays a critical role in establishing a moving and unpredictable attack surface. e.g., CPU architecture, OS, and virtual machine instances.

5. **Dynamic Run-Time Environment**: A dynamic run-time environment is one in which the conditions, configurations, and behaviors of software applications and systems are intentionally and continuously changed while they are being used. This strategy seeks to make the run-time environment more complex and unpredictable, making it harder for attackers to find weaknesses or launch effective cyberattacks. e.g., RAM addresses and instruction sets.

2.1.5 Classification of MTD

MTD approaches have been researched using a variety of classifications and criteria. We go over how the MTD techniques have been categorized in the literature in this part. We also analyze the similarities and contrasts between the MTD and deceit notions, as well as how they differ from one another [22]. Figure 2.2 Shows the Proposed Classification of MTD.

1. **Timeliness-Based MTD**

2. **Operation-Based MTD**

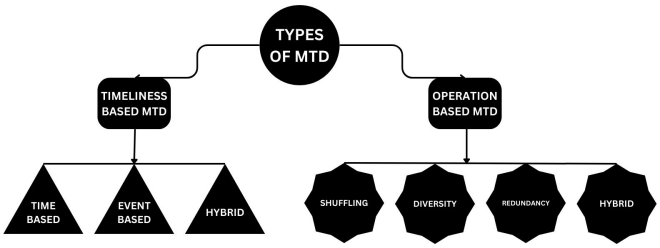


Figure 2.2: Types of MTD

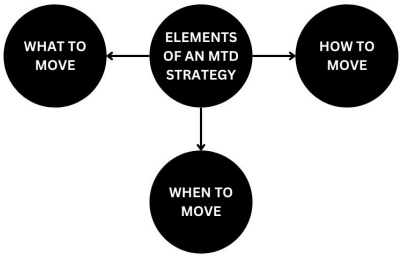


Figure 2.3: Elements of an MTD Strategy

2.1.6 Elements of an MTD Strategy

The protected target's vulnerabilities and attacked qualities remain the same before and after the MTD approach is used, but the target's static nature is destroyed since MTD is not only "changing," but also "moving." In order to accomplish the defense objectives, the vulnerabilities, and attacked traits can continuously change and attempt to stay one step ahead of the attacker. According to Cai et al[15], "What to move," "How to move," and "When to move,"' are the three essential

design questions that any MTD strategy needs to define. Figure 2.3 shows the Elements of an MTD Strategy.

1. **What to Move:** Identify the components of your system or network that can be changed to make an assault more difficult and unpredictable. The goal of MTD in IoT contexts is to diversify the attack surface and reduce the possible impact of successful attacks. These components could consist of:

    - **Device Behaviours:** Use variations in device behaviors to fool attackers trying to profile or exploit devices, such as changing data transmission patterns or sensor readings.
    - **Firmware and Software:** Update firmware and software versions on a regular basis to fix bugs and add modifications to the hardware's software stack.
    - **Network Topology:** To make it more difficult for attackers to map the network, introduce dynamic modifications to the network topology using techniques like Software-Defined Networking (SDN).
    - **Authentication and Authorization:** To lessen the impact of compromised credentials, rotate cryptographic keys, certificates, and access credentials.

2. **When to Move:** To balance operational effectiveness and security, movement pieces must be timed carefully. Think about the following elements:

    - **Event-Driven Moves:** These moves are triggered by specific events, such as anomalous behavior picked up by systems for detecting it or changes in the environment.
    - **Scheduled Changes:** Implement scheduled changes, taking into account the potential impact on device operation, to take place during maintenance windows or predetermined intervals.
    - **Threat Intelligence:** Adapt your moving frequency dynamically in response to new threats or adjustments in attacker tactics by using threat intelligence.

3. **How to Move:** It describes how the components or parameters of the system switch between different states. Given their finite resources and restrictive nature, IoT devices must carefully evaluate MTD implementation:

    - **Distributed Decision Making:** Allow devices to decide on their own how and when to move based on regional observations and specified rules.
    - **Centralized Control:** To maintain consistency and coordination among devices in situations with a centralized controller, manage and orchestrate moves from a central location.
    - **Automation:** Use scripts and tools for automation to speed up the process of altering items. Automation ensures consistency and lowers the possibility of human error.
    - **Monitoring and analysis:** Keep an eye out for abnormalities, strange behavior, and other indications that the system may be compromised. Identify patterns and prospective assaults by analyzing data.

2.1.7 Timeliness-Based MTD

Timeliness-based MTD categorizes MTD methods according to standards for deciding "when to move."Timeliness-Based Moving Target Defense (MTD) enhances system security by introducing time-dependent variations. Unlike traditional MTD, which increases system complexity, timeliness-based MTD dynamically adjusts system parameters at specific intervals or in response to temporal triggers. These changes, such as altering access controls or updating encryption keys, disrupt attacker activities. By aligning with attacker timing and behavior, timeliness-based MTD makes exploitation more challenging.

1. **Time-based:**The MTD interval, which can be either fixed or variable, is used in this method to determine when to do an MTD operation [125]. Moving Target Defence (MTD) uses a time-based technique that involves dynamically altering system configurations, settings, or behaviors over time to limit the attack surface and make it more difficult for potential attackers. MTD seeks to make it more difficult for attackers to foresee and use weaknesses by introducing differences at various time intervals. The breach of security could occur if the interval period is too long since the attacker would have had enough time to scan the system and then break in. In contrast, if the interval time is too short, the MTD will activate even if the system is not under attack, wasting defense resources and impairing performance [16].

2. **Event-based:** In Moving Target Defence (MTD), an event-based strategy entails dynamically altering system configurations, behaviors, or other characteristics in response to particular events or triggers. Various security incidents, threat indicators, or adjustments to the system's operating condition can serve as these triggers. The objective is to instantly modify the system's defenses to thwart prospective attackers. This method only does an MTD operation when a specific event takes place. Any indication that an attacker has gained access to a system or is attempting to conduct a specific attack can be considered an event. That is, events can initiate the proper MTD action if the defender is able to precisely forecast potential attackers. Attack prediction-based adaptive MTD techniques have been developed using machine learning [29], game theory [127], and control theory [88] in order to identify the important events that should cause the MTD action.

3. **Hybrid:** Moving Target Defence (MTD) employs a hybrid-based approach that incorporates several methods and tactics to fortify the defense against possible attackers. In order to maximize the advantages of each approach and develop a layered defense plan, it makes use of a combination of various MTD approaches, such as time-based, event-based, or other MTD strategies. Some MTD methodologies employ a hybrid approach. These methods [88],[60],[128] conduct MTD operations adaptively based on both the proactive and reactive event- and time-based MTD techniques, respectively.

2.1.8 Operation-Based MTD

Operation-based MTD divides MTD methods according to standards that specify "how to move." According to the nature of the operations, Hong and Kim [7] categorize MTD approaches into three categories: shuffling, diversity, and redundancy. Here are the specifics for each type:

Shuffling

Shuffling is a method frequently used in Moving Target Defence (MTD) to add unpredictability and diversity to the system or network, making it harder for attackers to anticipate and exploit weaknesses. We follow through with possible configuration changes for the system or network. Figure 2.5 shows the Shuffling Technique. Here's an overview of shuffling in MTD:

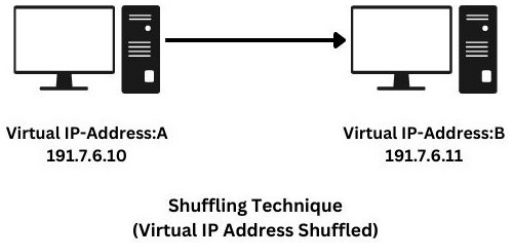


Figure 2.4: Shuffling Technique

1. **Network Topology Shuffling:** The basic idea behind this strategy is to regularly alter network routes in order to invalidate an attacker's path information. By utilizing SDN technology, Achleitner et al. [3] suggested a virtual topology generation framework against network scanning attacks. For SDN systems, Hong et al. [45] provided an ideal method of network reconfiguration based on the idea of shuffling-based MTD. They resolved a problem with assignment shuffles and demonstrated how diversifying network routing pathways increases network security.

2. **IP Shuffling:** IP shuffling is a particular kind of shuffling technique used in Moving Target Defence (MTD) to increase a system or network's security and resilience. It entails routinely switching or rotating the IP addresses linked to various system entities or network components. IP multiplexing (or demultiplexing) was used by Sharma et al. [96] to create IP shuffling MTD in an SDN context. In order to protect a large-scale network, MacFarland and Shue [17] employed a host IP address mutation in conjunction with an SDN controller that manages DNS exchanges. In order to prevent dangerous worms that accumulate data on victim targets in a networked system and make it challenging for attackers to locate new, vulnerable targets.

3. **Packet Header Randomization:** Unlinkability Through Random Identifier, or U-TRI, is a method put out by Wang et al. [115] that exploits the randomization of a packet header identifier to confound attackers.

4. **Software/Service Reconfiguration:** To improve a system's security and resilience, Moving Target Defence (MTD) uses the Software/Service Reconfiguration Shuffle approach. It entails routinely rearranging or changing the software or services used by various network endpoints or system components. In an IoT setting, Casola et al. [17] created an MTD technique based on the reconfiguration of devices whose firmware and cryptosystem versions are switched around. In order to create a secure Web by randomizing HTML components, Vikram et al. [108] developed a shuffling-based MTD approach on the application layer. The HTTP content/form page is used by the majority of bots that attack the Web. So, a suitable method of preventing Web bot attacks could be the randomization of those HTML elements and parameters.

5. **Port Hopping:** In order to effectively counter reconnaissance attacks by masking service identities and confusing potential attackers, Luo et al. [68] researched a port-hopping technique based on the idea of shuffling-based MTD. Based on the attack success probability as a function of the values of important design parameters, such as the size of a port pool, the number of probes, the number of vulnerable services, and the frequency of port hopping, this work measures the effectiveness of the suggested port hopping technique.

Diversity

Diversity is a fundamental principle in Moving Target Defense (MTD) that aims to increase the complexity and variability of a system or network to enhance its security and resilience. The concept of diversity in MTD involves introducing differences or variations in different aspects of the system, including configurations, components, behaviors, and responses. Figure 2.5 shows the Diversity Technique.

1. **Code Diversity:** When discussing code diversity in the context of MTD (Moving Target Defence), it's important to note that this term refers to the practice of introducing diversity and unpredictability into software or system components by using a variety of methodologies and strategies. To improve the system's overall security, the objective is to make it more difficult for attackers to anticipate or exploit weaknesses. An MTD method was created by Azab et al. [12] that randomly modifies the versions of a running program based on the idea of variety. The suggested approach breaks down a big program into parts (such as cells or tasks) that can be done by functionally equivalent versions. To make the suggested strategy more resilient, their approach has a recovery mechanism. Making a different variant choice at run-time makes it more difficult for an attacker to access the system and scan it. By affecting one variety that can be instantaneously replaced by another variant thanks to the recovery mechanism, the suggested strategy can even lessen the impact of an attack's success.

2. **Software Stack Diversity:** To improve service delivery and network resilience, Huang and Ghosh [46] presented a diverse MTD approach. They implemented a diversification strategy for virtual servers (VS), including operating systems, virtualization components, web servers, and application software. In terms of assault success likelihood, they assessed the proposed diversification strategy.

3. **Network Diversity:** By assessing the MTD technique based on a logical mission model, Zhuang et al. [14] evaluated the relationship between the diversity of network configurations and attack success likelihood. Based on experimental findings evaluated from the network

security simulator, NeSSi2, the authors investigated how the size of the network, the frequency of shuffling/adaptations, and the number of vulnerable machines affect the efficiency of the MTD technique.

4. **Programming Language Diversity:** To prevent code and SQL injection attacks, Taguinod et al. [102] concentrated on an MTD strategy based on the diversification of programming languages. In order to change the implemented language of the Web application without compromising or disrupting system functionality, the MTD technique was proposed to be applied in various layers of the Web application.

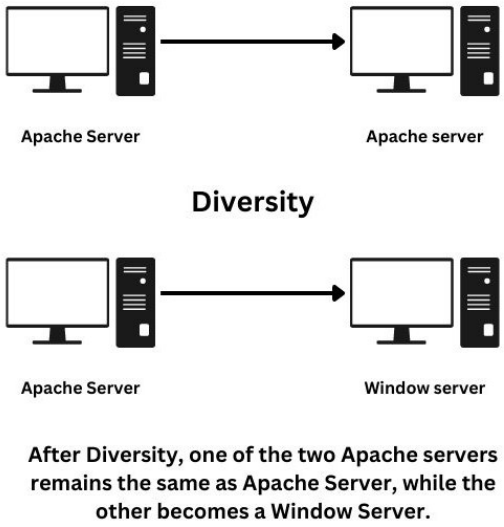


Figure 2.5: Diversity Technique

Redundancy

Redundancy in the context of MTD (Moving Target Defense) refers to the practice of incorporating redundant components or mechanisms within a system to enhance its resilience against attacks. Redundancy can help mitigate the impact of successful attacks by ensuring that alternative components or mechanisms are available to maintain system functionality or recover from an attack. Figure 2.6 shows the Redundancy Technique.

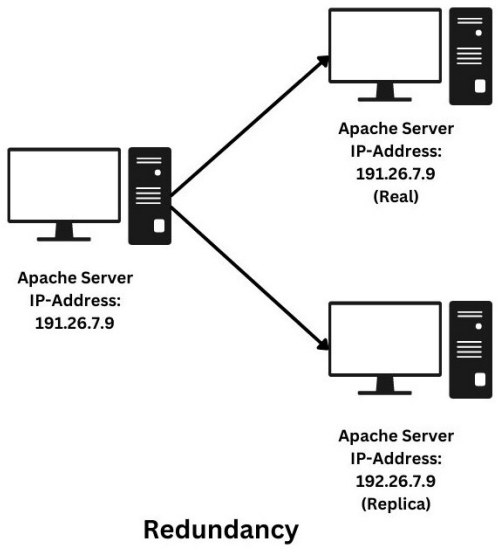


Figure 2.6: Redundancy Technique

1. **Redundant Authentication Mechanisms:** There is an additional degree of security when redundant authentication procedures are used. Using Multi-factor authentication (MFA), for instance, combines different authentication elements including tokens, smart cards, biometrics, and passwords. The additional factors offer an additional layer of security in the event that one authentication factor is compromised.

2. **Redundancy of Software Components:** By creating a self-protection model with architectural adaptation threat detection and mitigation, Yuan et al. [119] suggested a redundancy technique for Web servers to prevent malicious code injection assaults on Web servers. They made use of the so-called agreement-based redundancy, which offers copies of software com-

ponents during execution. The efficacy of the suggested MTD mechanism, however, was not examined in this experiment.

3. **Redundancy of Network Sessions:** By adapting the idea of MTD in terms of redundancy for cyber-physical system (CPS) contexts, Li et al. [66] suggested a traffic morphing mechanism. By maintaining a number of redundant network sessions with inter-packet delay distributions that are identical to those seen in regular network sessions, the proposed traffic morphing method is intended to safeguard CPS sessions. In order to reach its deadline, a CPS message will be broadcast via one of these sessions. This work demonstrated the suggested work's minimal complexity and excellent adaptability to the dynamics of CPS sessions while dynamically changing the morphing process.

Hybrid

In order to cooperate, hybrid MTD incorporates several MTD approaches from different types (such as redundancy, diversity, and shuffling). This section explains how MTD techniques work. can be integrated to improve system performance and security. Figure 2.7 shows the Redundancy Technique.

1. **Diversity-Redundancy(D+R):** In order to maximize system dependability, which is measured by availability, reliability, and service response time, Gorbenko et al. [39] designed a Web service based on the concepts of redundancy and diversity by offering varied replicas of Web services.

2. **Shuffling-Redundancy(S+R):** By using shuffling-based VM Live-Migration (VM-LM) and generating VM copies, Alavizadeh et al. [7] improved both the security and dependability of the system. The effectiveness of the combined MTD technique was evaluated by the authors based on the metrics of System Risk (Risk) and Reliability (R), which they used to gauge how well the created MTD technique performed in a cloud environment.

3. **Shuffling-Diversity(S+D):** Later on, Alavizadeh et al. [6] did a thorough analysis to assess the efficacy of two more combinations of shuffling and diversity MTD approaches by taking into account three crucial metrics: risk (Risk), attack cost (AC), and return on the attack (RoA).

4. **Shuffling-Diversity-Redundancy(S+D+R):** The three MTD methodologies (S+D+R) were integrated by Alavizadeh et al. [6] in the virtualization layer of a cloud in order to quantify the cloud's security level through thorough security analysis. The authors looked into the effectiveness of the combined MTD methodologies based on formal graphical security models for the cloud environment employing the three metrics (Risk, RoA, and AC) in addition to system availability (SA), much like [6].

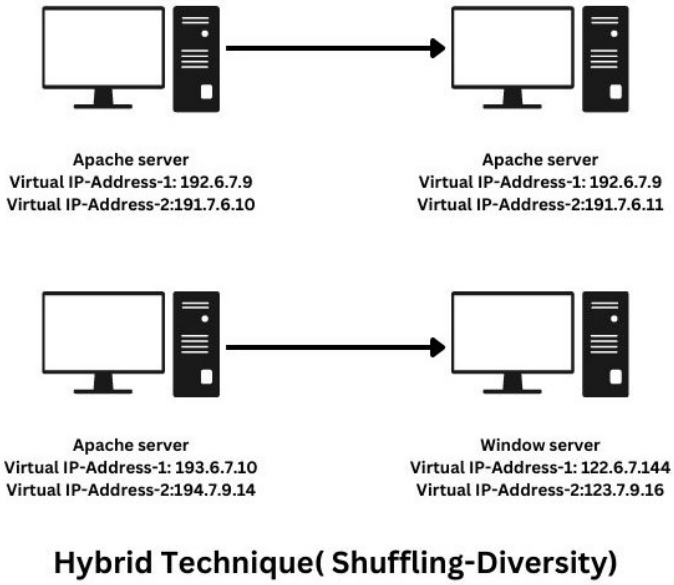


Figure 2.7: Hybrid Technique

2.2 Moving Target Defense for Internet of Things

2.2.1 Overview

The acronym "IoT" stands for "Internet of Things." It is a notion that refers to the internet-based interconnection of numerous physical items and equipment, including common household appliances, automobiles, industrial machinery, and other technology. These devices can gather and exchange data with other connected devices and centralized systems since they are built with sensors, software, and network connectivity. Making systems, networks, and applications more dynamic and unexpected for attackers is the goal of the cybersecurity tactic known as MTD, which tries to improve system, network, and application security. Due to the distinct difficulties and hazards posed by IoT devices, MTD is especially pertinent when used in an IoT setting. IoT devices are frequently underpowered in terms of computational power and security, making them easy targets for hackers. By bringing elements of variability and unpredictability into the IoT ecosystem, MTD for IoT can aid in reducing these risks. For IoT deployments to be more secure and resilient, MTD methods must be implemented. IoT technologies are used by many industries, including manufacturing, healthcare, transportation, energy, and agriculture, to increase productivity, collect data, and optimize operations. These sectors do, however, also present particular cybersecurity difficulties that MTD can aid in resolving.

2.2.2 History

The early 1980s are when the idea of the Internet of Things (IoT) first emerged. However, the late 1990s and early 2000s marked the beginning of the IoT's contemporary history as we know it today.

In the early to mid-2010s, the cybersecurity research community started showing interest in Moving Target Defense (MTD) tailored for Internet of Things (IoT) devices. This interest coincided with the rapid rise of IoT technology and a growing recognition of the security risks associated with interconnected devices. Consequently, researchers began delving into inventive strategies to tackle the distinct security challenges inherent in IoT ecosystems.

2.2.3 Definition

- **IoT:** The network of physical objects—"things"—that are integrated with sensors, software, and other technologies for the purpose of communicating and sharing data with other devices and systems over the internet is referred to as the Internet of Things (IoT)[73].
- **IIoT:** Industrial IoT (IIoT) is the use of IoT technology in industrial settings, particularly in relation to the instrumentation and control of sensors and other devices that use cloud-based services. See this Titan use case PDF for a solid illustration of the IIoT.

2.2.4 MTD in Various IoT Industries

1. **Manufacturing and Industrial IoT(IIoT):** To stop attackers from tracing the network and locating crucial assets, attackers must constantly change the network topology and communication channels between industrial IoT devices.
2. **Healthcare IoT:** To prevent unauthorized access and data breaches, frequently alter the identities and access credentials of medical IoT devices. Utilise MTD methods to protect the confidentiality and integrity of patient data when it is shared between several healthcare organizations.
3. **Transportation IoT:** To prevent future hacks, implement MTD by routinely updating communication protocols and encryption techniques in connected vehicles and transportation networks.
4. **Energy and Utility IoT:** To avoid unauthorized access and tampering, smart grids and utility systems should routinely alter the identities and network addresses of IoT devices. To stop the propagation of threats, create automated MTD actions to isolate infected devices or areas of the network.
5. **Agriculture IoT:** To prevent attackers from correctly predicting crop and soil monitoring patterns, periodically relocate agricultural IoT sensors and equipment. Implement lightweight MTD techniques that don't unduly tax agricultural IoT devices by taking into account their resource constraints.

2.2.5 Security Issues in MTD for IoT

1. **Resource Constraints:** Due to their limited computational power, memory, and energy resources, numerous IoT devices face significant constraints. The integration of dynamic

defense mechanisms like MTD may exacerbate these limitations, potentially burdening IoT devices with additional processing and memory demands. Consequently, this could lead to performance degradation and compromise the functionality of these devices.

2. **CIA Triad:**The confidentiality, integrity, and availability (CIA) trinity can be used to guide some policies in the implementation of security goals [90]. When it comes to security, confidentiality is the first thing that springs to mind because it is impossible to imagine a secure system where sensitive data is held in plain text. Users' sensitive data is hidden via data perturbation techniques like encryption, data obfuscation, and secret sharing. Another crucial component of the trinity is preserving the integrity of data at rest, code running on the device, or a network packet.

