Project Proposal: Integrating Time Domain in the Evaluation of Moving Target Defense

Wenxiao Zhang
Department of Computer Science and Software Engineering
The University of Western Australia
Perth, Australia

I. INTRODUCTION

With the development of network scale, cybersecurity has become a growing issue worldwide. Various cyberattacks have emerged and caused a huge amount of damage to both individuals and society as a whole [1]. In order to prevent these attacks and also to continuously change the system and network attack surface to confuse the attackers, Moving Target Defense (MTD) has been proposed [2]. Instead of eliminating all the vulnerabilities in the system, the core idea of it is to make the system become dynamic by continuously reconfiguring the system components in order to change the attack surface [4]. From this perspective, Introducing MTD to the system can help deal with the common issues of static nature that traditional security mechanisms have. For instance, some traditional Intrusion Detection Systems (IDS) and firewalls use static defense mechanisms that can have a higher risk of being exploited by Advanced Persistent Threat (APT), which can perform long-term vulnerability analysis and penetration testing on the target and is good at breaking static defenses [3] [4]. Such attack strategies can be effectively thwarted by applying MTD as it can dynamically move the components, which can effectively change the existing vulnerabilities of the system. Thus limiting the attack cycle and making it difficult for attackers to analyse vulnerabilities and exploit the system. Furthermore, In comparison with conventional defense mechanisms, MTD provides affordable defense opportunities as in many cases it can be deployed by utilising the existing system components and technologies instead of using a huge amount of time and resources to create a new defense solution [4].

Many approaches were proposed with different kinds of system modeling and metrics that can design, develop and evaluate MTD in multiple aspects [4]. However, not many of them considered the time domain in terms of modeling and assessing the effectiveness of specific MTD techniques deployed. Analytic modeling approaches to evaluate the performance impact of MTD were proposed previously [21] [22] [23]. The modeling of the time domain of these approaches was based on the assumption that the time duration of all events in the network system is exponentially distributed within a range of values referring to empirical observations. However, as a theoretical model, they only considered a generic MTD technique and did not investigate specific events that occurred when deploying specific MTD techniques. In addition, some of the previous researches have evaluated the effectiveness of combining multiple MTD techniques with time-based security metrics such as Attack Cost (AC) [10] [14]. However, the work also has limitations as the proposed MTD model was evaluated on the cloud environment only.

Accordingly, this project will evaluate the effectiveness of combining multiple MTD techniques in the time domain by modeling and analysing discrete events that occurred using simulations. This will be done using the existing implemented MTD simulator *MTDSim*, which was first developed by Brown [19] and further extended by Lee [20]. Further research into modeling and evaluations of combined MTD techniques in the time domain will be of great benefit for the cybersecurity of the modern IT industry. Since the time taken for actions of different events that occurred in the network may affect the evaluation of the performance of MTD greatly, exploring how MTD techniques can secure the system in the time domain can make the simulation results more reliable. Thus it will be a strong indication for IT companies to select reasonable MTD strategies to protect their network systems. The outline of this project proposal is as follows. Section II will provide the background of the proposed topic. Section III will investigate the related work of modeling and evaluating MTD with discrete events. Section IV will give details about the methodology and evaluation methods that will be applied to this project. Section V will show the Gantt Chart with the project timeline and demonstrate the tasks to be done.

II. BACKGROUND

A. MTD Classifications

Different MTD techniques can be categorised into different classifications [5]. The key principle of classifying MTD techniques revolve around these three questions [4]: what to move, when to move, and how to move.

1) What to Move: Multiple components of system that can be moved or reconfigured to change the attack surface. Cho et al. [4] summarised dozens of elements in the system that can be moved based on different MTD techniques. As it is shown in Table I, a large number of these components exist in the Application Layer and OS-Host/VM-Instance Layer.

2) When to Move: Timeliness-based MTD classification categorised MTD techniques into three different types to determine *when to move* [4]. Specifically, three categories are detailed as follows:

TABLE I: Moving elements of MTD techniques at different layers [4]

Layers	MTD Techniques		
	Shuffling	**Diversity**	**Redundancy**
Application	TCP/UDP ports	Web: Appache, IIS, GWS etc.	Web service replica
		App: .Net Framework, Java, PHP etc.	Application replica
		Database: SQL server, MySQL, Oracle etc.	Database backup and replica
		Others: Mail-server, Proxy-server etc.	Other service replica
OS-Host	IP address	Windows: Windows server 2003/2008, Windows 9.x, 8, 10 etc.	Host OS and VM replica
		Linux: Redhat, Debian, Caldera etc.	
		Solaris	
		Others: Unix, HP-UX etc.	
VM-Instance	Virtual IP address	Same as OS	
Virtual Machine Manager	Failover, Switchover	Xen	Hypervisor's replica
		Vmware	
		ESXi	
		Others: Kernel-based VM (KVM), Virtual-box (Vbox), IBM vSphere	
Hardware	Hardware replacement	Intel	Hardware backups and replica
		HP	
		Sun Solaris	
		Others: ARM, Atmega	