3. **Interoperability:**IoT devices originate from a variety of manufacturers and often operate on different operating systems and communication protocols. This diversity poses a challenge in ensuring seamless interoperability between MTD solutions and these devices. As a result, compatibility issues may arise, potentially introducing vulnerabilities into the system.

4. **Dynamic Environment:**IoT environments undergo constant change, with devices frequently connecting and disconnecting from the network. This dynamic nature presents challenges in adapting MTD strategies to these fluctuations while maintaining uninterrupted protection against emerging threats.

5. **Attack Surface Complexity:**The integration of dynamic components into IoT systems could inadvertently expand the attack surface, offering adversaries additional avenues to exploit vulnerabilities. It's essential to exercise caution to prevent MTD implementations from inadvertently introducing fresh security risks.

6. **Scalability:**IoT environments frequently comprise a diverse array of devices with varying characteristics. Coordinating and managing MTD strategies across this heterogeneous landscape can be complex and challenging, especially with the ongoing proliferation of connected devices.

The confidentiality, integrity, and availability (CIA) trinity can be used to guide some policies in the implementation of security goals [90]. When it comes to security, confidentiality is the first thing that springs to mind because it is impossible to imagine a secure system where sensitive data is held in plain text. Users' sensitive data is hidden via data perturbation techniques like encryption, data obfuscation, and secret sharing. Another crucial component of the trinity is preserving the integrity of data at rest, code running on the device, or a network packet.

2.3 Software-Defined Networking(SDN) Based MTD

2.3.1 Overview

The goal of software-defined networking (SDN), a type of network design is to make networks more adaptable, scalable, and programmable by dividing the network's control plane from the data plane, which actually forwards information. This division enables network administration to be automated and centralized, making it simpler to respond to shifting needs and improve network performance. SDN is a new technology that offers a flexible and affordable method for controlling and managing

networks. The primary distinction between SDN and the conventional network architecture is the proposal to separate the data and control planes.

Moving Target Defense strategies applied to Software-Defined Networking (SDN) introduce dynamic and adaptive security measures to mitigate cyber threats in network environments. Unlike traditional networking approaches, where network configurations are relatively static, SDN enables the centralized control and programmability of network infrastructure through software-defined controllers. By leveraging SDN's flexibility and programmability, MTD can enhance network security by continuously changing network parameters and configurations, thereby reducing the attack surface and increasing the complexity for potential adversaries.

2.3.2 History

The beginnings of SDN can be found in the early 2000s. By permitting centralized network control, Open Flow, which was released in 2008, created the framework. The Open Networking Foundation was established in 2011, which sparked cooperation and standardization across the sector.

The integration of Moving Target Defense (MTD) principles into Software-Defined Networking (SDN) environments has emerged relatively recently, evolving alongside advancements in both SDN technology and cybersecurity tactics. Though precise timelines may differ, the concept of incorporating MTD strategies into SDN began to attract significant interest in the early to mid-2010s.

2.3.3 Definition

The term "software-defined networking" (SDN) refers to a method of networking where traffic on a network is controlled by software-based controllers or application programming interfaces (APIs) that talk to the underlying hardware infrastructure.

2.3.4 Key Components

1. **SDN Controller:** the central processing unit that, in accordance with the network's policies and standards, decides how data traffic should be sent. OpenDaylight and OpenContrail are two instances of SDN controllers.

2. **Southbound APIs:** These are the interfaces that link the data plane network hardware (such as switches, routers, and so on) to the SDN controller. They make it possible for the controller to interact with and manage the hardware.

3. **Northbound APIs:** These APIs give programs and services a channel of communication with the SDN controller. They give programmers the ability to design unique apps that can dynamically configure the network according to particular needs.

4. **Data Plane:** Actual data packet forwarding within a network is handled by the data plane. It makes judgments on packet switching, routing, and forwarding. Based on the routing choices made by the control plane, the data plane is responsible for efficiently transporting data packets between network nodes.

5. **Control Plane:** High-level choices regarding how data should be routed in the network are made by the control plane. It establishes network regulations, administers routing tables, and

makes routing decisions. To exchange routing data and create a routing table, the control plane interacts with other network devices.

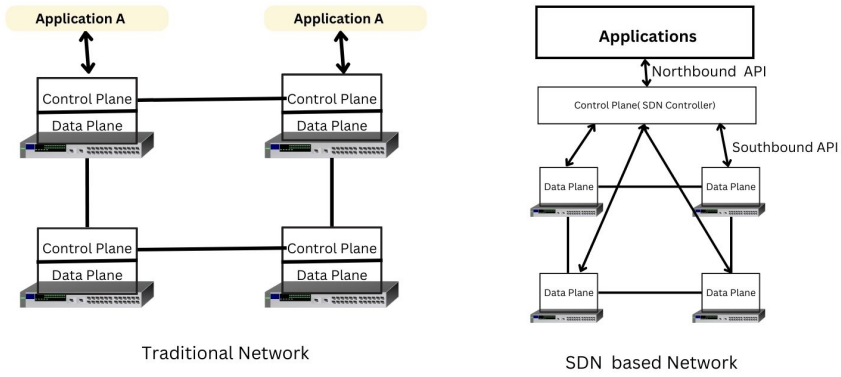


Figure 2.8: Traditional Network V/s SDN-based Network

2.3.5 Advantages of SDN-Based Network Over Traditional Network

1. **Centralized Network Control:** The control plane and data plane are separated by SDN, enabling centralized control of the whole network from a single controller. The management of the network is made easier by centralization, which also enables more effective and uniform regulations throughout the network.

2. **Programmability:** Because SDN switches are so programmable, network managers can establish and alter network policies and behaviors using software as opposed to hardware. This adaptability enables quick changes in response to shifting network requirements.

3. **Automation:** Automation of network provisioning, configuration, and management tasks is made possible by SDN. As a result, manual errors are decreased, network changes are expedited, and overall network agility is improved.

4. **Reduce Capital and Operational Cost:** The flexibility and automation that SDN infrastructure enables can result in cost savings over time, even though the initial setup may need expenditure. Lower operational costs are the result of less need for costly, proprietary gear and easier network management.

5. **Easier Network Testing and Development:** SDN lowers the risk of network interruptions by facilitating simple network testing and development of new features or services in isolated settings.

2.3.6 SDN Environment on MTD Techniques

- **Control Plane MTD:**This approach involves dynamically adjusting routing protocols, network policies, and controller configurations to introduce variability into the control plane. By implementing these modifications, SDN controllers create a more challenging environment for attackers attempting to anticipate and exploit network behavior. This proactive measure enhances network security by increasing the complexity encountered by potential adversaries.

- **Data Plane MTD:**SDN switches and forwarding devices employ methods such as flow migration, network obfuscation, and policy randomization to dynamically adjust the forwarding and processing of traffic. These techniques aim to decrease the visibility and predictability of network traffic patterns, rendering it more challenging for attackers to comprehend and target specific network components. Data Plane MTD strengthens network resilience by introducing layers of unpredictability to network operations.

- **Application Layer MTD:**SDN applications and services incorporate Moving Target Defense (MTD) principles to adjust access control policies, encryption settings, and authentication mechanisms based on real-time threat intelligence and network conditions. This dynamic adaptation of security measures at the application layer empowers organizations to respond effectively to evolving cyber threats and protect critical assets. By leveraging MTD principles in SDN applications, organizations can proactively adjust their security posture to mitigate risks and improve their overall resilience against sophisticated adversaries.

- **Monitoring and Detection:**Network telemetry data and real-time analytics are used by SDN-based Moving Target Defense (MTD) systems to find anomalies and possible security risks in the network. Organizations can successfully manage risks by proactively detecting any questionable conduct, instantly raising alerts, and triggering automated responses through continuous monitoring of network operations. By facilitating quick identification and reaction to possible threats, this real-time monitoring and detection capabilities greatly strengthens the network's overall security posture and increases its resilience against cyberattacks.

- **Adaptive Response Mechanisms:**SDN controllers possess the ability to dynamically adjust network defenses in response to the severity and nature of detected threats. This adaptive response may entail measures such as isolating compromised devices, rerouting traffic to secure paths, or implementing additional security measures immediately. By promptly adapting to emerging threats through these responsive mechanisms, organizations can effectively contain and mitigate security incidents, minimizing the impact of cyber attacks on network operations. This agility in adapting network defenses empowers organizations to proactively defend against threats and uphold the integrity and availability of their network infrastructure.

- **Integration with Threat Intelligence:**Moving Target Defense (MTD) systems based on SDN have the ability to interact with outside threat information feeds in order to identify new threats early on and modify network defenses appropriately. Organizations may anticipate changing cyber threats, improve their protection plans, and fortify their networks against

discovered vulnerabilities and attack vectors by integrating threat intelligence sources. Organizations may improve their overall security posture and proactively address potential threats by staying up to date with the ever-evolving threat landscape thanks to this integration.

2.4 SDN-Based MTD for IoT

2.4.1 Overview

SDN and MTD in combination can offer a potent method for safeguarding IoT networks. While MTD offers an extra layer of protection by making the network environment more difficult for attackers to explore, SDN enables dynamic control of network resources. In IoT installations, this combination can help safeguard sensitive data and important infrastructure. For IoT devices that are frequently difficult to access, SDN architecture has been viewed as a cost-effective solution that can simplify network management and dynamic configuration. When SDN is used, it can open the door for the use of MTD techniques in IoT applications. In figure:2.9 shows multiple concepts that would be beneficial for IoT Security and Defense.

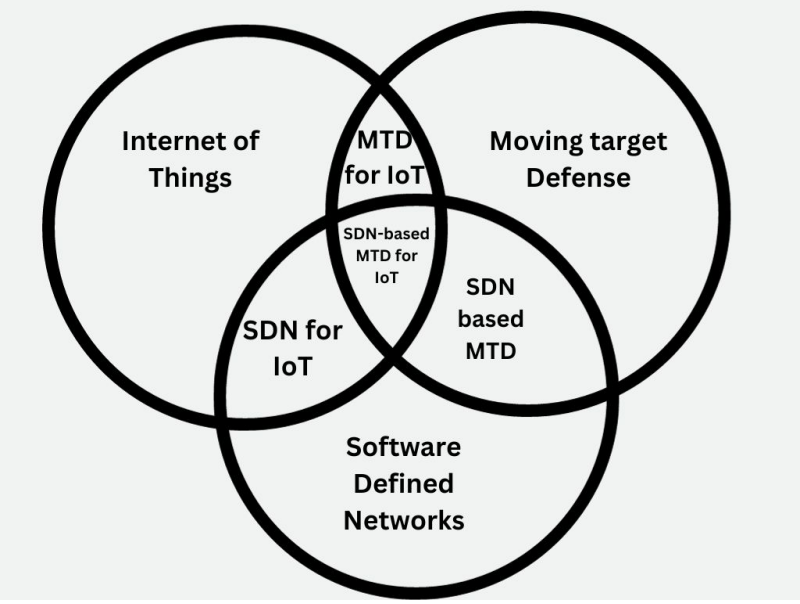


Figure 2.9: Venn Diagram of Three Technologies: IoT, MTD, and SDN

2.4.2 Advantages of SDN-Based MTD for IoT:

- **Enhanced Security:** Dynamic Attack Surface: MTD approaches frequently modify the attack surface, which makes it difficult for attackers to take advantage of flaws.
- **Adaptive and Contextual Security:** By giving MTD systems access to contextual data about network traffic, SDN enables them to decide when and how to modify network parameters.

- **Isolation and Segmentation:** SDN may establish distinct segments for various IoT device types, and MTD can improve segmentation even further by modifying the segmentation structure over time.
- **Reduced Attack Dwell Time:** SDN enables immediate reactions to security events discovered by MTD, decreasing the amount of time it takes for attackers to exploit IoT devices.
- **Reduced Operational Overhead:** Automated responses to security events are made possible by SDN and MTD, which ease the operational burden on security personnel.

2.4.3 Challenges of SDN-Based MTD for IoT:

- **Complexity:** It can be challenging to combine MTD and SDN, and the compatibility of various MTD and SDN solutions needs to be carefully handled.
- **Network Latency:** Frequent network configuration changes have the potential to increase latency, which may not be ideal for real-time applications like autonomous vehicles or IoT in healthcare.
- **Resource Intensiveness:** MTD and SDN implementation can be resource-intensive and call for hardware that is capable and effective management to prevent performance problems.
- **Cost:** Hardware, software, and human resources may need to be invested significantly in order to implement both MTD and SDN solutions.
- **Risk of Overcomplication:** If MTD and SDN are used carelessly, they could complicate the security architecture by adding more points of failure and expanding the attack surface.

2.5 Attack Modeling Techniques

Organizations use cutting-edge infrastructure management technologies to decrease the attack surface and adhere to best practices including software patching, hardening, and log analysis. However, cunning adversaries can compromise network assets by employing techniques like customized malware and zero-day attacks, which are frequently hard to identify or stop with intrusion detection systems and antivirus software. To effectively implement intelligent cyber defenses, gathering data (referred to as the threat model)[10],[94] regarding the attack methodology employed by an enemy is important. Advanced Persistent Threats (APTs) are highly trained and well-funded adversaries that launch targeted and sophisticated cyberattacks, frequently with the support of nation-states or organized cybercrime gangs. Persistence is a defining characteristic of Advanced Persistent Threats (APTs) [104],[10],[21] wherein the attackers strive to sustain unapproved entry into a targeted system or network for a prolonged duration. These threats usually combine sophisticated tactics, meticulous planning, and ongoing target surveillance.

2.5.1 Advanced Persistent Threats

Advanced Persistent Threat describes a class of extremely complex and focused cyberattacks carried out by competent and well-funded adversaries, frequently with the assistance of nation-states, criminal groups, or hacktivist organizations. Persistence is a defining characteristic of Advanced Persistent Threats (APTs), wherein the attackers strive to sustain unapproved entry into a targeted system or network for a prolonged duration[94]. Different phases of APT are followed below in Fig.10

Different Phases of APTs(Advanced Persistent Threats)

1. **Reconnaissance:** Gather information about the target organization. thorough investigation of the target, key person identification, comprehension of the network architecture, and evaluation of potential vulnerabilities[99].

2. **Weaponization:** Develop or acquire the tools and techniques for the attack. To prepare the cyberweapons, tasks such as malware development, exploit generation, payload construction, and others are carried out[94].

3. **Delivery:** Introduce the weaponized payload into the target environment. Deliver the malicious software to the intended recipient, frequently by means of drive-by downloads, spearphishing emails, or other delivery systems[99].

4. **Exploitation:** Exploit vulnerabilities to gain unauthorized access. Use the malware that was transmitted to take advantage of holes in the target system or network to gain access and create a foothold[10].

5. **Installation:** Establish a persistent presence in the compromised environment. Install malware, rootkits, or backdoors[99] on the hacked system to keep control and access for a long time.

6. **Command and Control (C2):** Establish communication channels with the compromised systems[99]. Provide means for the attacker to take command of and obtain information from the compromised systems via a remote connection.

7. **Action on Objectives:** Execute the primary goals of the APT, such as data exfiltration, espionage, or disruption. Execute the specified goals, such as gathering information, moving laterally, increasing privileges, and eventually having the intended effect on the target organization[94].

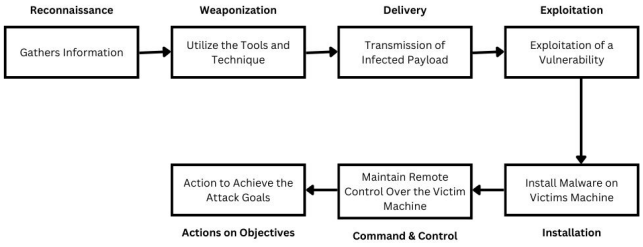


Figure 2.10: Different Phases of An APT

Key Features of APTs

1. **Targeted Approach:** APTs concentrate on specific organizations, sectors, or even governments. The attackers carry out in-depth reconnaissance to learn as much as possible about the target and modify their attacks accordingly.

2. **Multi-Stage Attacks:** Multiple stages are usually involved in APTs, including installation, exploitation, weaponization, reconnaissance, command and control, and action on goals. Within the overall attack life-cycle, each stage has a distinct function.

3. **Zero-Day Exploits:** Zero-day exploits, which target vulnerabilities that are not yet known to the software vendor or the general public, are frequently used by APTs. Attackers benefit from this since security updates and fixes cannot be accessible.

4. **Social Engineering:** Social engineering tactics are frequently utilized by APTs to manipulate individuals within their target organization. This could involve spear-phishing, phishing emails, or other techniques to obtain or retrieve private data.

5. **Data Exfiltration:** Stealing sensitive data, including trade secrets, intellectual property, and classified information, is one of APTs' main objectives. Exfiltrating stolen data to external servers under the control of the attackers is common.

2.5.2 Reconnaissance

Reconnaissance, also known as a scanning attack, gathers critical information like IP addresses, open ports, and OS versions to prepare for subsequent attacks, especially targeting vulnerable applications. It can be passive (using public resources) or active (direct interaction). Active reconnaissance is faster but riskier. Tools like Nmap [59] Aircrack-ng [55], and Nessus [13] aid in target scans, useful not just for attackers but also for defenders for network security.

2.5.3 Denial of Service (DoS)

Denial of Service (DoS) is an attack that disrupts a device's normal functions. A more severe form, Distributed Denial of Service (DDoS), utilizes numerous hosts, often forming a botnet, to overwhelm target networks or servers. This flood of requests, often using ICMP or ACK packets, renders the target unable to process them in real time. A variant, the crossfire attack, strategically targets specific network links, making detection challenging. DDoS protection is available from vendors like Cloudflare [78], offering solutions to detect, and drop malicious traffic, reroute it, and block offending IPs.

2.5.4 Zero-Day

Zero-day vulnerabilities, also referred to as 0-day, describe security flaws that are exploited by attackers before they are patched. In some cases, these vulnerabilities may go undetected for months or even years, providing attackers with the opportunity to exploit systems that are otherwise considered secure without any intervention or mitigation. According to a report[89], there was a significant increase in the exploitation of zero-day vulnerabilities in 2012 compared to previous years. These vulnerabilities were primarily utilized by state actors to spy on large companies, but financially motivated attacks also saw a rise. Another article[112] aims to evaluate the security implications of unknown vulnerabilities in computer systems.

2.6 Defense Methods

In this section, we offer a concise summary of different defense strategies and illustrate their effectiveness across different stages of a Cyber Kill Chain. This discussion will aid in comprehending how certain MTD techniques incorporate these conventional defenses as foundational elements.

2.6.1 Proxy Filtering

Proxy filtering[94] is the practice of employing a proxy server to manage and regulate internet traffic. This entails capturing requests from clients, transmitting them to the intended server, and simultaneously scrutinizing and filtering the content within both the requests and responses. A reverse proxy server, such as Nginx [86], can function as an intermediary between an intruder on the public network and the private network. Using a proxy can help protect the genuine network from the intruder's activities.

2.6.2 Intrusion Detection System

An Intrusion Detection System (IDS)[94] is a security tool designed to monitor network or system activities for indications of malicious behavior or policy violations. Upon detecting unauthorized access, misuse, or potential threats, an IDS alerts administrators or security personnel. Two kinds of IDS agents can be deployed within a network under attack: Network-based IDS (NIDS) [77] and Host-based IDS (HIDS) [62]. NIDS like Bro[77] and Snort[87] employ signature-matching methods relying on recognized attack patterns. They can identify incoming network traffic as potentially malicious. HIDS like audits [33] and tripwire [62] inspect the Indicators of Compromise on the physical server or virtual machine(VM) to detect malicious behavior, such as attempts at privilege escalation by regular users.

2.6.3 Web Analytics

Web analytics involves gathering, analyzing, and reporting website and online platform data to enhance user behavior and website performance. It includes tracking metrics like website traffic, user engagement, conversion rates, and customer behavior to improve online initiatives and digital strategy [94]. A significant volume of security-related data exists on the web, spanning social engineering sites, phishing platforms, and dark web forums. This includes discussions regarding specific target products or companies, as highlighted by Glass and Colbaugh [38].

2.6.4 Intrusion Prevention System

An Intrusion Prevention System (IPS) actively monitors network or system traffic, identifies potential security threats or malicious activities, and promptly takes action to prevent unauthorized access or exploitation. Network security threat prevention technologies like IPS[64] examine network traffic flows to detect and prevent vulnerability exploits, typically in the form of malicious inputs to target applications or services. Deployed behind the firewall, IPS provides an additional layer of analysis and takes automated actions based on traffic pattern matches, distinguishing it from IDS[94].In contrast to IDS, IPS automatically takes action based on traffic pattern matches, such as

1. Dropping malicious traffic packets.
2. Resetting connection.
3. Blocking request from a source.
4. Sending an alarm to the admin.

2.6.5 Data Execution Prevention

Data Execution Prevention (DEP)[100] functions similarly to a host-based intrusion prevention system, aiming to safeguard the host from attacks leveraging code injection as their attack vector.DEP, available in hardware or software forms, prevents malicious code execution on the host by monitoring program behavior and ensuring safe memory usage[94].

2.6.6 Access Control List

An Access Control List (ACL) [84] is a security tool utilized in computer systems and networks to regulate access to resources like files, folders, devices, or network services. It dictates which users or groups have permission to access specific resources and the actions they can perform, such as read, write, execute, or delete. ACL entries typically include a subject (user or group), an object (resource), and a set of associated permissions or access rights. ACLs[94] can be implemented at various enforcement points within a network. For instance, Netfilter examines network traffic and makes decisions on whether to permit or deny it based on properties like source/destination IP address, port number, and protocol.

2.6.7 Tarpit

A tarpit[56] is a network security measure that intentionally delays potential attackers by slowing their network connections. It responds slowly to connection attempts from suspicious or malicious sources, causing the attacker's system to be occupied for an extended period. Tarpits are commonly deployed to thwart automated attacks like worms or bots, consuming the attacker's resources and hindering further attacks. This strategy can reduce the impact of malicious activity and provide security personnel with additional time to respond effectively.