- **Time-based MTD**: This strategy triggers MTD operations periodically. All MTD operations are deployed based on a certain time interval, which is defined as *MTD interval*. *MTD interval* can be either a fixed time interval or an interval with a certain range of variation [6].
- **Event-based MTD**: This approach triggers MTD operations only when a certain event happens in the network. Events such as security alerts raised by Intrusion Detection System (IDS) can be a trigger to deploy MTD operation [30].
- **Hybrid**: Some MTD approaches take both periodically and event-driven strategies. These approaches deploy MTD operations adaptively based on current situations in the network.

Apart from these three categories, deploying MTD randomly without any triggering mechanism could be another possible approach. However, it is difficult to control and assess the effectiveness of a random defense mechanism as it is highly unstable.

3) How to Move: Operation-based MTD classifies MTD techniques into three different types to determine 'how to move' [4]. Specifically, each type of the MTD technique is detailed as follows:

- **Shuffling**: Operations aim to increase confusion and uncertainty for attackers by rearranging or randomizing system configurations.
- **Diversity**: Operations aim to improve system resilience by deploying system components with different implementations that can serve the same functionalities.
- **Redundancy**: Operations aim to increase system availability and reliability by providing multiple replicas of system components.

B. Adversary Tactics and Techniques

1) Cyber Kill Chain: Cyber Kill Chain (CKC) is a series of steps that the adversaries must complete in order to achieve their objective [16], which is always breaching the Confidentiality, Integrity, and Availability (CIA) of a system. The CKC model has been widely adopted in the cybersecurity industry to model the attack profile and evaluate the security level based on it. The steps of CKC are as follows [4]:

- **Reconnaissance**: The practice of discovering and collecting information about a system. The target of the attacker will be identified and selected during this step.
- **Weaponisation**: The practice of embedding malware into the deliverable payload. The malware contains the virus or/and the backdoor that can help exploit the target.
- **Delivery**: After the payload is created, the intruders can then deliver the above payload to the target.
- **Exploitation**: After the payload is delivered, the hacker can start exploiting the target by using the malware embedded in the payload.
- **Installation**: The practice of installing malware on the target.
- **Command and Control**: After the malware is installed, the intruders can take control of the target host by establishing a C2 channel.
- **Action on Objectives**: With the above six steps finished, the intruders can achieve their goal by breaching the CIA.

2) MITRE ATT&CK: MITRE ATT&CK is a new standard framework of adversary tactics and techniques based on real-world observations [17]. Unlike CKC, the tactics of MITRE ATT&CK are unordered and may not all take place in a single intrusion as the tactical objectives of the adversary can alter within an attack operation. The MITRE ATT&CK framework now has three different iterations [18]:

- **Enterprise**: The components that are present in traditional Information Technology (IT) threats and scenarios. It can also be divided based on cloud computing (XaaS) and operating systems (Windows, Linux, etc.).
- **Mobile**: The unique adversarial behavior found when attacking the operating systems (iOS, Android, etc) of the mobile device.
- **Industrial Control Systems (ICS)**: The features found in Operational Technology (OT) attacks and scenarios. The convergent nature of IT and OT will cause the features to overlap so that it is separate from Enterprise's ATT&CK framework.

III. RELATED WORK

This section will review and discuss the previous works related to the proposed project to find out their achievements as well as limitations, thereby coming up with appropriate

methodologies to integrate the time domain into modeling and assessing multiple MTD techniques. Therefore, Section III-A will discuss different combination strategies of multiple MTD techniques and their corresponding evaluation methods. Section III-B will investigate the existing models for simulating discrete events of MTD. Section III-C will investigate the existing metrics that can evaluate MTD in the time domain. Section will discuss different types of evaluation methods and the previous work *MTDSim*.

A. Combinations of Multiple MTD Techniques

As it is introduced in Section II-A3, MTD techniques can be classified into three categories: shuffling, diversity, and redundancy. Each technique uses a different approach to protect the system from a different perspective. For instance, shuffle-based MTD techniques are often designed to improve system security, while redundancy techniques usually aim to improve reliability and availability [8]. Investigating how multiple MTD techniques can be combined and evaluated can help improve the MTD scheduling strategy, thus enhancing the overall security and performance of the system.