2.6.8 DNS Redirect

DNS redirect, also referred to as DNS redirection, is a technique used to reroute DNS queries from their intended destination to another IP address or domain. This redirection can occur through methods like altering DNS records, adjusting DNS server configurations, or utilizing malicious software. While DNS redirect can serve legitimate purposes like load balancing or content optimization, it can also be exploited for malicious activities such as redirecting users to phishing websites or malware-infected servers. These attacks pose significant risks to network security and user privacy, potentially leading to unauthorized access to sensitive data or the installation of malicious software. To mitigate these risks, implementing robust DNS security measures such as DNSSEC (DNS Security Extensions) and DNS filtering is crucial.

2.6.9 Honey-Pot

A honeypot is a security tool used to detect, deflect, or study unauthorized attempts to access information systems. It mimics a legitimate computer, data, or network site, enticing potential attackers to interact with it. Honeypots are closely monitored to reveal attackers' tactics, techniques, and intentions. They can be deployed internally or externally to an organization's network and serve various purposes, including gathering intelligence on attackers, identifying new attack methods, and diverting attention away from critical systems. Honeypots come in different types, such as low-interaction and high-interaction, offering varying levels of simulation and interaction with attackers. Honeypots such as Cowrie [63] can help in better understanding the attacker's tools and tactics.

2.6.10 Anti-Virus

Antivirus (AV)[94] software safeguards hosts from various malicious programs like viruses, spyware, worms, rootkits, and key-loggers. AVs are categorized into two types: signature-based and behavior-based. Signature-based AVs compare programs against a database of known malicious signatures, while behavior-based AVs analyze program behavior by running them in a controlled environment, typically referred to as a sandbox.

Chapter 3

Classifying Moving Target Defenses

Moving Target Defense (MTD) aims to regularly alter the configuration of system components in a random manner. By doing so, it prevents attackers from relying on reconnaissance data collected earlier, as the system will have changed by the time they launch their attack. This heightened uncertainty raises the cost and difficulty for attackers to exploit the system effectively.

An MTD is defined by a three-part tuple: M for the movement strategy guiding system motion, T for the timing function indicating when switching actions occur, and C for the configuration space representing the range of configurations between which the system switches. These aspects categorize different MTDs effectively.

- **C->** What to Move
- **T->** When To Move
- **M->** How To Move

3.1 What to Move?

This section highlights prevalent Moving Target Defense (MTD) techniques identified in the reviewed literature. Common approaches include IP, port, route, and host mutations, along with the utilization of decoys and proxy servers. Additionally, several studies leverage MTD architecture to dynamically deploy traditional defense systems such as IDS. The subsequent part introduces each surveyed paper, furnishing brief and succinct one-paragraph descriptions for them.

3.1.1 Address and Port Mutation

Address and port mutation is a commonly employed strategy within MTD frameworks. Its objective is to bolster security by dynamically altering the IP addresses and ports linked to networked devices. By implementing this technique, the predictability for attackers to target specific devices or services is reduced, consequently enhancing the network's overall resilience against cyber threats.

1. In their paper cited as [48], the authors propose utilizing Software-Defined Networks (SDNs) for managing IP address translation for hosts within a network. In this setup, each host is allocated both a permanent (real) IP address and a temporary (virtual) IP address. The primary aim of this system is to ensure transparency for end-users, who only need to interact with the network using their real IP addresses, without being concerned about the virtual ones. This approach relies on two essential components: a switch with the ability to translate real IP addresses to virtual ones, and a controller responsible for coordinating this process and maintaining an up-to-date mapping of real-to-virtual address assignments across the entire network.

2. The MTD (Moving Target Defense) method outlined in reference [117] employs IP hopping, which is activated either by a timer or a detector. It utilizes a lightweight convolutional neural network (CNN) to identify potential threats. The architecture follows a multistep approach to determine its course of action. Initially, it samples packets, capturing each probe packet and optionally subsequent packets. These samples undergo processing where irrelevant information unrelated to scanning attacks is filtered out. Subsequently, the refined data is fed into a neural network, which has the capability to initiate address mutation if the packets are identified as malicious.

3. The study described in reference [40] introduces an innovative Moving Target Defense (MTD) framework leveraging Software-Defined Networks (SDNs) to establish an IP randomization strategy. In this architecture, each host in the network undergoes IP mutation at varying intervals, customized based on individual requirements. This adaptability ensures flexibility and responsiveness to current network conditions. Notably, the mutation interval can be adjusted dynamically, influenced by factors such as the existence of an active session between two hosts. During such sessions, the virtual IPs of participating hosts remain unchanged to maintain network stability and performance.

4. In their paper referenced as [72], the authors introduce an MTD (Moving Target Defense) technique built upon Software-Defined Networking (SDN) to facilitate transparent IP mutation. In this method, as a packet enters the network and reaches the initial switch along its path, the SDN controller takes charge of assigning new, randomized network IPs for both the source and destination addresses. As the packet traverses subsequent switches along its route, each switch forwards a message to the controller. The controller then assesses whether the current switch is the last one before reaching the destination node. If it isn't, the packet is simply forwarded as usual. However, if the current switch is indeed the last one before reaching the destination, the controller restores the original IPs in the packet, ensuring seamless delivery to the intended recipient.

5. In [96], the authors propose a method that employs multiple SDN controllers for managing IP shuffling. This strategy enables the establishment of a decentralized architecture, where each

controller communicates with the others, ensuring redundancy and fault tolerance. In the event of a controller failure, the workload can be distributed among the remaining controllers, ensuring continuous operation.To facilitate communication and coordination, the system utilizes two communication channels: one for inter-controller communication and another for communication between controllers and switches. This architecture enhances the scalability and resilience of the system, allowing efficient programming of Moving Target Defense (MTD) mutations to enhance network security.

6. The sole article discussing port mutation is [23], which introduces an SDN-based Moving Target Defense (MTD) method employing port hopping as a defensive measure. In this technique, the SDN controller plays a pivotal role by integrating data from the Common Vulnerability Scoring System (CVSS) provided by the vulnerability scanner and active protection insights from the Intrusion Detection System (IDS). By analyzing this information, the controller identifies virtual machines (VMs) with the highest security risks. If the threat level surpasses a predefined threshold, the controller initiates port hopping for the affected host as a proactive security measure.

7. In [19], an innovative SDN-based Moving Target Defense (MTD) method is introduced, focusing on frequent and rapid IP hopping across every node along the packet's path. The responsibility of the SDN controller is to furnish each switch in the network with two distinct random seeds. One seed is employed to seed the pseudo-random generator, while the other facilitates synchronization. Additionally, there exists a third seed, referred to as the IP seed, which is locally generated on each node and shared among all network nodes. This IP seed is essential for routing calculations. In conjunction with the pseudo-random generator seed, the IP seed contributes to the generation of new IP packets. After each hop, both destination and source addresses undergo randomization. The synchronization seed is utilized to create hash chains, which are precomputed and stored on every network node. These hashes are subsequently employed in reverse order to generate a specific IP address. This address is then utilized in an update packet to synchronize all nodes along the forwarding path. This comprehensive approach ensures dynamic and robust IP hopping, enhancing the network's security posture.

8. In [70], a novel Moving Target Defense (MTD) technique employing port hopping is presented, offering per-packet synchronization without the necessity of additional communication channels. This synchronization is achieved through the calculation of a keyed-hash message authentication code (HMAC) of the message. The HMAC not only ensures data integrity on the receiver's end but also serves as an input for encoding and decoding algorithms. By leveraging this HMAC, the system can calculate virtual addresses and ports to be assigned to each message and subsequently reverse this process to recover the original network data. However, it's important to note that this approach relies on the utilization of a shared session key, which must be known by both communicating parties for successful operation.

9. In [74], a novel Markov game framework tailored for the analysis of Moving Target Defense (MTD) strategies is presented. This framework exhibits versatility, as it is applicable across a wide array of techniques and can gauge the efficacy of each method based on its unique parameters and the overall system configuration. The paper successfully demonstrates the application of this framework to evaluate an IP-hopping MTD technique designed for concealing both single and multiple targets. This showcases the capability of the Markov game

framework to assess the strength and effectiveness of MTD approaches in various scenarios and settings.

10. In their paper referenced as [4], the authors present a bio-inspired approach to Moving Target Defense (MTD) that leverages Software-Defined Networking (SDN) to enhance the security of distributed systems. This technique involves the utilization of short-lived nodes during computational tasks, which are subsequently discarded. However, these nodes have the capability to reappear later with altered properties.The algorithm employed in this method facilitates the seamless movement of nodes between hardware and hypervisors, enabling functionalities such as IP hopping and the modification of node characteristics, such as switching to a different operating system. Moreover, the proposed system integrates proactive node monitoring on deployed machines to detect anomalies in memory, thereby enabling the identification of potential ongoing attacks. This holistic approach combines bio-inspired principles with SDN technology to fortify distributed systems against cyber threats.

11. In [71], an additional SDN-based Moving Target Defense (MTD) technique is introduced, leveraging DNS (Domain Name System) and network address translation (NAT) to accomplish IP and medium access control (MAC) address translation. In this approach, each host is assigned a virtual IP address with a very short time to live (TTL), compelling frequent resolution requests from the defender's DNS records, which are continuously refreshed with new addresses.Upon detection of such requests, an SDN controller updates the NAT with mapping rules that translate the virtual IPs and MACs to their corresponding real ones. To enhance protection against insider threats, the hardware and operating systems of the systems are enclosed within a trusted computing base (TCB). This ensures that only authorized and secure entities have access to critical components, thereby strengthening the overall security posture of the system.

12. In [49], a novel address mutation technique for Moving Target Defense (MTD) is presented, characterized by its speed, unpredictability, and adaptability to adversarial behavior. In this approach, when a client attempts to connect to a host, it initiates a DNS query and receives a short-lived destination IP address.The system is governed by a controller, which may involve multiple controllers for enhanced scalability. These controllers manage flow rules based on temporary IPs and allocate address ranges to routers. To optimize routing table size, the address ranges typically remain within the same physical subnet. However, this optimization comes at the expense of predictability, as the addresses may change frequently due to mutation. This dynamic approach enhances the system's resilience against potential attacks by making it harder for adversaries to predict and target specific IP addresses.

13. In [69], a novel Moving Target Defense (MTD) technique is introduced, leveraging address and port mutation to achieve high levels of unpredictability. This approach utilizes source and service identity, along with time, to enhance the unpredictability rates.At the commencement of each time interval, new virtual addresses and ports are allocated to hosts from a pool of available resources. This allocation is facilitated using a secret key shared between authenticated clients and the server. The real-to-virtual address translation, along with the interception and replacement of IPs in DNS queries, is orchestrated by a port- and address-hopping gateway. This gateway dynamically modifies both addresses and ports, enhancing the system's ability to thwart potential attacks by making it challenging for adversaries to anticipate and target specific network components.

14. In [27], a novel Moving Target Defense (MTD) technique is introduced, which employs IP hopping for both genuine and decoy nodes, inspired by a strategic game between defenders and attackers. The proposed approach acknowledges the adversary's capability to differentiate between real and decoy nodes by exploiting timing disparities in responses or employing protocol fingerprinting techniques. To simulate these attack scenarios, the study employs principles from game theory, enabling the formulation of a strategic game where the attacker endeavors to identify the authentic node amidst the decoys.This model serves as the foundation for identifying the most effective IP randomization scheme. This enables defenders to optimize their MTD strategy by dynamically altering IP addresses, thereby increasing the difficulty for attackers to distinguish between real and decoy nodes.

15. In [50], the authors propose an innovative address-mutation scheme aimed at dynamically altering hosts' IP addresses based on their location and the current time. Within this scheme, each host is equipped with a collection of short-lived IPs, enabling connections with other hosts in the network. These IPs are structured in a manner where each is exclusively assigned to a specific host for communicating with a particular destination, and this allocation is accessible through DNS requests. Once the designated time period elapses, all IPs undergo mutation. The switches in the network infrastructure play a crucial role in this process, as they are responsible for substituting the sender's real IP address with a short-lived one suitable for the specific connection. The assignment of IPs and the programming of flow rules on switches are centrally managed by a controller. This approach ensures dynamic IP address management, thereby enhancing the network's security and resilience against potential threats.

16. In [34], the authors propose leveraging the extensive address space provided by IPv6 to continually rotate the IP addresses of both clients and servers. This technique involves address translation irrespective of the current connection state, leading to an increased number of packet losses within the network. However, the method does not result in the dropping or renegotiation of sessions. This approach enables two hosts to communicate over Ethernet by encapsulating each frame as a User Datagram Protocol (UDP) datagram. New headers are added to each packet, and the entire packet may potentially undergo further encryption. Both address obfuscation and optional tunnel encryption are achieved using a symmetric key, which must be pre-established and known to both parties beforehand. It's important to note that the scheme does not include key establishment as part of its process, thus necessitating out-of-band communication for key sharing.

Table 3.1: A Taxonomy of Surveyed MTD Architectures, Covering the Address and Port Mutation

What	How	When	Technology	Paper
IP	Randomness	Timer	SDN	[48]
IP	Randomness	Event	SDN	[117]
IP	Randomness	Timer,Event	SDN	[40]
IP	Randomness	Event	SDN	[72]
IP	Randomness	Timer	SDN	[96]
Port	Randomness	Event	SDN	[23]
IP	Randomness	Timer	SDN	[19]
Port	Pre-Shared Key	Event	Not defined	[70]
IP	Markov Game	Event	Not defined	[74]
IP,VM	Randomness	Timer,Event	Not defined	[4]
IP,Mac	Randomness	Timer	SDN	[71]
IP	Randomness	Timer	Not defined	[49]
Port,Address	Pre-Shared Key	Timer	Not defined	[69]
IP	Game Theory	Event	Not defined	[27]
IP	Randomness	Event	Not defined	[50]
IP	Randomness	Timer	Not defined	[34]

3.1.2 Route Mutation

Route mutation, another prevalent MTD method discussed in several articles, involves dynamically assigning random paths between hosts for each traffic flow or even per packet. This technique is particularly advantageous in Software-Defined Networks (SDNs) because it isn't constrained by the physical links between network devices. By continuously changing the routes, it becomes challenging for attackers to predict and target specific paths, enhancing the network's resilience against various types of attacks, including DDoS and crossfire attacks. The effectiveness of route mutation lies in its ability to disperse network traffic across multiple routes, thus preventing congestion and minimizing the impact of targeted attacks. This distributed approach significantly increases the volume of traffic required to overwhelm the network, making it more resilient against large-scale attacks.

1. In [47], the authors introduce MTD architectures that leverage SDN (Software-Defined Networking) and intent-based networking to bolster network security and prevent crossfire attacks. Under this methodology, dynamic network management techniques are employed to thwart potential threats effectively.In this architecture, every packet forwarded by SDN switches is routed to an SDN controller, which assesses its legitimacy. The controller then takes one of two actions to redirect the traffic to a shadow network, effectively neutralizing potential attacks. The first approach involves altering the destination address of the malicious packet, while the second method modifies the destination port within the packet. These proactive measures enable the network to effectively mitigate the impact of potential attacks and maintain its security posture.

2. In [85], the author introduces an innovative SDN-based Moving Target Defense (MTD) technique focused on topology mutation. This approach comprises four key components: Virtualized Network (VN), Network Topology Management (NTM), Network Monitoring (NM), and

Topology Mutation Control (TMC). The Virtualized Network (VN) represents both physical network components and software-defined elements. NTM encompasses specialized submanagers responsible for controlling and orchestrating the lifecycle of virtual resources within the network function virtualization infrastructure domain and virtual network function instances. NM is tasked with gathering network information from various security elements dispersed throughout the network. Finally, TMC plays a pivotal role in determining a new network topology. This change can be triggered periodically or in response to alerts from NM. The new network state may be selected from a predefined set of configurations or generated dynamically as a response to detected threats. This comprehensive framework enables the network to continuously adapt and evolve its topology, enhancing its resilience against potential cyber threats.

3. In [126], a pioneering MTD framework for spatiotemporal route mutation is introduced, employing a stochastic optimization model to bolster network security. This system is overseen by a controller tasked with distributing traffic among nodes while considering various parameters. These parameters include link capacity, predefined network and user requirements, node stability, and overall user quality of experience (QoE). At regular intervals, the controller updates nodes with new routes based on the current network conditions and user needs. This dynamic approach enables the network to adapt to changing threats and conditions, optimizing its security posture while ensuring a satisfactory user experience.

4. In [99], a novel technique is introduced that combines Software-Defined Networking (SDN) and Moving Target Defense (MTD) to counter Distributed Denial of Service (DDoS) threats in high-speed networks, particularly within Internet Service Provider (ISP) networks. This approach necessitates collaboration among multiple ISPs to establish a joint environment where each ISP monitors data flow and shares security events. The network's architecture involves border routers utilizing the Border Gateway Protocol (BGP), alongside multiple backup routers, to dynamically reshape the network's configuration in response to detected threats. Additionally, IP hopping and host-level MTD techniques are employed to deploy honeypots strategically, enticing and trapping potential attackers, thereby mitigating the impact of DDoS attacks. This collaborative and multi-layered defense strategy enhances the network's resilience against sophisticated and large-scale DDoS attacks.

5. In [45], a novel technique is presented to defend against Denial of Service (DoS) and reconnaissance attacks by dynamically reconfiguring the network topology through the manipulation of connections between nodes using Software-Defined Networking (SDN) technology. This mechanism operates in either an active mode, where the network topology is periodically reconfigured, or a reactive mode, which selects the most effective configuration in response to detected attacks. Utilizing a heuristic algorithm, the system aims to achieve a near-optimal solution for large-scale networks. By continuously adjusting the network topology based on evolving threats, this approach enhances the network's resilience and ability to withstand DoS and reconnaissance attacks.

6. In [113], an SDN-based approach is introduced to counter reconnaissance attacks by employing a shadow network to fabricate scan responses. The system comprises three primary components: a scan sensor, reflector, and shadow network. The scan sensor is tasked with detecting scan traffic and subsequently notifying the reflector, a network device capable of redirecting traffic flow. Multiple instances of these elements can exist within a system, with segments

of a protected network situated behind each one. Upon detecting scan traffic, the reflector redirects it to a shadow network, where forged scan responses are generated to mislead the attacker. This innovative strategy enhances network security by deceiving potential attackers and thwarting their reconnaissance efforts.

7. The paper [122] introduces an innovative MTD technique based on Software-Defined Networking (SDN), which relies on path hopping to obfuscate traffic within the network. This method not only distributes traffic across multiple paths but also scrambles IP addresses and ports during transit. In this approach, an SDN controller plays a pivotal role in directing packets to their destinations. It achieves this by configuring appropriate flow rules on each OpenFlow switch along the packet's path and providing them with new virtual source and destination addresses. When a switch receives a new packet and successfully matches it against its flow rules, it first alters the source and destination addresses and ports before forwarding it to the next hop. This technique ensures that intercepted packets cannot be traced back to either the sender or the intended recipient, enhancing the security and privacy of communication within the network.

8. In their work detailed in [11], the authors introduce an innovative SDN-based Moving Target Defense (MTD) technique aimed at mitigating the impact of crossfire attacks through the implementation of path mutations. This proposed approach is structured around a two-stage defense mechanism orchestrated by SDN controllers. During the initial stage, the emphasis is on obfuscating the construction of the link map to confound potential attackers, particularly during the reconnaissance phase. Subsequently, in the second stage, the system focuses on detecting and effectively mitigating the ongoing attack itself. A key aspect of this architecture lies in its ability to dynamically reorganize routes, thereby circumventing congested connections and ensuring minimal disruption to network services. By leveraging path mutations within an SDN framework, this technique enhances the network's resilience against crossfire attacks, effectively safeguarding against potential disruptions while maintaining operational continuity.

Table 3.2: A Taxonomy of Surveyed MTD Architectures, Covering Route Mutation

What	How	When	Technology	Paper
Route	Randomness	Event	SDN	[47]
Topology	Randomness	Timer	SDN	[85]
Route	State	Event, Timer	Not Defined	[126]
Path, Host, IP	Not Defined	Event	SDN	[99]
Topology	Finding Optimal Shuffle	Event, Timer	SDN	[45]
Topology	Randomness	Event	SDN	[113]
Path	Randomness	Event	SDN	[122]
Path	State	Event	SDN	[11]

3.1.3 Host Mutation

Host mutation is a powerful MTD technique that facilitates swift and dynamic swapping of virtual machines (VMs). When VMs are affected, they can be either identical and quickly replicated in

various locations, potentially possessing different addresses, or each VM can be unique. Achieving uniqueness involves customizing the software stack on every VM while ensuring they maintain the same functionality.

The process involves swapping VMs in and out of operation, allowing them to reappear at a later time. This flexibility is akin to route mutation, enabling the system to adapt rapidly to changing conditions and threats. Host mutation is particularly effective against DDoS attacks due to its ability to disperse and redistribute computing resources across different locations and configurations, making it challenging for attackers to target specific VMs or overwhelm the system.

1. In [14], an innovative Moving Target Defense (MTD) architecture is introduced to mitigate the impact of Distributed Denial of Service (DDoS) attacks by introducing a layer of proxy servers between users and application servers. Each proxy server functions as a Network Address Translator (NAT), maintaining mappings between internal and external network addresses.The system is centrally managed by a controller, which employs a DNS mechanism to ensure that only a subset of these proxy servers are active and accepting connections at any given time. This is achieved by dynamically modifying the address returned by DNS queries. Users are assigned to a specific proxy server through short-lived DNS entries.The controller also maintains records of past and current user-to-proxy assignments and possesses the capability to isolate users identified as malicious within a few shuffles. This dynamic approach enhances the network's resilience against DDoS attacks by constantly altering the network topology and distributing traffic across multiple proxy servers.

2. In [31], a novel Moving Target Defense (MTD) technique is introduced, leveraging Software-Defined Networking (SDN)-enabled cloud infrastructure for VM migration to combat Distributed Denial of Service (DDoS) attacks. The proposed architecture supports both proactive and reactive defense strategies, aiming to minimize migration frequency and select optimal new locations with the lowest attack probability to reduce defender costs. In the proactive approach, VM migration occurs preemptively to preemptively relocate resources away from potential attack vectors. Meanwhile, the reactive defense strategy kicks in once an attack is detected, prompting VM migration to more secure locations. Additionally, a honeypot node is employed in the reactive defense to deceive attackers and gather valuable data about their methods and behavior. A subset of articles that employ host mutation techniques apply them to proxy servers positioned between clients and the actual application server. This configuration facilitates the identification of attackers by periodically shuffling client-proxy assignments, particularly after each detected attack. Clients associated with the affected server are subsequently redistributed among other available servers, with a central server maintaining records of past assignments. This method effectively flags potentially malicious users, allowing for their isolation and eventual identification if they persist in their attack attempts.

3. In [18], the authors present an innovative Moving Target Defense (MTD) technique that leverages deep reinforcement learning to defend against Distributed Denial of Service (DDoS) attacks. In this approach, each user is assigned a score and is randomly allocated to one of several reverse proxies, which serve as intermediaries for requests to the application server. During an attack on any given reverse proxy, the scores of users currently assigned to it are decreased, and these users are subsequently reassigned to a different reverse proxy. To optimize the identification and isolation of malicious users, the deep Q-learning algorithm is employed to suggest the most effective user shuffle. This dynamic approach enhances the

network's resilience against DDoS attacks by continuously adjusting user assignments based on real-time conditions and attack patterns.

4. In [32], a novel SDN-based Moving Target Defense (MTD) technique is introduced, which employs VM mutation to minimize overheads in cloud management while enhancing security against cyberattacks. This approach involves selecting a new location for a virtual machine (VM) based on various factors including computation and storage capacity, network bandwidth, and the VM's reputation, which is determined by its history of cyberattacks. The migration process is initiated upon detection of a Denial of Service (DoS) attack by an Intrusion Detection System (IDS). Once migration is completed, all clients, except the attacker, are redirected to the new VM location. This proactive approach effectively mitigates the impact of attacks by relocating vulnerable VMs to more secure environments, thereby enhancing the resilience of the cloud infrastructure against cyber threats.

5. In [106], a novel Moving Target Defense (MTD) technique is introduced to mitigate the impact of Distributed Denial of Service (DDoS) attacks by employing a combination of proxies, client shuffling, and proxy replacement. This approach aims to reduce the attacker's ability to collect information about the proxies while minimizing the impact of attacks on the network. The system includes a lookup server responsible for authenticating and assigning clients to proxies. To mitigate the risk of insider threats harvesting IPs, the client-to-proxy assignments remain consistent across sessions. When a server comes under attack, it is replaced by two new servers, and its clients are distributed between them. This process can be iterated multiple times, effectively concealing the identity of the insider attacker and ultimately isolating them from the network. This dynamic defense strategy enhances the resilience of the network against DDoS attacks by constantly altering the network topology and distributing client traffic across multiple servers.

6. In [54], a novel Moving Target Defense (MTD) technique is introduced to combat Distributed Denial of Service (DDoS) attacks by employing server relocation and client assignment shuffling. When a server detects an ongoing attack, it swiftly replicates itself across different network environments, with a portion of the original clients assigned to each new instance. To address the risk of attackers posing as legitimate clients and launching new attacks on replicated servers, a coordination server is introduced to oversee the defense system. This coordination server tracks the process of replicating servers and shuffling client assignments, ensuring the integrity and effectiveness of the defense mechanism. The replication of servers and client assignment shuffling can be iterated multiple times, if necessary until the attacker is detected and isolated. To expedite this process, a greedy algorithm is proposed, which optimizes the selection of servers and clients for replication and reassignment. This dynamic defense strategy enhances the resilience of the network against DDoS attacks by continually altering the network topology and redistributing client traffic across multiple server instances.

7. In [83], a novel Moving Target Defense (MTD) technique centered on Virtual Machine (VM) migration is introduced, utilizing a probabilistic model to enhance network security. The approach is grounded on the assumption that active VMs are susceptible to attacks, with a higher likelihood of being targeted as they remain active for longer durations. To mitigate this risk, the technique incorporates a risk assessment factor to determine the necessity of VM migration. Periodically, each active VM evaluates its risk level and decides whether to replicate onto one of its replacement instances based on this assessment. Moreover, the

technique calculates the threat level associated with each potential replacement location, enabling the selection of the safest option with the lowest risk factor. By leveraging this probabilistic model, the MTD technique dynamically adjusts VM deployment, reducing the exposure time of vulnerable VMs and enhancing the network's resilience against potential attacks.

8. In [53], a novel Moving Target Defense (MTD) technique is introduced as a countermeasure against Distributed Denial of Service (DDoS) attacks. This technique involves the deployment of a layer of proxies between the client and the application server, exclusively catering to authenticated users. The defender maintains a sizable pool of proxies, of which only a subset are active at any given time, thereby reducing operational costs. Each proxy is configured with a concealed IP address to enhance security. When a user initiates a connection, they first authenticate themselves through proof-of-work on an authentication server. Subsequently, the authentication server assigns a single proxy to the user. In the event that a proxy detects an ongoing attack, it promptly notifies the authentication server and proceeds to shut down. Following this, each user is reassigned to a newly activated proxy node situated at a different network location. This dynamic defense strategy effectively mitigates the impact of DDoS attacks by continually shuffling client assignments across a distributed network of proxies, thereby enhancing the resilience of the system against potential threats.

9. In their publication [46], the authors propose an innovative approach utilizing VM-based servers to bolster cybersecurity defenses. This technique involves deploying a cluster of VMs, each hosting unique software configurations, with the aim of broadening the attack surface and introducing unpredictability to thwart potential threats. The VMs within this framework are subject to rotation, either following a predetermined schedule or triggered by specific events. At any given time, only a subset of these VMs remains operational. Incoming client requests are directed through a load balancer tasked with distributing the workload among the currently active VMs.To facilitate effective management and coordination, the VMs are linked to a central control station. Periodically, these VMs transmit status data to the control station, enabling informed decisions regarding state changes such as VM rotation or activation. By dynamically adjusting the availability and software configurations of VM-based servers, this technique introduces complexity and uncertainty into the network environment, thereby enhancing its resilience against potential cyber threats.

Table 3.3: A Taxonomy of Surveyed MTD Architectures, Host Mutation

What	How	When	Technology	Paper
Proxies	Randomness	Timer	Not Defined	[14]
VM	State	Event	SDN	[31]
Proxies	Deep Learning	Event	Not Defined	[18]
VM	State	Event	SDN	[32]
Proxies	Randomness	Event	Not Defined	[106]
VM	Randomness	Event	Not Defined	[54]
VM	State	Timer	Not Defined	[83]
Proxies	Randomness	Event	Not Defined	[53]
VM	Randomness	Event	Not Defined	[46]

3.1.4 Dynamic Resource Allocation via MTD

This MTD strategy starts with the premise that the defender has a finite number of countermeasure systems, such as IDS, and acknowledges that their deployment can potentially disrupt network performance. With this understanding, the focus shifts towards optimizing the placement of these defense systems within the network. The aim is to strike a balance between effective defense and minimal impact on network operations. This optimization process involves continual adjustments to the placement of countermeasures over time, ensuring that they remain effective while mitigating any negative effects on network performance. By strategically positioning defense systems, the defender seeks to bolster security without compromising the overall functionality and efficiency of the network.

1. In their study described in [93], the authors employed attack graphs to develop a general-sum Markov game aimed at optimizing security resource allocation in cloud network security. The authors leveraged both the Common Vulnerability Scoring System (CVSS) and the architects' assessments of how security system placement affects performance to devise long-term strategies against multi-stage attacks like Advanced Persistent Threats (APTs). This comprehensive approach facilitates the formulation of effective strategies to mitigate complex security threats in cloud networks.

2. In the study outlined in [26], the authors constructed a dynamic game with perfect information to model interactions between attackers and defenders. They employed a zero-sum Markov game framework based on attack graphs, assuming a scenario involving multistage attacks. This methodology aimed to determine the most effective tactic for the defender, compelling the attacker to adopt suboptimal strategies and thereby increasing the overall cost of the attack. The selection of optimal tactics relied on using Common Vulnerability Scoring System (CVSS) scores as a reward metric to pinpoint critical nodes and deploy countermeasures as needed. This approach enables defenders to strategize and allocate resources more effectively in combating sophisticated cyber threats.

3. In their work presented in [92], the authors address the challenge of deploying a limited number of Intrusion Detection Systems (IDS) within a large network. They propose a solution based on a Stackelberg game framework, which involves strategic interactions between the network administrator and potential attackers to determine the optimal IDS configuration.Their architecture incorporates a security scanner to identify vulnerabilities and assign Common Vulnerability Scoring System (CVSS) scores to them. These scores, along with the estimated value of the nodes, are utilized to compute the optimal solution for activating or deactivating network or host-based IDS. By leveraging game theory and vulnerability assessment, this approach allows for efficient allocation of IDS resources, maximizing network security within resource constraints.

4. In [25], a novel SDN-based MTD strategy is presented, employing a dynamic game model with a reward-and-punishment mechanism. In this game, the network administrator engages with various players, including clients, some of whom may have malicious intentions. When undesirable behavior is identified, such as through IDS alerts, the administrator can penalize the offending player by reducing the network resources allocated to them for the subsequent time period. This approach leverages game theory to incentivize cooperative behavior among

network participants while discouraging malicious activities. By dynamically adjusting resource allocations based on detected behaviors, the system enhances overall network security and resilience against potential threats.

5. In [107], a novel MTD technique is introduced with the objective of disrupting botnet operations by enhancing the complexity and reducing the likelihood of detection through the dynamic placement of detection systems. The paper outlines two key metrics to facilitate this approach: the minimum detection probability and the attacker's uncertainty. These metrics are utilized to dynamically adjust the placement of detection points within a resource-constrained network. The minimum detection probability metric serves as a lower bound indicating the likelihood of botnet detection based on the defender's strategy. On the other hand, the attacker's uncertainty metric quantifies the level of effort required by the attacker to uncover the placement of detectors. By dynamically adapting the placement of detection systems based on these metrics, the proposed technique aims to increase the difficulty of botnet detection while simultaneously reducing the likelihood of successful attacker discovery. This approach contributes to bolstering network security against botnet threats in resource-constrained environments.

Table 3.4: A taxonomy of surveyed MTD architectures, Dynamic Resource Allocation via MTD

What	How	When	Technology	Paper
IDS	Markov Game	Event	Not Defined	[93]
Counter Measures	Markov Game	Event	Not Defined	[26]
IDS	Game Theory	Event	Not Defined	[92]
Network Bandwidth	Game Theory	Event	SDN	[25]
IDs	Defined Strategy	Event	Not Defined	[107]

3.1.5 Architectures Utilizing Multiple MTD Methods

In this segment, we aim to examine architectures that integrate multiple MTD (Moving Target Defense) methods. This strategic approach is designed to yield a highly versatile defense against attackers. By combining various MTD techniques, these architectures can create a dynamic and adaptive defense system capable of effectively countering a wide range of threats and attack vectors. The integration of multiple MTD methods allows for a layered defense approach, where each method complements and reinforces the others, enhancing the overall security posture of the system. This comprehensive approach not only increases the complexity for potential attackers but also provides greater resilience and adaptability in responding to evolving cyber threats.

1. Reference [110] describes an SDN-based MTD (Moving Target Defense) engine that offers nine distinct strategies for countering various threats, including crossfire attacks. These strategies encompass network configuration, route, topology, and DNS service mutation, address shuffling, traffic reflection and manipulation, network diversification, and network elements migration. The management of this engine is centralized through a controller, which houses all the strategies and can operate in both proactive and reactive modes, selecting the most appropriate strategy based on the prevailing threat. The controller then communicates the decision to multiple agents tasked with implementing the chosen strategy at the node level.

2. In [121], the authors propose an MTD (Moving Target Defense) solution designed to safeguard networks against various types of attacks. This system possesses the capability to identify the type of threat and select the most suitable response from numerous potential mutations. The optimal approach is determined based on several parameters. Each attack is represented by a set of indicators, which are utilized after quantization to compute the total attack cost. Similarly, each defense is associated with a set of indicators used to calculate the total defense cost. The third parameter, defense efficiency, evaluates the effectiveness of a specific defense against a given threat. Lastly, the defender/attacker payoff parameters reflect the losses/gains incurred during an attack. Leveraging these parameters, the authors employ a genetic algorithm to train the system to find the optimal solution. This approach enables the maximization of defense effectiveness while minimizing defense costs.

3. In their work described in [97], the authors propose an SDN-based MTD (Moving Target Defense) technique that modifies the attack surface through host mutation, port obfuscation, and the integration of decoy servers. When a new connection enters the system, it initially passes through an Intrusion Detection System (IDS). If the connection is identified as malicious, the IDS can obscure the response. Conversely, if the connection is not flagged as malicious, it undergoes examination against a hierarchical structure of hosts within the network, referred to as a "tower." Depending on predefined rules, the connection may be obfuscated if deemed necessary. This approach optimizes network performance by selectively obfuscating potential threats.

4. In [109], the authors introduce an SDN-based technique aimed at thwarting attackers from exploiting DNS query lists and the time window attack to locate application servers. This system involves the randomization of domain names and the mutation of network addresses. Deployment by the defender necessitates authentication, DHCP, and DNS servers. To access a requested server, a legitimate client must first authenticate itself by submitting proof-of-work schemes to the authentication server. Upon authentication, the client receives the current domain name associated with the server, allowing it to query the DNS system for the corresponding IP address. This method serves to enhance security by obfuscating server identification, thereby impeding potential attacks.

5. The paper [101] introduces a network address randomization mechanism designed to maintain continuous service availability while enhancing uncertainty for potential attackers. This is achieved by deploying a substantial number of decoy servers alongside the authentic ones, with address translation conducted on both types of servers. The system comprises two primary components: an authentication server responsible for user verification and furnishing them with the current IP addresses of genuine application servers, and a randomization controller tasked with deploying decoys and facilitating network address translation. This approach aims to bolster security by complicating attacker reconnaissance efforts and ensuring uninterrupted service delivery.

6. In [123], the authors present an MTD (Moving Target Defense) technique leveraging SDNs to mutate both data paths and information about communicating entities. An SDN controller assumes the responsibility of randomizing the IPs contained within packets and determining their paths through the network via a weighted random path selection algorithm. Subsequently, utilizing this dynamically generated information, the controller configures the flow

tables in switches along the packets' route. This innovative approach enhances network security by continuously altering communication patterns, thereby thwarting potential attacks.

7. In [2], the authors introduce an SDN-based system designed to thwart reconnaissance attacks by offering each host within the network distinct perspectives. This system facilitates address translation, path mutation, relocation of vulnerable hosts, deployment of honeypots, and detection of malicious traffic flows. The architecture comprises an SDN controller responsible for generating flow rules, a deception server tasked with traffic manipulation, and a virtual network view generator that produces a representation of the virtual network. By implementing these functionalities, the system aims to confuse and misdirect potential attackers, thereby bolstering network security.

8. In [28], the authors introduce an MTD (Moving Target Defense) technique employing decoy nodes and IP hopping. They model the interaction between attacker and defender as an optimal stopping problem, aiming to minimize the tradeoff between security and performance during IP randomization. The timing of IP randomization is determined based on the elapsed time since the last address shuffle and the anticipated speed of the attacker in identifying false nodes. This timing is then compared against the cost for clients to reconnect to the node, quantified by the number of connections. By optimizing this process, the technique aims to enhance security without overly disrupting network performance.

Table 3.5: A Taxonomy of Surveyed MTD Architectures, Architectures Utilizing Multiple MTD Methods

What	How	When	Technology	Paper
All	Randomness	Timer, Event	SDN	[110]
Various	Genetic Algorithm	Event	Not Defined	[121]
Hosts, Ports	Randomness	Event	SDN	[97]
Domain name, IP	Randomness	Event	SDN	[109]
IP, Decoy	Randomness	Timer	Not Defined	[101]
IP, Path	Randomness	Event	SDN	[123]
IP, Path, Hosts	Randomness	Event	SDN	[2]
IP, Decoy	Randomness	Event	Not Defined	[28]

3.1.6 Other

1. In their study, [24] introduces an SDN-based Moving Target Defense (MTD) strategy employing a multiagent reinforcement learning framework to simulate interactions between attackers and defenders, aiming to maximize attacker rewards. The authors implement a Deep Q-Learning algorithm to determine the optimal action for a given state. Leveraging a neural network-based approach offers broader applicability, enabling decision-making even in previously unseen states.

2. In their research outlined in [124], the authors propose an SDN-based Moving Target Defense (MTD) method specifically designed to counter fingerprinting attacks. They employ a detection engine to monitor incoming traffic for indications of network scanning activities. Upon

detection, outgoing traffic is altered to mask genuine network fingerprints. To distinguish between malicious and legitimate communications, the study models the interactions between attackers and defenders as a signaling game.

3. The research described in [24] introduces an SDN-based Moving Target Defense (MTD) approach employing a multiagent reinforcement learning framework to simulate interactions between attackers and defenders with the aim of maximizing attacker rewards. The authors utilize a Deep Q-learning algorithm to determine the most suitable action for a given state. Employing a neural network-based solution offers increased generality and enables decision-making even in previously unencountered states.

Table 3.6: A taxonomy of surveyed MTD architectures, Others

What	How	When	Technology	Paper
Strategy	Game Theory	Event	SDN	[24]
Fingerprint	Signaling Game	Event	SDN	[124]
Software	Randomness	Timer, Event	Not Defined	[12]

3.2 When to Move?

Moving Target Defense (MTD) is a cybersecurity strategy designed to bolster the resilience of systems against cyber threats by dynamically adjusting their attributes and configurations. MTD systems can be categorized based on how the time period between successive alterations is determined, leading to two main classifications: Constant Period Switching (CPS) and Variable Period Switching (VPS).

3.2.1 Constant Period Switching

The fundamental concept behind these techniques involves shifting the MTD system at consistent intervals. In current research, this aspect is often implied as a design decision. For instance, frameworks that depict MTD as a game typically execute the defender's actions after a fixed time period, unless they incorporate time discretization into the state information. Alternatively, some approaches explicitly state the interval for moving the MTD system to maintain security, such as moving it every 60 seconds [51].

Many research works in the field of Moving Target Defense (MTD), such as those cited as [51],[75],[95],[92],[23], do not explicitly discuss the concept of a time interval for system movement but rather assume that the system moves at regular intervals. In these studies, authors typically (1) depict the problem as a single-step game and then extend the solution to multiple stages, which may not necessarily be repeated games, and (2) differentiate between the issues of when to switch and how to switch, focusing primarily on the latter. This approach often leads system administrators or defenders to adopt a constant time interval when implementing MTD solutions.

In their study, referenced as [103], the authors experiment with varying time periods from 60 to 300 seconds to assess the effectiveness of operating system (OS) rotation within a network-based system. They measure effectiveness based on three factors: the likelihood of thwarting an exploit, the impact magnitude of an exploit, and application availability. They find that a

shorter time interval of 60 seconds often suffices to counter Network Mapping (Nmap) attacks, though it may allow for accurate OS fingerprinting. However, this isn't advantageous for attackers as the fingerprinted OS could shift during the attack. On the other hand, a longer interval of 300 seconds appears less promising against automated attacks but fares well against human experts. To facilitate faster rotations, the authors implement multiple machines with different OSs, redirecting traffic to a new OS at each interval. While this minimizes the inconvenience of rapid switching, it increases maintenance costs. To mitigate this, they suggest maintaining at least two systems for small switching periods, especially when transitioning between OSs. With two systems, one sets up the environment while the other handles traffic, switching roles at each interval.

In [32], researchers leverage historical attack data to derive a cyber attack inter-arrival rate (CAIA). They then formulate an optimization problem aiming to maximize the switching time period, ensuring that the constant timing function remains below the CAIA. Similarly, in [11], authors analyze traceroute data between source-destination pairs to determine an appropriate time interval for obfuscating links or mutating routes of ICMP packets in the network. This strategy aids in mitigating crossfire attacks.

In [5], the MTD states involve a bipartite mapping between hosts and virtual IPs (vIPs), with each edge having a constant mutation time. Certain hosts are designated as High-Frequency Mutation (HFM) while others are Low-Frequency Mutation (LFM). To maintain host availability, HFM hosts are mapped to two vIPs for a short duration compared to the switching time period. In [8], hosts maintain local timers triggering vIP changes, communicated to other nodes via ad hoc messaging for routing facilitation.

3.2.2 Variable Period Switching (VPS)

In VPS, the timing period for switching between configurations may be determined based on various factors such as system workload, network traffic patterns, or even opponent behavior. This variability adds an element of unpredictability to the defense mechanisms, making it more challenging for attackers to anticipate and exploit vulnerabilities. Variable Period Switching (VPS) is a switching strategy used in Moving Target Defense (MTD) systems. Unlike Constant Period Switching (CPS), where the switching time remains constant, VPS introduces variability in the timing function. The primary objective of VPS is to adjust the switching time dynamically based on the current condition of the system. This variability aims to enhance the system's adaptability and resilience by making it more difficult for attackers to predict and exploit vulnerabilities.

A VPS paired with an MTD mechanism resembles a two-layer MTD approach. The initial layer manages a meta MTD to alter the time-surface, while the subsequent layer handles the actual cyber-surface switching. We'll now categorize MTD approaches into the following sub-classes.

1. **On-Event Switching:** This subcategory encompasses works with variable timing functions. It revolves around responding to specific events, like attack detection or server unavailability, by triggering a switch within a specified time period, usually immediately.

   In the FlipIt games case study[105], the system's state is represented by a boolean variable indicating control by the defender or attacker, which is unknown to both parties. The defender's strategy assumes the attacker's control, influencing the timing function. Similarly, in another study, authors[124] adjust beliefs about sender types based on detecting suspicious packets, affecting the timing function. In a separate work, authors[80] use a belief variable to indicate vulnerability exploitation, triggering a switch action when the belief exceeds a set threshold.

2. **Strategic:** To grasp these techniques, we devised a timing function generating integer values from 0 to t_m These integers correspond to discrete time intervals, where t signifies the time of a move. In their study referenced in [35], the authors adopt a simplified approach to defining the timing function, prioritizing scalability over-optimization. This function evaluates the impact of known vulnerabilities in the current system configuration and suggests a switching time t is 0,1,....., infinity If no vulnerabilities surpass a predefined threshold, the system remains unchanged until a new vulnerability emerges.

3. **Hybrid Switching:** While initially seeming counter-intuitive, researchers in [27] explore MTD mechanisms incorporating both CPS and VPS. One layer adjusts the response time window to strategic adversaries through an event-based timing function, while another layer relocates decoy nodes to evade active adversaries using a constant timing function. This study serves as a foundation for comparing the effectiveness of CPS versus VPS for specific surfaces and determining optimal combinations of both.

3.3 How to Move?

In the context of Moving Target Defense (MTD) for network systems, a range of strategies are utilized to bolster security against potential attacks.

3.3.1 Randomness

- **Dynamic System Parameters:**By integrating randomness, system parameters such as network configurations, IP addresses, port numbers, and encryption keys undergo continual, non-deterministic alterations. This dynamic shift injects an aura of uncertainty into the system, posing challenges for attackers aiming to pinpoint and exploit specific vulnerabilities.

- **Increased Unpredictability:** Randomness introduces an element of unpredictability into the network environment. Attackers typically rely on identifying and exploiting known system configurations or patterns to execute successful attacks. Through the constant, random adjustment of these parameters, MTD aims to disrupt attackers' ability to gather reliable intelligence and effectively plan their assaults.

- **Complexity for Attackers:** The integration of randomness amplifies the complexity of the attack surface for adversaries. With parameters continually fluctuating, attackers encounter hurdles in consistently identifying and exploiting vulnerabilities. This heightened complexity serves as a deterrent, compelling attackers to invest more time and resources into comprehending the evolving system, thereby diminishing the likelihood of successful attacks.

- **Defense Against Known and Unknown Threats:** Randomness within MTD not only safeguards against known attack vectors but also offers protection against zero-day exploits and emerging threats. By infusing randomness into system elements, organizations can proactively fortify their networks against a broad spectrum of potential attacks, irrespective of attackers' familiarity with the system.

- **Continuous Adaptation:** The dynamic essence of randomness ensures ongoing adaptation to evolving threats. By routinely altering system parameters through random algorithms or

processes, MTD maintains a robust level of security resilience, making it arduous for attackers to exploit consistent patterns or configurations over time.

3.3.2 Pre-Shared Key Derivation

Several research papers, including those referenced as [70] and [69], suggest the derivation of fresh values from pre-shared keys exchanged between hosts and servers.

- **Pre-Shared Keys:** These are cryptographic keys shared between authorized entities before any communication occurs. They are used for secure communication channels and authenticating the identities of communicating parties.
- **Deriving New Values:** Instead of relying solely on static keys, research suggests utilizing pre-shared keys to dynamically generate unique values. Through cryptographic processes, the system can derive new values, influencing various aspects of the network environment.
- **Enhancing Security:** By deriving new values from pre-shared keys, the system introduces an additional layer of security through cryptographic mechanisms. These values can be used to modify network settings, encryption keys, access controls, or other security parameters, bolstering the system's overall security.
- **Dynamic Configuration:** Pre-shared key derivation enables dynamic adjustments of system parameters based on cryptographic calculations. This flexibility ensures that the network environment remains dynamic, making it challenging for attackers to exploit static configurations or predict system behavior.
- **Authentication and Authorization:** Pre-shared keys are crucial for authenticating and authorizing communication. Deriving new values from these keys ensures that only authorized entities with the correct keys can access specific resources or perform certain actions within the network.
- **Adaptive Security:** Generating new values from pre-shared keys allows the system to adapt to evolving security requirements and threats. Continuously deriving and updating values based on these keys enables organizations to strengthen defense mechanisms proactively and respond effectively to cyber threats.

3.3.3 Current State-Based Calculations

Additional articles, such as [31], [126], [32], [11], and [83], propose the computation of fresh values based on the present state of the system. This method may entail evaluating factors like network conditions (e.g., link congestion, available storage space) or historical occurrences (e.g., past attacks on a node) to identify optimal strategies for enhancing security.

- **Dynamic Decision-Making:** By scrutinizing the real-time state of the system, organizations can dynamically compute new values or adjust security parameters to adapt to changing circumstances. This strategy facilitates adaptive security measures that can swiftly address emerging threats or vulnerabilities based on the system's immediate status.

- **Network Parameters:** Elements like link congestion, available storage space, bandwidth usage, and traffic patterns are factored in when determining new values or making security decisions. Through monitoring these network parameters, organizations can optimize system performance and bolster security by fine-tuning configurations accordingly.
- **Historical Data Analysis:** Previous cyber attacks on specific nodes or known system vulnerabilities exploited in the past are taken into account during the decision-making process. By drawing insights from historical data, organizations can proactively mitigate risks, fortify defenses, and prevent similar security breaches from recurring in the future.
- **Risk Assessment:** Real-time evaluations of the current system state aid in conducting on-the-spot risk assessments, analyzing vulnerabilities, assessing the threat landscape, and identifying potential attack vectors. By integrating the system's present condition into the equation, organizations can prioritize security measures and allocate resources efficiently to address the most pressing security challenges.
- **Continuous Monitoring and Adjustment:** The dynamic nature of assessments based on the current system state ensures that security measures remain adaptive and responsive to evolving conditions. By vigilantly monitoring network parameters and historical data, organizations can implement timely security updates, adjust configurations, or enact response actions to uphold a resilient security posture.

3.3.4 Game Theory Modeling

An emerging trend in Moving Target Defense (MTD) strategies is the utilization of game theory to simulate interactions between attackers and defenders. Several research papers, including those referenced as [93], [24], [26], [92], [124], [25], [74], and [27], delve into how concepts from game theory can inform strategic decision-making in response to evolving threats. By scrutinizing the dynamics of the attacker-defender relationship, organizations can deploy optimal security measures to mitigate risks effectively.

- **Strategic Decision-Making:** Explored in research papers such as [93], [24], [26], [92], [124], [25], [74], and [27], game theory informs strategic decision-making in cybersecurity. By comprehending the motivations, actions, and tactics of both attackers and defenders, organizations can devise effective security measures to counter evolving threats.
- **Attacker-Defender Dynamics:** Through game theory modeling, a deeper understanding of the interactions between attackers and defenders is attained. Analysis of the strategies, risks, and potential gains of each side enables organizations to anticipate adversary behavior and proactively implement countermeasures to mitigate risks.
- **Optimal Security Measures:** Game theory modeling aids in identifying security measures that disrupt attacker strategies, elevate the cost of attacks and fortify the resilience of the system. By strategically deploying defenses informed by game-theoretic analysis, organizations bolster their security posture and minimize vulnerabilities.
- **Risk Mitigation:** Game theory modeling assists in evaluating and mitigating risks by exploring various scenarios and potential outcomes of attacker-defender interactions. Through simulation of different strategies and responses, organizations pinpoint vulnerabilities, prioritize security investments, and construct robust defense mechanisms against cyber threats.

- **Adaptive Defense Strategies:** Leveraging game theory modeling enables organizations to adapt their defense strategies in real time to the evolving threat landscape. Continuous analysis of game dynamics and adjustment of security measures empower organizations to stay ahead of attackers and adeptly defend against sophisticated cyber threats.

3.4 SDN-Based Shuffling MTD

1. **MAC Address Shuffling:**The MAC address shuffling technique depicted in Figure 3.1 outlines a sequence of steps for transmitting data from Host A to Host B, aiming to improve security by obfuscating MAC addresses in a Software-Defined Network (SDN). Here is a detailed explanation of the address shuffling process:

    - **Initiating Communication:**The process starts with Host A planning to send data to Host B within the SDN environment. Before the data can be transmitted, Host A must determine the MAC address of Host B to correctly address the data packets.
    - **Address Resolution Protocol (ARP) Request:**Host A begins communication by broadcasting an ARP request packet to the network. This ARP request aims to obtain the MAC address of Host B, which is crucial for directing data packets to the correct destination.
    - **Switch Interaction:**The switch SW1, connected to Host A, receives the ARP request packet from Host A.As SW1 lacks the necessary information to handle the ARP packet internally, it forwards a packet-in message to the SDN controller for further guidance.
    - **SDN Controller Response:**Upon receipt of the packet-in message from SW1, the SDN controller handles the ARP request from Host A. Rather than broadcasting ARP packets throughout the network, the SDN controller promptly replies to the ARP request packet, enhancing communication efficiency.
    - **MAC Address Obfuscation:**If the SDN controller does not possess the real MAC address (rMAC) of Host B (hd), it undertakes additional procedures to securely obtain this information. Subsequently, the SDN controller selects a virtual MAC address (vMAC) for Host B and incorporates this vMAC in the ARP reply packet.
    - **Flow Table Update:**As a vital component of the address shuffling process, the SDN controller modifies the flow tables of SDN switches along the communication path from Host A to Host B. This adjustment ensures that network switches are aware of the updated virtual MAC address (vMAC) assigned to Host B, thereby enabling smooth data transmission.

    By adhering to this address shuffling procedure, the MAC addresses of both Host A and Host B undergo dynamic obfuscation, thereby strengthening network security and deterring potential attackers from monitoring particular machines or intercepting communication within the SDN environment. This approach improves privacy and confidentiality by implementing dynamic changes to MAC addresses during data transmission.

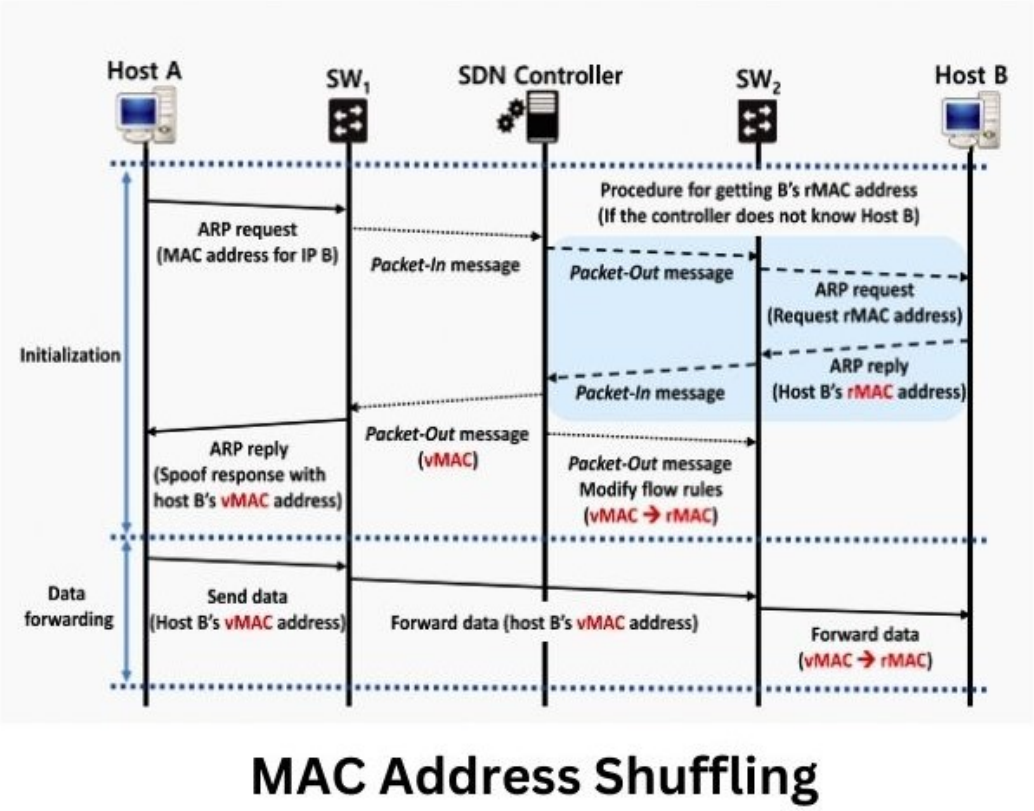


Figure 3.1: MAC Address Shuffling Sequence to Send Data from Host A to Host B.

In a Software-Defined Network (SDN), additional procedures for MAC obfuscation aim to heighten security and privacy by dynamically altering and concealing MAC addresses. Here's an explanation of these procedures:

- **ARP Reply Packet Injection:**In an SDN, the controller receives packet-in messages from switches upon new packet arrivals, making forwarding decisions. If the controller doesn't know the destination host's location, it forwards ARP packets to find it. After updating the virtual MAC (vMAC) on the controller, the actual ARP reply packet is discarded. Instead, the controller blocks the ARP packet broadcast and injects a modified ARP reply packet containing the destination host's vMAC. This modified packet is then directly sent to the requesting host, enabling communication using the vMAC.
- **Spoof Response for Non-Existing Hosts:**The controller doesn't only respond to ARP requests for active hosts but also for temporarily non-existing hosts in the network.

This introduces uncertainty for attackers, potentially causing them to spend more time checking for active hosts. Establishing a probability model based on network size and responding to ARP requests accordingly is crucial for maintaining effective deception against attackers.

- **Untraceable Virtual MAC Address Generation:**The controller generates a set of vMACs using vendor Organizational Unique Identifiers (OUI) information. The first 24 bits of the vMAC provide fake vendor information, while the last 24 bits are randomly generated. Generated vMACs should not overlap with existing ones or those recently used and stored in ARP caches of devices.
- **Assignment of Virtual MAC Address:**When SDN switches receive an ARP packet, the controller checks its ARP mapping table. If the requested IP address is found, the controller returns the associated vMAC. If not, the controller selects a vMAC from the pre-generated set based on the hash value of the requested IP address.

2. **IP Address Shuffling:**The IP address shuffling method, illustrated in Figure 3.2, delineates the series of steps for transmitting data from Host A to Host B, with the aim of enhancing security in a Software-Defined Network (SDN) by concealing the IP addresses. Here's an explanation of the address shuffling sequence derived from the provided details:

   - **Initiating Communication:**Host A commences the process of transmitting data to communicate with Host B within the SDN environment. Prior to dispatching data packets, Host A must resolve the IP address of Host B to ensure precise routing of the packets.
   - **DNS Request for IP Address:**Host A begins the communication by attempting to resolve the IP address of Host B through a Domain Name System (DNS) request packet.If the IP address of Host B is not present in the DNS cache of Host A, a DNS request is sent to the DNS server to obtain the IP address.
   - **SDN Controller and DNS Server Interaction:**The IP address shuffling process involves collaboration between the SDN controller and the DNS server. The SDN controller is responsible for assigning virtual IP addresses (vIPs) to hosts, while the DNS server maintains the real IP addresses (rIPs) of hosts.
   - **Virtual IP Assignment:**Upon receiving the DNS request from Host A, the DNS server promptly replies with the real IP address of Host B along with the corresponding domain name. Meanwhile, the SDN controller modifies the real IP address (rIP) of Host B to a virtual IP address (vIP) within the DNS reply packet, employing a predetermined sequence number.
   - **Address Conversion and Data Transmission:**The SDN controller ensures the utilization of the virtual IP (vIP) of Host B in every data frame transmitted from Host A to Host B. This initial communication setup involves converting the IP addresses to their virtual counterparts, thereby facilitating secure and obfuscated data transmission.
   - **Performance Considerations:**The IP address shuffling process incurs an additional delay of 173 ms for the first frame due to vIP assignment and conversion. Subsequent frames experience delays primarily related to data forwarding and vIP modification, ensuring smooth and secure communication.

This address shuffling sequence periodically shuffles and converts the IP addresses of Host A and Host B into virtual addresses, thereby boosting network security and privacy within the SDN environment. By introducing dynamic changes to IP addresses during data transmission, this technique complicates potential attackers' efforts to track hosts or exploit network vulnerabilities.

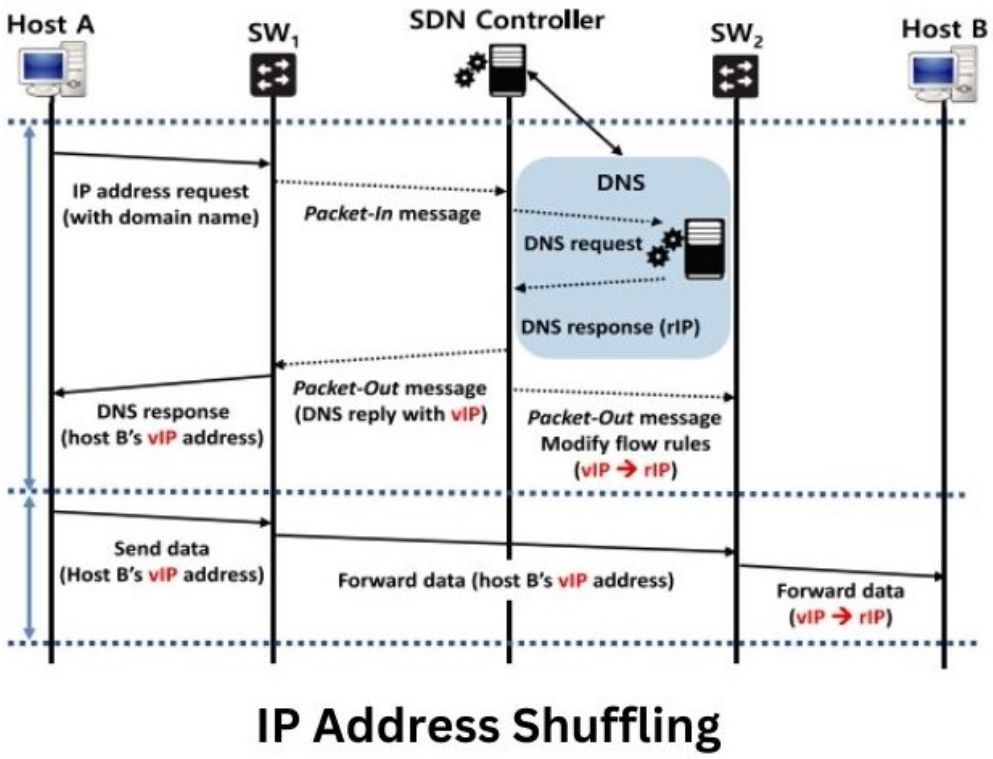


Figure 3.2: IP Address Shuffling Sequence to Send Data From Host A to Host B.

Chapter 4

Attack Graphs

4.1 Attack Graph

A cyber attack graph is a visual representation illustrating all potential attack paths against a cyber network, depicting a scenario where an attacker has successfully breached the system. There are two widely recognized formats of attack graphs. The first format utilizes a directed graph structure where nodes represent various network states, while edges depict exploits that transition one state to a more compromised state, ultimately demonstrating the progression of a successful attack. The second format employs a directed graph where nodes denote either pre or post-conditions associated with an exploit, and edges illustrate the outcomes resulting from having a pre-condition that allows for an exploit's post-condition.

The definition of attack graph as given in [9] is, An attack graph G is the directed graph $G = (E \cup C, R_r \cup R_i)$ where $E \cup C$ is the vertex set and $R_r \cup R_i$ is the edge set, given a collection of exploits E, a set of security conditions C, a require relation $R_r \subseteq C * E$, and an imply relation $R_i \subseteq E*C$. An exploit is denoted as $v(h_s, h_d)$ which indicates that a vulnerability v which originated from a source host h_s on the destination host h_d. For local hosts, any exploits are denoted by $v(h)$.

A security condition is a predicate of the form $c(h_s, h_d)$ that denotes the fulfillment of a security-related condition c involving the source host h_s and the destination host h_d (where a condition only concerns one host, it is represented as $c(h)$). A vulnerability on a specific host or connection between two hosts are two examples of security situations.

An attack graph (AG) [118] shows the relationships between vulnerabilities exploitable by attackers and the privileges they can acquire. Different AGs can be created based on node and edge representations. Typically, nodes represent states like hosts, privileges, or vulnerabilities, while directed edges signify transitions between states. Exploiting a vulnerability in AGs often leads to privilege escalation on hosts, like obtaining root access.

In their work, [91] proposed an attack graph (AG) framework that incorporates nodes characterized by five attributes: user level (e.g., none, guest, privileged, root, or admin), machine type (e.g., single host, subnet, or multiple subnets), vulnerabilities, capabilities (e.g., read, write, install a virus), and state(s). Meanwhile, each edge within this framework encompasses an action and a condition. Despite its lack of scalability, this model facilitates dynamic network modeling. Reference [81] introduced an exploit dependency graph, where nodes denote privileged nodes, pre-condition nodes (representing vulnerabilities), or post-condition nodes (representing exploits), with

edges symbolizing state transitions. Additionally, Reference [67] proposed a Bayesian AG approach, which outlines potential attack paths utilizing Bayesian networks. In this model, each node signifies a state, with multiple nodes representing a single state but with varying user privileges. Conditional probability tables assigned to edges infer exploitability. Nevertheless, conventional AGs encounter challenges related to scalability and dynamic reconfigurations. The task of identifying all attack paths in large-scale networks is exponentially complex, often necessitating the use of heuristics for mitigation. Furthermore, changes in network topology or configurations inherently impact AGs, a challenge that has not been sufficiently addressed in existing literature.

4.2 Categories of Attack Graphs

4.2.1 Based on Purpose

1. **Discovery Attack Graphs:** Discovery attack graphs concentrate on revealing vulnerabilities and potential attack routes within a network or system.

2. **Analysis Attack Graphs:** Analysis attack graphs serve the purpose of evaluating the consequences of different attacks and pinpointing potential strategies for mitigation.

4.2.2 Based on Representation

1. **Directed Attack Graphs:** Directed graphs within attack graphs illustrate the sequence of actions accessible to an attacker, typically shown as directed edges connecting nodes representing states or configurations.

2. **Undirected Attack Graphs:** Undirected graphs in the context of attack graphs depict relationships between entities without a designated direction. They are frequently utilized to visualize the connectivity of network elements or dependencies between vulnerabilities.

4.2.3 Based on Scope

1. **Local Attack Graphs:** Local attack graphs concentrate on a specific network segment, system, or application.

2. **Global Attack Graphs:** Global attack graphs encompass the entire network or system, accounting for interconnections between different components.

4.2.4 Based on Techniques

1. **Probabilistic Attack Graphs:** Probabilistic attack graphs integrate probabilities of successful exploitation or detection, commonly employed for risk assessment and prioritization purposes.

2. **Deterministic Attack Graphs:** Deterministic attack graphs are based on deterministic rules or conditions, typically utilized for comprehensive analysis of attack paths.

4.2.5 Based on Attributes

1. **Static Attack Graphs:** Static attack graphs document the system's state and vulnerabilities at a specific moment in time.

2. **Dynamic Attack Graphs:** Dynamic attack graphs account for the system's state evolution over time, integrating alterations in configurations, patches, or network topology.

4.2.6 Based on Source of Data

1. **Manual Attack Graphs:** Manual attack graphs are crafted by security experts relying on their understanding of the system and potential attack vectors.

2. **Automated Attack Graphs:** Automated attack graphs are produced through automated tools or algorithms, often utilizing vulnerability scanners, network mapping tools, and other security software.

4.2.7 Based on Time Horizon

1. **Immediate Attack Graphs:** Immediate attack graphs concentrate solely on immediate threats and vulnerabilities, excluding considerations of long-term strategic attacks.

2. **Strategic Attack Graphs:** Strategic attack graphs, on the other hand, account for long-term, multi-step attack strategies aimed at achieving specific objectives over time.

4.3 Security Vulnerabilities

Security vulnerabilities are weaknesses within systems, networks, software, or applications that can be exploited by attackers, posing risks to data and assets. They arise from design flaws, implementation errors, or inadequate security measures. These vulnerabilities exist across different technology layers and can be exploited by malicious actors to compromise systems or data. The U.S. government owns the National Vulnerability Database (NVD) [57], which serves as a repository for vulnerability data. The data is represented using the security content automation protocol. Within the NVD, one can find references to security checklists, software flaws, misconfigurations, product names, and impact metrics. It analyzes entries from the Common Vulnerabilities and Exposures (CVE) [57] by gathering information from multiple public domain sources.

The Common Vulnerability Scoring System (CVSS) [57] is the leading metric for evaluating the severity of security vulnerabilities in computer systems, using information from CVE. CVSS scores indicate the severity of individual vulnerabilities on specific hosts, not the entire system. It consists of three metric groups: base, temporal, and environmental. The score is primarily determined by the base metric, ranging from 0 to 10, with 10 being the most severe, and can be further refined by considering temporal and environmental factors. To measure the severity of vulnerabilities, an exploitability metric has been developed to assess the likelihood of a vulnerability being exploited by an attacker. This metric evaluates exploitability through two methods:

- Common Vulnerability Scoring System(CVSS)-Based Exploitability Estimation[57]
- Time-to-Compromise (TTC)-Based Exploitability Estimation[65]

4.4 Network Topology

Figure 4.1 depicts a network topology where a workstation (WS) manages user requests and a database server (DB) stores sensitive data like personal credentials and business plans. Additionally, the network includes security measures such as firewalls and an Intrusion Detection System (IDS). The firewall is utilized to enforce traffic-blocking rules, while the IDS monitors network traffic for suspicious activities.

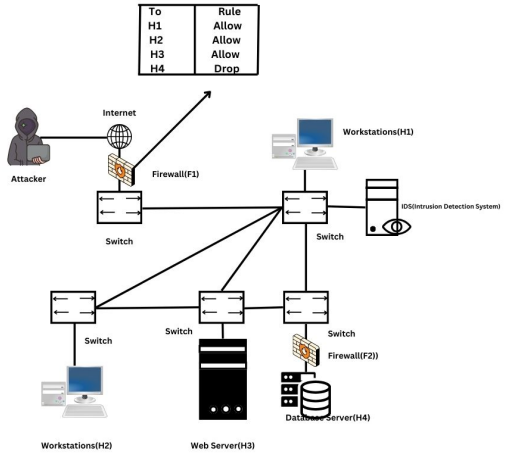


Figure 4.1: An Example of Network Topology

As shown in Fig. 4.1 the active hosts are categorized into two types based on their accessibility and connectivity within the network:

1. H_{ex} **Type Hosts:** H_{ex} type hosts are those that can be accessed by outside entities, such as remote attackers. The identified hosts are prone to exploitation and can be directly targeted by remote attackers due to existing vulnerabilities. An unauthorized attacker can exploit weaknesses on a host of type H_{ex} to compromise nearby hosts without needing access rights to their configurations or files. In the specific scenario outlined in Figure 4.1, hosts H1, H2, and H3 are categorized as H_{ex} type hosts. The firewall rule (F1) specifically allows the attacker to directly access only these hosts.

2. H_{in} **Type Host:** Hosts categorized as H_{in} type are only accessible by internal entities within the network. They cannot be directly reached by remote attackers and can only be accessed through hosts with direct communication pathways to them. In order to access a H_{in} type host, an attacker must first compromise a host within the network that can establish a connection to the H_{in} host. In the network layout shown in Figure 4.1, host H4 is classified

as a H_{in} type host. The existing firewall rule prevents direct access by remote attackers to H4, thus requiring the compromise of another host that can communicate with H4.

Distinguishing between H_{ex} and H_{in} hosts holds significance in network segmentation, as it influences accessibility and connectivity. This segmentation plays a vital role in determining the paths attackers may exploit to traverse the network. Understanding the nuances between these types of hosts is crucial for assessing network security and crafting defensive tactics. It enables the identification of potential attack vectors and the deployment of appropriate security measures to protect critical assets.In this situation, authorized users can access any host in the network using either user or root privilege accounts. However, for attackers to exploit a host's vulnerabilities, they must first acquire the necessary privilege level of the target host. It's assumed that attackers possess the capability to exploit vulnerabilities found on a target host.

Table 4.1: Vulnerabilities of an Example Network

Host	Vulnerability	CVE_ID	Cause of Vulnerability	Access	CVSS
H1	V1	CVE-2002-1644	Remote shell	Remote	6.5
	V2	CVE-2007-5616	Privilege Escalation	Remote	7.2
	V3	CVE-2006-2421	Buffer Overflow	Remote	6.5
	V4	CVE-2007-5616	Privilege Escalation	Local	7.2
H2	V5	CVE-2018-9843	Remote code execution	Remote	9.8
	V6	CVE-2000-0659	Buffer Overflow	Remote	5.0
	V7	CVE-2016-3129	Remote shell	Remote	8.5
	V8	CVE-2018-18387	Privilege Escalation	Local	8.8
H3	V9	CVE-2015-4108	Remote code execution	Remote	6.8
	V10	CVE-2018-2384	SQL Injection	Remote	7.5
	V11	CVE-2017-6516	Privilege Escalation	Local	7.2
H4	V12	CVE-2007-6304	Denial of Service	Remote	5.0
	V13	CVE-2008-3234	Remote Shell	Remote	6.5
	V14	CVE-2001-1180	Privilege Escalation	Local	7.2

Within the network, hosts are presumed to contain inherent vulnerabilities, as outlined in Table 4.1. These vulnerabilities are identified by CVE numbers and assigned CVSS severity scores. The attacker exploits these specific vulnerabilities across different hosts to advance through the attack path.

- In the initial phase of the attack, the attacker exploits the vulnerability CVE-2002-1644 on H1, which is categorized as an H_{ex} host. Since H1 is remotely accessible over the Internet, it exposes its vulnerabilities to anonymous users, allowing unauthorized write operations on its home directory.

- Host H2, categorized as an H_{ex} host, is accessible remotely, making it a prime target for attackers seeking to exploit vulnerabilities. An exploitable command shell vulnerability, identified on H2, allows unauthorized execution of commands, potentially granting control of the system. The attacker exploits the CVE-2018-9843 vulnerability, enabling remote code execution on H2.

- Host H3, like H2, is categorized as an H_{ex} host, meaning it's accessible remotely, making it a potential target for attackers looking to exploit network vulnerabilities. An exploitable vulnerability is found in H3's command shell, allowing unauthorized commands and potential system compromise. By exploiting CVE-2015-4108, the attacker gains remote code execution access to H3's command shell, enabling them to execute arbitrary code, manipulate the system, potentially escalate privileges, and further their attack within the network.
- Finally, the attacker gains root privilege on the DB server H4 by exploiting system vulnerability CVE-2001-1180. This sequence of exploitation allows the attacker to progressively escalate privileges and navigate through the network, ultimately gaining control over the DB server.

4.5 Three-Tier Attack Graph (TAG)

In the novel graphical model introduced alongside the Three-Tier Attack Graph (TAG) approach [118], a distinct separation is made between network topology information and vulnerability data within the Attack Graph. This division segregates the physical layout of the network, including connectivity and routing, from the identification of security weaknesses in network hosts.

- **Separation of Network Topology and Vulnerability Information:** In the updated graphical model, network topology information and vulnerability data are handled distinctly within the Attack Graph. This separation delineates between the physical structure of the network, such as its connectivity and routing, and the specific security vulnerabilities associated with network hosts. By categorizing these elements separately, the model simplifies the process of constructing the Attack Graph and enables a more targeted examination of vulnerabilities and potential attack routes. Essentially, this division streamlines the analysis by providing a clearer view of both the network layout and its security weaknesses, thereby enhancing the understanding of interconnected systems.
- **Directed Acyclic Graph (DAG) Conversion:** Attack Graphs often contain loops because network hosts are interconnected, posing challenges in pinpointing attack paths. However, the TAG model addresses this issue. If the Attack Graph contains loops, it can be converted into a Directed Acyclic Graph (DAG) without jeopardizing the analysis's integrity. This conversion is vital for simplifying attack path representation, making the analysis more manageable. Methods such as cycle handling techniques [111] or d-separation algorithms [42] achieve this transformation. By untangling interconnected pathways, these methods enhance the graph's clarity, aiding analysts in understanding and prioritizing security risks effectively.

The Three-Tier Attack Graph (TAG) is a graphical model employed in network security to illustrate the connections among vulnerabilities, privileges, and possible attack routes within a network. Here is a detailed explanation of the Three-Tier Attack Graph model:

1. **Layer 1 - Connectivity Layer (L1):**
    - The initial tier in the TAG, termed the Connectivity Layer, is dedicated to showcasing the network's topology and connectivity details.
    - It visually represents the interconnections and associations among various network entities, including hosts, routers, switches, and other devices.

- This layer serves as a fundamental framework for grasping the intricate interplay of network components, delineating how they interact and communicate within the network infrastructure.

2. **Layer 2 - Remote Vulnerability Layer (L2):**
   - In the second layer, known as the Remote Vulnerability Layer, the focus shifts to vulnerabilities that can be exploited remotely by attackers without needing direct physical access to the target system.
   - This layer encompasses vulnerabilities present in network services, protocols, and applications that are accessible over the network, making them susceptible to exploitation from external sources.
   - Attack pathways within this layer typically revolve around exploiting weaknesses in network services or configurations that are reachable from outside the network perimeter.

3. **Layer 3 - Local Vulnerability Layer (L3):**
   - Moving to the third layer, the Local Vulnerability Layer, attention is drawn to vulnerabilities that demand local access to the target system for exploitation.
   - These vulnerabilities often entail scenarios such as privilege escalation, lateral movement within the network, or exploiting weaknesses that necessitate physical or close proximity to the target system.
   - Within this layer, attack paths may encompass compromising internal systems, elevating privileges, or exploiting vulnerabilities accessible only from within the network environment.

In brief, the Three-Tier Attack Graph (TAG) introduces a novel graphical model for analyzing network security threats. This model separates network topology and vulnerability information, providing a structured framework that simplifies the representation of attack paths. By converting Attack Graphs into Directed Acyclic Graphs (DAGs), the TAG approach streamlines the analysis process, making it more efficient and effective.

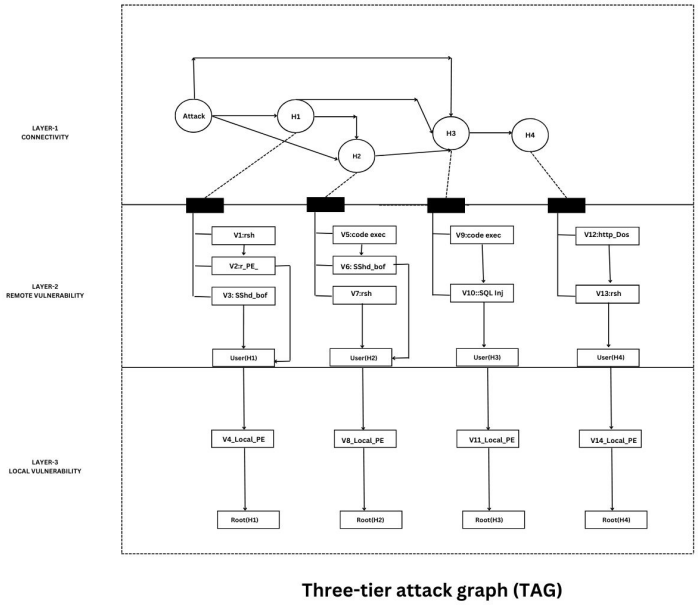


Figure 4.2: Three Tier Attack Graph

In the illustrated TAG model (Fig. 4.2), derived from the network layout presented in Fig. 4.1 and vulnerability details outlined in Table 4.1, Layer 1 (Connectivity) holds significant importance in elucidating the connections among network hosts and potential threats. This layer is specifically dedicated to mapping out the interconnections and accessibility pathways linking various network hosts with external entities, including potential attackers.

Considering that each host in the network harbors vulnerabilities, Layer 1 of the TAG model aids in comprehending how these vulnerabilities could be exploited based on the network's topology. By analyzing the connectivity insights provided in L1, security analysts can discern the potential routes attackers might take to access critical hosts within the network.

In the framework of the TAG model, Layer 2 (Remote vulnerability) is dedicated to illustrating the relationships among remote vulnerabilities exploitable by external attackers. Specifically, Layer 2 offers insights into how vulnerabilities present in one host can be utilized by an external entity to compromise other hosts within the network.

For example, let's consider a scenario where host H1 exhibits three potential remote vulnerabilities labeled as V1, V2, and V3, along with various combinations of these vulnerabilities. This scenario enables the identification of different potential attack paths, as outlined below:

- **(V1 → V2 → User (H1)):** This attack path signifies that by exploiting vulnerability V1 followed by vulnerability V2, the attacker can attain user privilege on host H1.
- **(V2 → User (H1)):** In this instance, vulnerability V2 is exploited to acquire user privilege on host H1.
- **(V3 → User (H1)):** Exploiting vulnerability V3 enables the attacker to gain user privilege on host H1.

Table 4.2: Potential Attack Path of All Host

Attack Path(Host H1)	Attack Path(Host H2)	Attack Path(Host H3)	Attack Path(Host H4)
V1->V2->User(H1)	V5->V6->User(H2)	V9->V10->User(H3)	V12->V13->User(H4)
V2->User(H1)	V6->User(H1)	V10->User(H1)	V13->User(H4)
V3->User(H1)	V7>User(H1)		

Layer 2 of the TAG model offers valuable insights into the exploitation of remote vulnerabilities by external attackers, leading to compromises of network hosts and the acquisition of user privileges. This underscores the significance of implementing proactive security measures to mitigate such threats effectively. Table 4.3 shows the Potential attack paths of all hosts for layer 2.

Within the framework of the TAG model, Layer 3 (Local Vulnerability) is responsible for illustrating the network's local vulnerabilities that an attacker may employ to get root privileges on a host. The greatest level of system access, known as root privilege, gives the user total authority over the system and all of its resources. For attackers looking to escalate their access and launch more complex and destructive operations within the network, taking advantage of local vulnerabilities to obtain root capability is an essential first step.

To effectively mitigate the risks associated with local vulnerabilities and root privilege escalation, organizations should consider implementing a variety of defense strategies. These include:

- Enforcing robust access controls and privilege management protocols to restrict user privileges and prevent unauthorized entry to critical assets.
- Consistently applying patches and updates to systems to mitigate known vulnerabilities and minimize the network's exposure to potential threats.
- Installing intrusion detection and prevention systems for monitoring and swift response to anomalous behaviors signaling potential exploitation of local vulnerabilities.
- Performing routine security evaluations and penetration tests to uncover and address vulnerabilities proactively, thwarting malicious exploitation attempts.

To improve the scalability of Attack Graphs (AGs), the introduction of the two-layer attack graph (TL-AG) [44] has proven effective. TL-AG separates reachability information from vulnerability data, enhancing the manageability of AGs. Building on this concept, we have further refined this approach by dividing vulnerability information into two distinct layers: remote vulnerabilities and local vulnerabilities. This division allows us to more effectively address compromises involving both user privileges and root privileges within the network. By incorporating this layered structure, we can more comprehensively analyze attack paths, capturing potential scenarios that exploit vulnerabilities at different privilege levels. This approach enhances our overall security analysis and defense strategies within the network.

4.6 Advantages of Three-Tier Attack Graph (TAG) in the Moving Target Defense (MTD)

1. **Structured Representation:** The TAG model categorizes network vulnerabilities and attack pathways into three layers: connectivity, remote vulnerability, and local vulnerability.

This organized framework aids in comprehending the interconnections among network components and vulnerabilities, enabling a structured method for security analysis.

2. **Comprehensive Analysis:** Through the integration of network topology details and system vulnerability data within distinct layers, the TAG model facilitates a thorough examination of potential attack routes. This comprehensive methodology empowers security administrators to pinpoint crucial elements and prioritize defensive measures efficiently.

3. **Efficient Attack Path Prediction:** The TAG model facilitates the prediction of attack paths by considering the criticality of hosts and vulnerabilities. This predictive capability helps in proactively identifying and mitigating potential attack paths, reducing the likelihood of successful attacks

4. **Simplified Complexity:** By segregating network reachability data from vulnerability details within the TAG model, the complexity associated with generating attack graphs is diminished. This streamlining enhances the clarity regarding the connections among network components and vulnerabilities, resulting in a more streamlined and effective security analysis process.

5. **Conversion to Directed Acyclic Graph:** When the TAG model featuring loops becomes cyclic, it can be transformed into a Directed Acyclic Graph (DAG) without compromising data integrity. This transition boosts the effectiveness of scrutinizing attack routes and pinpointing vulnerabilities, thereby bolstering the efficiency of defense strategies.

6. **Enhanced Security Decision-Making:** The TAG model empowers security administrators to make well-informed decisions about defense strategies through its structured framework for vulnerability analysis and prediction of attack paths. This improved decision-making capacity aids in prioritizing defensive actions on crucial hosts, thereby fortifying the overall security stance of the network.

Chapter 5

Challenges of Moving Target Defense

5.1 What To Move

Identifying system attributes that can be modified with minimal impact on performance is crucial for maintaining system stability. While system configurations are typically kept static for this reason, understanding which attributes can be altered without disrupting performance is essential. Research on Moving Target Defense (MTD) has primarily focused on the operating system and network levels. Some modifiable attributes at these levels include memory allocation, network routing, and process scheduling. When considering a change to a system attribute, it's vital to understand its dependencies. For instance, altering a memory address utilized by an application may necessitate rewriting parts of the binary code.

5.2 When To Move

The challenge lies in deciding when to modify system attributes, typically addressed through three methods: periodic, event-driven, or a hybrid of both. Periodic methods alter attributes based on a set schedule, either fixed or random, with the aim of nullifying any gathered attacker information. However, this approach can introduce unnecessary overhead on the system, as changes occur regardless of active attacks. Additionally, if a system is compromised during a scheduled change, its effectiveness is compromised. Event-based approaches modify attributes in response to detecting suspicious activities, effectiveness being contingent upon the detection mechanisms in place. A hybrid approach combines periodic and event-driven methods to enhance overall effectiveness.

5.3 How To Move

To advance, we need to craft a plan to modify the recognized system characteristics. An effective approach should heighten uncertainty and hinder the attacker's efforts. Sharma et al. [1] have proposed shuffling, diversification, and redundancy as three methods to introduce unpredictability

into a system. Shuffling entails alternating between various instances of the identified system attributes. Specifically, at the network level, shuffling can involve changes in IP addresses and adjustments to network topology.

1. The practice of shuffling IP addresses has received considerable attention in research [71],[58], [102],[16]. Studies have explored the randomization of Virtual IP addresses within Software-Defined Networking (SDN) environments, as examined by Sharma et.al [1] and Jafarian et al. [96]. Additionally, Kampanakis et al. [16] conducted evaluations regarding the impact on attacker overhead resulting from host randomization within an SDN setup.

2. Topology shuffling can be accomplished by alternating between virtual topologies within a Software-Defined Networking (SDN) environment. Achleitner et al. [96] proposed a method leveraging SDN to delay reconnaissance attacks by restricting the amount of information attackers can gather about the physical infrastructure beneath. Similarly, Hong et al. [76] introduced the Shuffle Assignment Problem (SAP), which utilizes SDN to reconfigure network topology, seeking an optimal or near-optimal solution while integrating countermeasures against network attacks.

3. Address Space Layout Randomization (ASLR) [41] is a security technique that randomizes the memory addresses of executables and shared libraries by allocating them to different memory locations. ASLR is effective in defending against code injection attacks, as it prevents attackers from predicting the exact memory locations where malicious code can be injected into executables. Another related technique is Global Offset Table (GOT) randomization, which involves rearranging the entries in the GOT. The GOT contains mappings between the actual address and the virtual address of shared library code and data [58].

4. The Trusted Dynamic Logical Heterogeneity System (TALENT), introduced by Okhravi et al. [48], presents a framework enabling applications to transition between hardware platforms and operating system virtualization while maintaining the application's state. N-version programming [20] has been employed to generate diverse variants of functionally similar programs. These variants can be rotated, ensuring that in the event of an attack, only the affected variant is impacted, enhancing system resilience.

Huang et al. [30] introduced a system that establishes a collection of virtual servers. This system employs a periodic or event-driven method to replace an online server with one from the offline server set. Yuan et al. [120] proposed a system utilizing agreement-based redundancy to produce duplicates of software components during runtime. Frequently, techniques centered on diversity and redundancy are merged with shuffling to enhance effectiveness.

Implementing Moving Target Defense (MTD) enhances system dynamism and shields it from potential threats, striking a balance between security and performance. It's important to understand that MTD doesn't offer absolute protection but rather increases the difficulty of exploitation. Traditional defense mechanisms and MTD complement each other, bolstering overall security. While much research has been conducted and some implementations achieved, challenges remain, and enhancements are needed. Developing lightweight MTD solutions for IoT is crucial, as securing the IoT ecosystem would fortify mainstream infrastructure security.

Chapter 6

Experiments

The experiment detailed in this chapter focuses on configuring an SDN (Software-Defined Networking) environment using Open vSwitch (OVS) and the Ryu controller to establish network communication, routing, and security features. The experiment aims to demonstrate the implementation and verification of a network topology utilizing the `rest_router(L2)` applications within the Ryu SDN framework.

The proposed work *"SDN Network Configuration and Routing Experiment using Open vSwitch and Ryu Controller"* involving the configuration of Open vSwitch, setting up routing through the rest_router application, and establishing network communication within an SDN framework using the Ryu controller, the following applications play important roles:

- **Open vSwitch (OVS):** Open vSwitch is a popular open-source virtual switch that is used in SDN environments to connect virtual machines and physical networks. It provides switching, bridging, and routing functionalities, making it a key component for network virtualization and SDN deployments.
- **Ryu Controller:** Ryu is an open-source SDN controller platform that provides a flexible and programmable framework for building SDN applications. In this experiment, the rest_router application within Ryu is utilized to manage routing functionalities and control network traffic flow.
- **CURL:** CURL is a command-line tool and library for transferring data with URLs. In the experiment, CURL commands are used to configure ports on the router, specify gateway interfaces, and add static routes, enabling communication between different network components.
- **DHCP Server:** A DHCP (Dynamic Host Configuration Protocol) server is used in the experiment to automatically assign IP addresses to hosts within the network. By leasing out new IP addresses after a specified time, the DHCP server ensures efficient IP address management and network connectivity.
- **Rest_ router(L2) Application:** The Rest_router(L2) application is an advanced Layer 2 network management tool aimed at improving security and simplifying network configuration. It combines features like dynamic IP address allocation, DHCP server management, and port hopping to enhance protection against network threats while making network administration more efficient.

6.1 SDN Network Configuration and Routing Experiment using Open vSwitch and Ryu Controller

The steps we discussed in our experiments are described below:

1. **Connecting Ryu with Open vSwitch (OVS):** The primary objective of this step is to establish a communication link between the Ryu controller and the Open vSwitch to facilitate network control and traffic handling in the SDN setup. Verify that both the Ryu controller and Open vSwitch are correctly configured and operational within the SDN environment. Use appropriate commands or configuration settings to connect the Ryu controller with the Open vSwitch instance. Ensure that the connection between Ryu and OVS is successful by checking for any error messages or warnings. Conduct basic tests to confirm that the Ryu controller can communicate with OVS and manage network traffic effectively.

2. **Start the Ryu Application:** We initiate the Ryu application using the following command:

   ryu-manager ryu.app.rest_router

   This command starts the rest_router application within Ryu, which is responsible for managing routing functionalities. Upon successful execution, the following messages indicate that the router configuration is set:

   [RT][INFO] switch_id=0000000000000003: Set SW config for TTL error packet in.
   [RT][INFO] switch_id=0000000000000003: Set ARP handling (packet in) flow [cookie=0x0]
   [RT][INFO] switch_id=0000000000000003: Set L2 switching (normal) flow [cookie=0x0]
   [RT][INFO] switch_id=0000000000000003: Set default route (drop) flow [cookie=0x0]
   [RT][INFO] switch_id=0000000000000003: Start cyclic routing table update.
   [RT][INFO] switch_id=0000000000000003: Join as router.

   These log messages confirm that the switch is configured correctly, ARP handling is set up, L2 switching is operational, and the router is ready to manage the routing table updates cyclically.

3. **Configure Ports Using CURL:** Next, we configure the ports on the router using CURL commands to specify gateway interfaces. The command format is:

   curl -X POST -d '"gateway": "<INTERFACE IP>"' http://localhost:8080/router/<SWITCH[_]ID>

   Here is an explanation of the components in the command format:

   - curl is a command-line tool used for transferring data with URLs. In this context, it is utilized to send HTTP requests to configure ports on the router.
   - The -X POST option specifies that the HTTP request method is POST. This method is commonly used to submit data to be processed by the server.

- The -d option is used to send data in the HTTP POST request. In this case, the data being sent is in JSON format and includes the gateway information. Replace <INTERFACE IP> with the appropriate IP address that corresponds to the gateway interface being configured.
- This part of the command specifies the URL to which the HTTP POST request is sent. Replace <SWITCH_ID> with the specific switch ID of the router where the port configuration is being applied.

Replace <INTERFACE IP> with the appropriate IP address and <SWITCH[_]ID> with the specific switch ID.

4. **Configure IP Addresses on Gateways:** In Step 4 of the network setup process, configuring IP addresses on the gateways (MTG 1 and MTG 2) is crucial to establish connectivity between the gateways and the routers. This configuration ensures that the gateways can effectively communicate with the routers, enabling proper data transmission paths between the hosts and routers through the gateways. Proper configuration of IP addresses on the gateways facilitates seamless communication paths between hosts and routers through these intermediary devices. By ensuring that the gateways have valid IP addresses, network traffic can be efficiently routed between different network segments, enhancing overall network connectivity and performance.

5. **Adding Static Routes:** To facilitate routing between different subnets, add static routes using the following CURL commands:

   curl -X POST -d '"destination": "192.168.115.0/24", "gateway": "192.168.113.2"' http://localhost:8080/router/0000000000000002

   curl -X POST -d '"destination": "192.168.116.0/24", "gateway": "192.168.114.2"' http://localhost:8080/router/0000000000000002

   The provided CURL commands are used to add static routes for two distinct destination subnets (192.168.115.0/24 and 192.168.116.0/24) through a shared gateway linked to a specific router identified by the switch ID 0000000000000002. Each CURL command includes the following components:

   The usage of the -X POST option to specify the HTTP request method as POST.

   The -d option is followed by a JSON payload that defines the destination subnet and the gateway IP address for routing.
   The URL http://localhost:8080/router/0000000000000002, where the POST request is directed to set up the static route on the router associated with the provided switch ID. These commands define routes for specific destination subnets through designated gateways, enabling the network to forward packets to the correct destinations.

6. **Assign Initial IP and Gateway to Hosts:** We configure the hosts with initial IP addresses and gateways to ensure they can communicate within the network. This involves setting the IP configuration manually or through a DHCP server. To identify and facilitate communication with other networked devices, every host in the system requires a distinct IP address. Network

administrators have the option to allocate IP addresses initially manually or automatically using a Dynamic Host Configuration Protocol (DHCP) server. Hosts must be configured with gateway information in addition to IP address. Data traffic between the local network and external networks enters and exits through the gateway. Hosts can determine where to deliver data packets meant for destinations outside of their local network by defining the gateway. Topology Diagram is shown in figure 6.1.

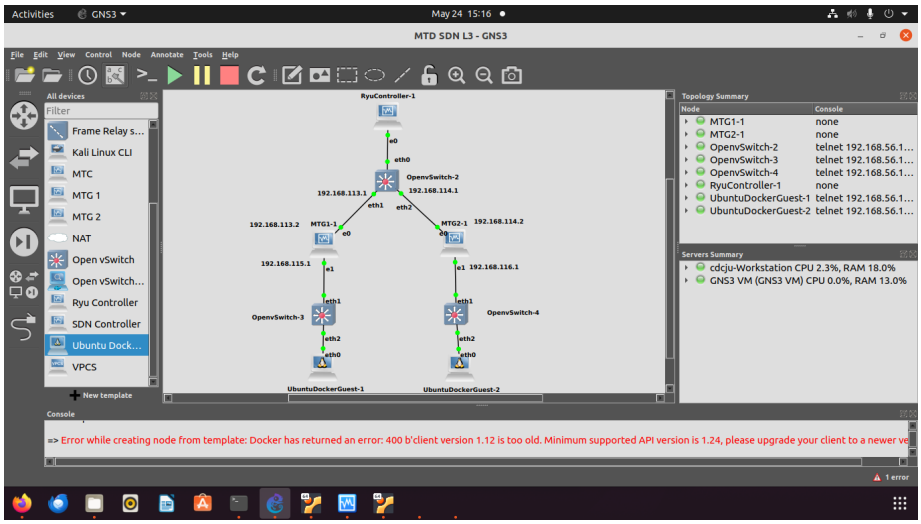


Figure 6.1: Network Topology with IP Shuffling Technique

7. **Start the DHCP Script on Gateways:** Start the DHCP Script on Gateways," the process involves initiating a DHCP server on the gateways (MYG1, MTG2) to automate the assignment of IP addresses to devices within the network.

    - **Enabling DHCP Server:** Setting up a DHCP server on gateways allows network administrators to manage and distribute IP addresses centrally. The DHCP server dynamically assigns IP addresses, subnet masks, gateway information, and other network parameters to connected devices, thereby simplifying the network configuration process.
    - **Updating DHCP Server IP Addresses:** The mentioned script is used to update the DHCP server with new IP addresses, distributing these addresses to devices after a specified period. This dynamic allocation ensures that devices receive different addresses each time they connect to the network, which enhances both security and network management.
    - **Automatic IP Address Assignment:** The process of leasing out new IP addresses at regular intervals automates the IP address assignment, adding randomness to the addresses. This practice supports the Network Access Server (NAS) feature of Moving Target Defense (MTD), where dynamically changing network configurations increase complexity and unpredictability, thereby enhancing security against potential attacks.
    - **Port Hopping (PH) in MTD:** The script also facilitates the implementation of Port Hopping (PH) on host machines as part of MTD. PH involves periodically changing the

ports on which services are running. By altering the ports at specified intervals, the network becomes less vulnerable to port-based attacks, making it more challenging for attackers to exploit static port configurations and gain unauthorized access.

8. **Topology Verification:** To verify the topology setup, observe the logs generated during the experiment. These logs provide detailed insights into the configuration and operational status of the network components, confirming the successful implementation of the desired network topology. All logs are in Figure 6.2.

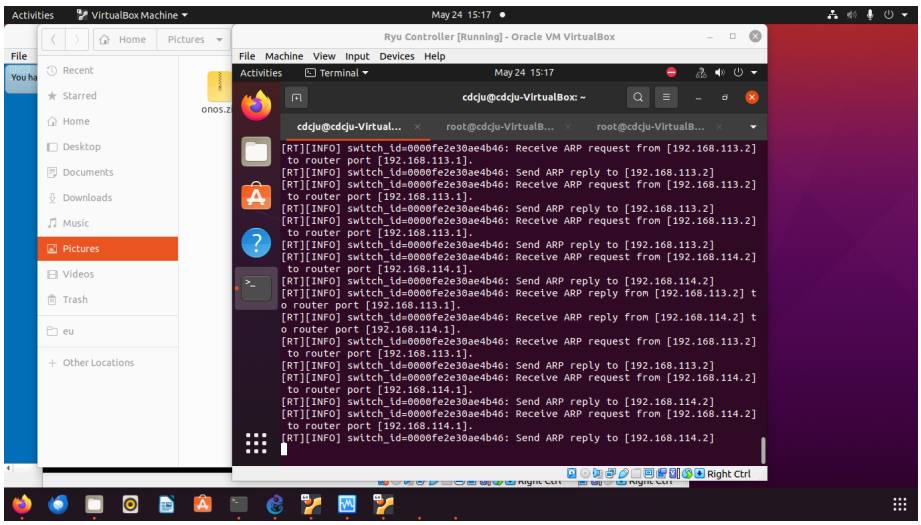


Figure 6.2: Network Topology Logs

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In conclusion, *"Challenges in Moving Target Defense Techniques"* makes significant contributions to the field of Computer Science and Engineering by addressing crucial issues in network security. Through an in-depth exploration of Moving Target Defense (MTD) strategies and the utilization of attack graph-based security analytics, the thesis sheds light on the evolving landscape of cybersecurity threats and the need for proactive defense mechanisms. The research statement of the thesis focused on evaluating the suitability of attack graph-based security analytics for implementing MTD strategies. By introducing the Three-Tier Attack Graph (TAG) concept and examining its benefits in the context of MTD, the thesis provides a novel perspective on enhancing network security through dynamic defense mechanisms. This contribution not only expands the existing knowledge base but also offers practical insights for improving the resilience of modern networks against sophisticated cyber threats. Furthermore, the thesis delves into a comprehensive review of recent MTD methods, emphasizing their operational mechanisms, target threats, and integration of threat models. By identifying and analyzing the challenges associated with key MTD components, the research highlights areas for further exploration and refinement in the development of effective defense strategies. This critical evaluation sets the stage for future advancements in MTD research and implementation, paving the way for more robust and adaptive security solutions.

The experiment successfully demonstrated the configuration and routing within an SDN environment using Open vSwitch (OVS) and the Ryu controller. By establishing communication between Ryu and OVS, configuring router ports and IP addresses, adding static routes, and automating IP assignments with a DHCP server, the network topology was effectively managed. The use of the rest_router(L2) application enhanced network security and efficiency. The final verification through logs confirmed the proper setup and functionality of the network, showcasing the potential of SDN in dynamic and secure network management.

The contributions of this research extend beyond theoretical frameworks to practical applications, offering valuable insights for enhancing the security posture of modern networks and mitigating the evolving threats posed by cyber adversaries. As the cybersecurity landscape continues to evolve, the findings of this thesis serve as a foundation for further research and development in the field of network security and defense.

7.2 Future Work

1. **Development of Lightweight MTD Solutions:** Considering the limited resources of IoT devices, there is a necessity to create lightweight Moving Target Defense (MTD) solutions. These solutions should efficiently safeguard these devices without negatively impacting their performance.

2. **Addressing Security Lapses:** Considering the significant repercussions of security breaches in IoT on mainstream infrastructure, future endeavors can focus on crafting robust Moving Target Defense (MTD) techniques. These techniques aim to secure the IoT ecosystem, thereby bolstering security for mainstream infrastructure.

3. **Integration of Software Defined Networking (SDN):** Future research efforts could concentrate on refining and improving the efficacy of Software-Defined Networking (SDN) solutions for implementing Moving Target Defense (MTD) in IoT devices. This focus aims to optimize SDN-based MTD techniques specifically tailored for enhancing security in IoT environments.

4. **Application of MTD to Kill-Chain Phases:** Although existing research predominantly targets thwarting attackers during the reconnaissance phase, forthcoming investigations could delve into the feasibility of deploying Moving Target Defense (MTD) throughout subsequent phases of the Intrusion Kill Chain Model. A promising avenue could involve inhibiting attackers from exfiltrating pilfered data, constituting an "Action on Objective" stage in the kill chain.

5. **Hybrid MTD:** Hybrid Moving Target Defense (MTD) research, which involves integrating multiple strategies to safeguard against a broader spectrum of attackers, is crucial. Although some studies have investigated hybrid MTD techniques, further exploration is necessary to determine the optimal times and methods for deploying various strategies. This approach aims to enhance protection while minimizing network impact.

6. **Machine Learning(ML) and Artificial Intelligence (AI):** By incorporating machine learning and AI approaches into MTD, it is possible to increase the effectiveness of detection and reaction against complex and advanced assaults. These technologies can assist in finding trends, anomalies, and potential attack routes by analyzing vast amounts of data, enabling automated decision-making and adaptive defense systems.

7. **Privacy-Preserving Techniques:** Exploring privacy-preserving mechanisms for sharing network data and attack information is vital, particularly in safeguarding sensitive data and adhering to data protection regulations. This research aims to develop methods that enable the sharing of pertinent information for security purposes while preserving privacy and ensuring compliance with relevant data privacy laws and regulations.

8. **Collaborative Defense Approaches:** Exploring collaborative defense approaches that involve coordinated efforts among different entities, such as organizations, vendors, and researchers, to collectively improve network security and resilience.

9. **Flexible MTD techniques must be established:** The principles and methods of adaptive MTD are still in their infancy in terms of state-of-the-art MTD technology. Based on the

degree of system vulnerabilities or attack patterns/strength, adaptivity in triggering MTD activities can be achieved, which necessitates the sophisticated detection or learning capabilities of the defender. For the defense to choose the best MTD deployment, they must understand how to learn an attacker's conduct or learn system security conditions.

10. **Collaborative and Distributed MTD:** Another interesting area of research is to examine how MTD approaches can be distributedly and cooperatively implemented across networks and systems. Investigating decentralized MTD architectures, distributed threat intelligence sharing, and coordinated defense techniques are all part of this effort to safeguard interconnected systems and networks.

11. **Enhanced Security Features and Integration:**Future research should aim at incorporating advanced security functionalities within the SDN framework using the Ryu controller and Open vSwitch. This involves the implementation and evaluation of security applications like Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) within the SDN environment. Utilizing the flexibility of SDN, researchers can develop dynamic security policies that can adapt in real-time to new threats. Additionally, incorporating other security technologies, such as blockchain for secure transaction logging and AI/ML algorithms for anomaly detection, can substantially improve the network's security.

Bibliography

[1] A. Abbasi, J. Wetzels, T. Holz, and S. Etalle. Challenges in designing exploit mitigations for deeply embedded systems. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 31–46. IEEE, 2019.

[2] S. Achleitner, T. La Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha. Cyber deception: Virtual networks to defend insider reconnaissance. In *Proceedings of the 8th ACM CCS international workshop on managing insider security threats*, pages 57–68, 2016.

[3] S. Achleitner, T. F. La Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha. Deceiving network reconnaissance using sdn-based virtual topologies. *IEEE Transactions on Network and Service Management*, 14(4):1098–1112, 2017.

[4] N. O. Ahmed and B. Bhargava. Mayflies: A moving target defense framework for distributed systems. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, pages 59–64, 2016.

[5] E. Al-Shaer, Q. Duan, and J. H. Jafarian. Random host mutation for moving target defense. In *International Conference on Security and Privacy in Communication Systems*, pages 310–327. Springer, 2012.

[6] H. Alavizadeh, J. Jang-Jaccard, and D. S. Kim. 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)*, pages 573–578. IEEE, 2018.

[7] H. Alavizadeh, D. S. Kim, J. B. Hong, and J. Jang-Jaccard. Effective security analysis for combinations of mtd techniques on cloud computing (short paper). In *Information Security Practice and Experience: 13th International Conference, ISPEC 2017, Melbourne, VIC, Australia, December 13–15, 2017, Proceedings 13*, pages 539–548. Springer, 2017.

[8] M. Albanese, A. De Benedictis, S. Jajodia, and K. Sun. A moving target defense mechanism for manets based on identity virtualization. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 278–286. IEEE, 2013.

[9] M. Albanese, S. Jajodia, and S. Noel. Time-efficient and cost-effective network hardening using attack graphs. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pages 1–12. IEEE, 2012.

[10] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials*, 21(2):1851–1877, 2019.

[11] A. Aydeger, N. Saputro, K. Akkaya, and M. Rahman. Mitigating crossfire attacks using sdn-based moving target defense. In *2016 IEEE 41st conference on local computer networks (LCN)*, pages 627–630. IEEE, 2016.

[12] M. Azab, R. Hassan, and M. Eltoweissy. Chameleonsoft: A moving target defense system. In *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 241–250. IEEE, 2011.

[13] S. A. Baho and J. Abawajy. Analysis of consumer iot device vulnerability quantification frameworks. *Electronics*, 12(5):1176, 2023.

[14] N. Bandi, H. Tajbakhsh, and M. Analoui. Fastmove: Fast ip switching moving target defense to mitigate ddos attacks. In *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–7. IEEE, 2021.

[15] G.-l. Cai, B.-s. Wang, W. Hu, and T.-z. Wang. Moving target defense: state of the art and characteristics. *Frontiers of Information Technology & Electronic Engineering*, 17(11):1122–1153, 2016.

[16] T. E. Carroll, M. Crouse, E. W. Fulp, and K. S. Berenhaut. Analysis of network address shuffling as a moving target defense. In *2014 IEEE international conference on communications (ICC)*, pages 701–706. IEEE, 2014.

[17] V. Casola, A. De Benedictis, and M. Albanese. A moving target defense approach for protecting resource-constrained distributed devices. In *2013 IEEE 14th International Conference on Information Reuse & Integration (IRI)*, pages 22–29. IEEE, 2013.

[18] X. Chai, Y. Wang, C. Yan, Y. Zhao, W. Chen, and X. Wang. Dq-motag: deep reinforcement learning-based moving target defense against ddos attacks. In *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, pages 375–379. IEEE, 2020.

[19] S.-Y. Chang, Y. Park, and B. B. A. Babu. Fast ip hopping randomization to secure hop-by-hop access in sdn. *IEEE Transactions on Network and Service Management*, 16(1):308–320, 2018.

[20] L. Chen and A. Avizienis. N-version programming: A fault-tolerance approach to reliability of software operation. In *Proc. 8th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-8)*, volume 1, pages 3–9, 1978.

[21] P. Chen, L. Desmet, and C. Huygens. A study on advanced persistent threats. In *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 International Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings 15*, pages 63–72. Springer, 2014.

[22] S. Cho, Y. Alavizadeh, M. Ben-Asher, and L. Kim. Nelson, 2020 cho j. *H., Sharma DP, Alavizadeh H., Yoon S., Ben-Asher N., Moore TJ, Kim DS, Lim H., Nelson FF, Toward proactive, adaptive defense: a survey on moving target defense, IEEE Communications Surveys & Tutorials*, 22(1):709–745, 2020.

[23] A. Chowdhary, A. Alshamrani, D. Huang, and H. Liang. Mtd analysis and evaluation framework in software defined network (mason). In *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pages 43–48, 2018.

[24] A. Chowdhary, D. Huang, A. Sabur, N. Vadnere, M. Kang, and B. Montrose. Sdn-based moving target defense using multi-agent reinforcement learning. In *Proceedings of the first International Conference on Autonomous Intelligent Cyber defense Agents (AICA 2021), Paris, France*, pages 15–16, 2021.

[25] A. Chowdhary, S. Pisharody, A. Alshamrani, and D. Huang. Dynamic game based security framework in sdn-enabled cloud networking environments. In *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pages 53–58, 2017.

[26] A. Chowdhary, S. Sengupta, A. Alshamrani, D. Huang, and A. Sabur. Adaptive mtd security using markov game modeling. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 577–581. IEEE, 2019.

[27] A. Clark, K. Sun, L. Bushnell, and R. Poovendran. A game-theoretic approach to ip address randomization in decoy-based cyber defense. In *Decision and Game Theory for Security: 6th International Conference, GameSec 2015, London, UK, November 4-5, 2015, Proceedings 6*, pages 3–21. Springer, 2015.

[28] A. Clark, K. Sun, and R. Poovendran. Effectiveness of ip address randomization in decoy-based moving target defense. In *52nd IEEE Conference on Decision and Control*, pages 678–685. IEEE, 2013.

[29] R. Colbaugh and K. Glass. Predictability-oriented defense against adaptive adversaries. In *2012 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 2721–2727. IEEE, 2012.

[30] A. Cui and S. J. Stolfo. Symbiotes and defensive mutualism: Moving target defense. In *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, pages 99–108. Springer, 2011.

[31] S. Debroy, P. Calyam, M. Nguyen, R. L. Neupane, B. Mukherjee, A. K. Eeralla, and K. Salah. Frequency-minimal utility-maximal moving target defense against ddos in sdn-based systems. *IEEE Transactions on Network and Service Management*, 17(2):890–903, 2020.

[32] S. Debroy, P. Calyam, M. Nguyen, A. Stage, and V. Georgiev. Frequency-minimal moving target defense using software-defined networking. In *2016 international conference on computing, networking and communications (ICNC)*, pages 1–6. IEEE, 2016.

[33] P. Deshpande, S. C. Sharma, S. K. Peddoju, and S. Junaid. Hids: A host based intrusion detection system for cloud computing environment. *International Journal of System Assurance Engineering and Management*, 9:567–576, 2018.

[34] M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront. Mt6d: A moving target ipv6 defense. In *2011-MILCOM 2011 Military Communications Conference*, pages 1321–1326. IEEE, 2011.

[35] I. El Mir, A. Chowdhary, D. Huang, S. Pisharody, D. S. Kim, and A. Haqiq. Software defined stochastic model for moving target defense. In *Proceedings of the Third International Afro-European Conference for Industrial Advancement—AECIA 2016*, pages 188–197. Springer, 2018.

[36] X. Feng, Z. Zheng, D. Cansever, A. Swami, and P. Mohapatra. A signaling game model for moving target defense. In *IEEE INFOCOM 2017-IEEE conference on computer communications*, pages 1–9. IEEE, 2017.

[37] A. Forni. Gartner identifies the top 10 strategic technology trends for 2017. *URL: https://www. gartner. com/en/newsroom/press-releases/2016-10-18-gartner-identifies-the-top-10-strategic-technology-trends-for-2017*, 2016.

[38] K. Glass and R. Colbaugh. Web analytics for security informatics. In *2011 European Intelligence and Security Informatics Conference*, pages 214–219. IEEE, 2011.

[39] A. Gorbenko, V. Kharchenko, and A. Romanovsky. Using inherent service redundancy and diversity to ensure web services dependability. *Methods, Models and Tools for Fault Tolerance*, pages 324–341, 2009.

[40] C. Gudla and A. H. Sung. Moving target defense discrete host address mutation and analysis in sdn. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 55–61. IEEE, 2020.

[41] D. Gurung, S. Gurung, and M. P. Pradhan. Moving target defense-challenges, problems and suitability for internet of things. In *2023 4th International Conference on Computing and Communication Systems (I3CS)*, pages 1–4. IEEE, 2023.

[42] J. Homer, S. Zhang, X. Ou, D. Schmidt, Y. Du, S. R. Rajagopalan, and A. Singhal. Aggregating vulnerability metrics in enterprise networks using attack graphs. *Journal of Computer Security*, 21(4):561–597, 2013.

[43] J. B. Hong and D. S. Kim. Assessing the effectiveness of moving target defenses using security models. *IEEE Transactions on Dependable and Secure Computing*, 13(2):163–177, 2015.

[44] J. B. Hong and D. S. Kim. Towards scalable security analysis using multi-layered security models. *Journal of Network and Computer Applications*, 75:156–168, 2016.

[45] J. B. Hong, S. Yoon, H. Lim, and D. S. Kim. Optimal network reconfiguration for software defined networks using shuffle-based online mtd. In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pages 234–243. IEEE, 2017.

[46] Y. Huang and A. K. Ghosh. Introducing diversity and uncertainty to create moving attack surfaces for web services. In *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, pages 131–151. Springer, 2011.

[47] M. F. Hyder and T. Fatima. Towards crossfire distributed denial of service attack protection using intent-based moving target defense over software-defined networking. *IEEE Access*, 9:112792–112804, 2021.

[48] J. H. Jafarian, E. Al-Shaer, and Q. Duan. 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*, pages 127–132, 2012.

[49] J. H. Jafarian, E. Al-Shaer, and Q. Duan. An effective address mutation approach for disrupting reconnaissance attacks. *IEEE Transactions on Information Forensics and Security*, 10(12):2562–2577, 2015.

[50] J. H. H. Jafarian, E. Al-Shaer, and Q. Duan. Spatio-temporal address mutation for proactive cyber agility against sophisticated attackers. In *Proceedings of the First ACM Workshop on Moving Target Defense*, pages 69–78, 2014.

[51] S. Jajodia, N. Park, E. Serra, and V. Subrahmanian. Share: A stackelberg honey-based adversarial reasoning engine. *ACM Transactions on Internet Technology (TOIT)*, 18(3):1–41, 2018.

[52] Ł. Jalowski, M. Zmuda, and M. Rawski. A survey on moving target defense for networks: A practical view. *Electronics*, 11(18):2886, 2022.

[53] Q. Jia, K. Sun, and A. Stavrou. Motag: Moving target defense against internet denial of service attacks. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2013.

[54] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell. Catch me if you can: A cloud-enabled ddos defense. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 264–275. IEEE, 2014.

[55] E. John, C. Kalu, and P. Asuquo. Comparative performance analysis of cybersecurity tools on a wireless network with wpa2 encryption.

[56] P. Johnson and M. Ekstedt. The tarpit–a general theory of software engineering. *Information and Software Technology*, 70:181–203, 2016.

[57] P. Johnson, R. Lagerström, M. Ekstedt, and U. Franke. Can the common vulnerability scoring system be trusted? a bayesian analysis. *IEEE Transactions on Dependable and Secure Computing*, 15(6):1002–1015, 2016.

[58] P. Kampanakis, H. Perros, and T. Beyene. Sdn-based solutions for moving target defense network protection. In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pages 1–6. IEEE, 2014.

[59] L. Kandle. Penetration testing in a small business network. 2022.

[60] A. D. Keromytis, R. Geambasu, S. Sethumadhavan, S. J. Stolfo, J. Yang, A. Benameur, M. Dacier, M. Elder, D. Kienzle, and A. Stavrou. The meerkats cloud security architecture. In *2012 32nd International Conference on Distributed Computing Systems Workshops*, pages 446–450. IEEE, 2012.

[61] M. Khosravi-Farmad, A. A. Ramaki, and A. G. Bafghi. Moving target defense against advanced persistent threats for cybersecurity enhancement. In *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 280–285. IEEE, 2018.

[62] G. H. Kim and E. H. Spafford. The design and implementation of tripwire: A file system integrity checker. In *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pages 18–29, 1994.

[63] A. Kyriakou and N. Sklavos. Container-based honeypot deployment for the analysis of malicious activity. In *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–4. IEEE, 2018.

[64] S. Lee, S. Lee, J.-R. Lee, and H.-y. Youm. Extraction of network threat signatures using latent dirichlet allocation. *Journal of Internet Computing and Services*, 19(1):1–10, 2018.

[65] D. J. Leversage and E. J. Byres. Estimating a system's mean time-to-compromise. *IEEE Security & Privacy*, 6(1):52–60, 2008.

[66] Y. Li, R. Dai, and J. Zhang. Morphing communications of cyber-physical systems towards moving-target defense. In *2014 IEEE international conference on communications (ICC)*, pages 592–598. IEEE, 2014.

[67] Y. Liu and H. Man. Network vulnerability assessment using bayesian networks. In *Data mining, intrusion detection, information assurance, and data networks security 2005*, volume 5812, pages 61–71. SPIE, 2005.

[68] Y.-B. Luo, B.-S. Wang, and G.-L. Cai. Effectiveness of port hopping as a moving target defense. In *2014 7th International Conference on Security Technology*, pages 7–10. IEEE, 2014.

[69] Y.-B. Luo, B.-S. Wang, X.-F. Wang, X.-F. Hu, G.-L. Cai, and H. Sun. Rpah: Random port and address hopping for thwarting internal and external adversaries. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 263–270. IEEE, 2015.

[70] Y.-b. Luo, B.-s. Wang, X.-f. Wang, and B.-f. Zhang. A keyed-hashing based self-synchronization mechanism for port address hopping communication. *Frontiers of Information Technology & Electronic Engineering*, 18(5):719–728, 2017.

[71] D. C. MacFarland and C. A. Shue. The sdn shuffle: Creating a moving-target defense using host-based software-defined networking. In *Proceedings of the second ACM workshop on moving target defense*, pages 37–41, 2015.

[72] S. Macwan and C.-H. Lung. Investigation of moving target defense technique to prevent poisoning attacks in sdn. In *2019 IEEE World Congress on Services (SERVICES)*, volume 2642, pages 178–183. IEEE, 2019.

[73] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan. Internet of things (iot) security: Current status, challenges and prospective measures. In *2015 10th international conference for internet technology and secured transactions (ICITST)*, pages 336–341. IEEE, 2015.

[74] H. Maleki, S. Valizadeh, W. Koch, A. Bestavros, and M. Van Dijk. Markov modeling of moving target defense games. In *Proceedings of the 2016 ACM workshop on moving target defense*, pages 81–92, 2016.

[75] P. K. Manadhata. Game theoretic approaches to attack surface shifting. In *Moving target defense II: Application of game theory and adversarial modeling*, pages 1–13. Springer, 2012.

[76] H. Marco-Gisbert and I. Ripoll Ripoll. Address space layout randomization next generation. *Applied Sciences*, 9(14):2928, 2019.

[77] P. Mehra. A brief study and comparison of snort and bro open source network intrusion detection systems. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(6):383–386, 2012.

[78] M. Nadeem, S. W. Zahra, M. N. Abbasi, A. Arshad, and S. Riaz. A comprehensive examination of cloudflare's ip-based distributed denial of service mitigation. recent trends in parallel computing. 2023; 10 (2): 45–54p. *A Comprehensive Examination of Cloudflare's IP-based Nadeem et al. STM Journals*, page 2, 2023.

[79] R. E. Navas, H. Sandaker, F. Cuppens, N. Cuppens, L. Toutain, and G. Z. Papadopoulos. Ianvs: A moving target defense framework for a resilient internet of things. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE, 2020.

[80] S. Neti, A. Somayaji, and M. E. Locasto. Software diversity: Security, entropy and game theory. In *HotSec*, 2012.

[81] S. Noel, S. Jajodia, B. O'Berry, and M. Jacobs. Efficient minimum-cost network hardening via exploit dependency graphs. In *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, pages 86–95. IEEE, 2003.

[82] H. Okhravi, W. W. Streilein, and K. S. Bauer. Moving target techniques: Leveraging uncertainty for cyber defense. *Lincoln Laboratory Journal*, 22(1):100–109, 2016.

[83] W. Peng, F. Li, C.-T. Huang, and X. Zou. A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces. In *2014 IEEE International Conference on Communications (ICC)*, pages 804–809. IEEE, 2014.

[84] J. Qian, S. Hinrichs, and K. Nahrstedt. Acla: A framework for access control list (acl) analysis and optimization. In *Communications and Multimedia Security Issues of the New Century: IFIP TC6/TC11 Fifth Joint Working Conference on Communications and Multimedia Security (CMS'01) May 21–22, 2001, Darmstadt, Germany*, pages 197–211. Springer, 2001.

[85] M. Rawski. Network topology mutation as moving target defense for corporate networks. *International Journal of Electronics and Telecommunications*, 65, 2019.

[86] W. Reese. Nginx: the high-performance web server and reverse proxy. *Linux Journal*, 2008(173):2, 2008.

[87] R. U. Rehman. *Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID*. Prentice Hall Professional, 2003.

[88] J. Rowe, K. N. Levitt, T. Demir, and R. Erbacher. Artificial diversity as maneuvers in a control theoretic moving target defense. In *National Symposium on Moving Target Research*, 2012.

[89] J. Sadowski. Zero tolerance: More zero-days exploited in 2021 than ever before. *Mandiant. Retrieved April*, 12:2023, 2022.

[90] N. Saputro, S. Tonyali, A. Aydeger, K. Akkaya, M. A. Rahman, and S. Uluagac. A review of moving target defense mechanisms for internet of things applications. *Modeling and Design of Secure Internet of Things*, pages 563–614, 2020.

[91] R. Sawilla and D. Skillicorn. Partial cuts in attack graphs for cost effective network defence. In *2012 IEEE Conference on Technologies for Homeland Security (HST)*, pages 291–297. IEEE, 2012.

[92] S. Sengupta, A. Chowdhary, D. Huang, and S. Kambhampati. Moving target defense for the placement of intrusion detection systems in the cloud. In *Decision and Game Theory for Security: 9th International Conference, GameSec 2018, Seattle, WA, USA, October 29–31, 2018, Proceedings 9*, pages 326–345. Springer, 2018.

[93] S. Sengupta, A. Chowdhary, D. Huang, and S. Kambhampati. General sum markov games for strategic detection of advanced persistent threats using moving target defense in cloud networks. In *Decision and Game Theory for Security: 10th International Conference, GameSec 2019, Stockholm, Sweden, October 30–November 1, 2019, Proceedings 10*, pages 492–512. Springer, 2019.

[94] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati. A survey of moving target defenses for network security. *IEEE Communications Surveys & Tutorials*, 22(3):1909–1941, 2020.

[95] S. Sengupta, S. G. Vadlamudi, S. Kambhampati, A. Doupé, Z. Zhao, M. Taguinod, and G.-J. Ahn. A game theoretic approach to strategy generation for moving target defense in web applications. In *AAMAS*, volume 1, pages 178–186, 2017.

[96] D. P. Sharma, D. S. Kim, S. Yoon, H. Lim, J.-H. Cho, and T. J. Moore. Frvm: Flexible random virtual ip multiplexing in software-defined networks. 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)*, pages 579–587. IEEE, 2018.

[97] Y. Shi, H. Zhang, J. Wang, F. Xiao, J. Huang, D. Zha, H. Hu, F. Yan, B. Zhao, et al. Chaos: An sdn-based moving target defense system. *Security and Communication Networks*, 2017, 2017.

[98] R. Sobers. Must-know data breach statistics for 2021. *Available online: varonis. com/blog/data-breach-statistics/(accessed on 16 June 2021)*, 98.

[99] J. Steinberger, B. Kuhnert, C. Dietz, L. Ball, A. Sperotto, H. Baier, A. Pras, and G. Dreo. Ddos defense using mtd and sdn. In *NOMS 2018-2018 IEEE/IFIP network operations and management symposium*, pages 1–9. IEEE, 2018.

[100] N. Stojanovski, M. Gusev, D. Gligoroski, and S. J. Knapskog. Bypassing data execution prevention on microsoftwindows xp sp2. In *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pages 1222–1226. IEEE, 2007.

[101] J. Sun and K. Sun. Desir: Decoy-enhanced seamless ip randomization. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

[102] M. Taguinod, A. Doupé, Z. Zhao, and G.-J. Ahn. Toward a moving target defense for web applications. In *2015 IEEE international conference on information reuse and integration*, pages 510–517. IEEE, 2015.

[103] M. Thompson, N. Evans, and V. Kisekka. Multiple os rotational environment an implemented moving target defense. In *2014 7th International Symposium on Resilient Control Systems (ISRCS)*, pages 1–6. IEEE, 2014.

[104] M. Ussath, D. Jaeger, F. Cheng, and C. Meinel. Advanced persistent threats: Behind the scenes. In *2016 Annual Conference on Information Science and Systems (CISS)*, pages 181–186. IEEE, 2016.

[105] M. Van Dijk, A. Juels, A. Oprea, and R. L. Rivest. Flipit: The game of "stealthy takeover". *Journal of Cryptology*, 26:655–713, 2013.

[106] S. Venkatesan, M. Albanese, K. Amin, S. Jajodia, and M. Wright. A moving target defense approach to mitigate ddos attacks against proxy-based architectures. In *2016 IEEE conference on communications and network security (CNS)*, pages 198–206. IEEE, 2016.

[107] S. Venkatesan, M. Albanese, G. Cybenko, and S. Jajodia. A moving target defense approach to disrupting stealthy botnets. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, pages 37–46, 2016.

[108] S. Vikram, C. Yang, and G. Gu. Nomad: Towards non-intrusive moving-target defense against web bots. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 55–63. IEEE, 2013.

[109] K. Wang, X. Chen, and Y. Zhu. Random domain name and address mutation (rdam) for thwarting reconnaissance attacks. *PloS one*, 12(5):e0177111, 2017.

[110] L. Wang. Shoal: A network level moving target defense engine with software defined networking. *EAI Endorsed Transactions on Security and Safety*, 7(25):e5–e5, 2020.

[111] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In *Data and Applications Security XXII: 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security London, UK, July 13-16, 2008 Proceedings 22*, pages 283–296. Springer, 2008.

[112] L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel. k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. *IEEE Transactions on Dependable and Secure Computing*, 11(1):30–44, 2013.

[113] L. Wang and D. Wu. Moving target defense against network reconnaissance with software defined networking. In *Information Security: 19th International Conference, ISC 2016, Honolulu, HI, USA, September 3-6, 2016. Proceedings 19*, pages 203–217. Springer, 2016.

[114] S. Wang, L. Zhang, and C. Tang. A new dynamic address solution for moving target defense. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 1149–1152. IEEE, 2016.

[115] Y. Wang, Q. Chen, J. Yi, and J. Guo. U-tri: Unlinkability through random identifier for sdn network. In *Proceedings of the 2017 Workshop on Moving Target Defense*, pages 3–15, 2017.

[116] B. C. Ward, S. R. Gomez, R. Skowyra, D. Bigelow, J. Martin, J. Landry, and H. Okhravi. Survey of cyber moving targets second edition. *MIT Lincoln Laboratory Lexington United States, Tech. Rep*, 2018.

[117] X. Xu, H. Hu, Y. Liu, H. Zhang, and D. Chang. An adaptive ip hopping approach for moving target defense using a light-weight cnn detector. *Security and Communication Networks*, 2021:1–17, 2021.

[118] S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim. Attack graph-based moving target defense in software-defined networks. *IEEE Transactions on Network and Service Management*, 17(3):1653–1668, 2020.

[119] E. Yuan, S. Malek, B. Schmerl, D. Garlan, and J. Gennari. Architecture-based self-protecting software systems. In *Proceedings of the 9th international ACM Sigsoft conference on Quality of software architectures*, pages 33–42, 2013.

[120] C. Zhang, L. Duan, T. Wei, and W. Zou. Secgot: Secure global offset tables in elf executables. In *Conference of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, pages 995–998. Atlantis Press, 2013.

[121] H. Zhang, K. Zheng, X. Wang, S. Luo, and B. Wu. Efficient strategy selection for moving target defense under multiple attacks. *IEEE Access*, 7:65982–65995, 2019.

[122] L. Zhang, Q. Wei, K. Gu, and H. Yuwen. Path hopping based sdn network defense technology. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 2058–2063. IEEE, 2016.

[123] Z. Zhao, D. Gong, B. Lu, F. Liu, C. Zhang, et al. Sdn-based double hopping communication against sniffer attack. *Mathematical Problems in Engineering*, 2016, 2016.

[124] Z. Zhao, F. Liu, and D. Gong. An sdn-based fingerprint hopping method to prevent fingerprinting attacks. *Security and Communication Networks*, 2017, 2017.

[125] J. Zheng and A. S. Namin. A survey on the moving target defense strategies: An architectural perspective. *Journal of Computer Science and Technology*, 34:207–233, 2019.

[126] Z. Zhou, C. Xu, X. Kuang, T. Zhang, and L. Sun. An efficient and agile spatio-temporal route mutation moving target defense mechanism. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[127] Q. Zhu and T. Başar. Game-theoretic approach to feedback-driven multi-stage moving target defense. In *Decision and Game Theory for Security: 4th International Conference, GameSec 2013, Fort Worth, TX, USA, November 11-12, 2013. Proceedings 4*, pages 246–263. Springer, 2013.

[128] R. Zhuang, S. Zhang, S. A. DeLoach, X. Ou, A. Singhal, et al. Simulation-based approaches to studying effectiveness of moving-target network defense. In *National symposium on moving target research*, volume 246. Citeseer, 2012.