1) Shuffling and Redundancy: Alavizadeh et al. [8] proposed an MTD approach that combined both shuffle and redundancy techniques (S+R) into a Hierarchical Attack Representation Model (HARM) [7]. They evaluated the effectiveness of the approach by measuring the *system risk* and *reliability*. The experiment was conducted on the cloud-band model. They found that in comparsion to the S-only technique, deploying S+R causes a gentle increment on the *system risk*, but the *reliability* of S+R is overall way more higher than S-only. In addition, comparing to R-only technique, deploying S+R can significantly decrease the *system risk*, but the *reliability* will be affected. Results showed the effectiveness of the proposed MTD approach when minimising *system risk* and maximising *reliability*. Later, they extended their work in [9] by introducing a new security metric *Unattackability*. By conducting experiments against different levels of attack rate, they found out that the proposed model can provide an estimation of the attack rate the system could withstand, as well as suggestions for security hardening that would help improve the security posture of the system to deal with higher attack rates.

2) Shuffling and Diversity: Alavizadeh et al. [10] analysed the combination of shuffling and diversity (S+D) techniques using a similar approach with [8] [9]. That is, the system modeling was also conducted based on HARM, and the simulation was also performed in the cloud. This time they chose to use three key metrics, including *Risk* (R), *Attack cost* (AC), and *Return on Attack* (RoA). They found that S+D approach can effectively satisfy the desirable evaluation criteria by minimising R and RoA while maximising AC, which is unachievable for either S-only or D-only approach. Subsequently, Alavizadeh et al. [11] further this research and explored multiple security metrics based on either attackers' or defenders' perspectives. For example, *Attack Success Probability* (ASP) and *Shortest Attack Path* (SAP) were selected as additional metrics to evaluate the MTD approach they proposed. They found that deploying shuffling strategies inappropriately can increase the SAP and ASP, which could lower the security level of the system.

3) Diversity and Redundancy: While either diversity or redundancy is usually combined with shuffling techniques in MTD implementation, literature related to the combination of Diversity and Redundancy (D+R) so far mainly focused on dealing with real-world issues by leveraging these two techniques instead of assessing and comparing the effectiveness of the combination of it with either D-only or R-only approach.

Laszka et al. [12] proposed a framework for integrating D+R with hardening techniques for Industrial Internet of Things(IIoT) system. They designed a high-level model of IIoT system to evaluate security risk, in addition to a security investment model and a security risk model to evaluate the effectiveness of combining R+D with hardening strategy. They evaluated their approach by introducing a case study of a real-world water distribution network. The result shows that integrating D+R with hardening strategy can effectively reduce security risk with a reasonable cost.

Alleg et al. [13] proposed a Virtual Network Function (VNF) placement solution by employing D+R to achieve resilient service function chain. The idea is to improve global service availability by optimizing resource utilization against vulnerable VNFs and their dependence on specific redundancy. Their analytical results proved the efficiency of their solution with respect to reduce resource consumption and improve service availability.

4) Shuffling, Diversity and Redundancy: Alavizadeh et al. [14] combined Shuffling, Diversity, and Redundancy (S+D+R). Similar to [8], [9], and [10], the authors carried on their investigations of the effectiveness of the combined MTD techniques based on HARM for the cloud environment. And in this time, they use four security metrics, which are R, RoA, AC, and system availability (SA). They found that all security metrics are improved after deploying S+D+R in comparison with deploying a single MTD technique. However, one of the limitations of their work is the lack of consideration of the system performance. As deploying multiple MTD techniques can lead to more resource consumption, the overall performance of the system could be affected, which may also have a negative impact in terms of Quality of Service (QoS).

5) Discussion: While previous work has proposed a variety of approaches to evaluate the performance of different combinations of multiple MTD technologies, few have treated the time domain as a major consideration. For instance, literature [9] and [10] applied some time-based metrics on the attacker's side to evaluate their proposed MTD model on the cloud-based environment. However, the evaluation methods in the time domain are insufficient as they did not fully consider the issues caused by the time duration of specific events occurred in the network. Time-related issues such as resource occupation conflicts that occurred during system reconfiguration can affect the overall MTD evaluation [23]. Thus, the proposed project will focus on tackle such issues by further integrating the time domain in the simulation. It will then come up with a further evaluation of the performance of the combination of multiple MTD techniques and an analysis of the pattern of time duration

between the attack and the specific MTD operations to get more constructive outcomes.

B. Discrete Event Simulation Model

Multiple events will occur in the network when the MTD approach is introduced. Modeling discrete events is an essential process for evaluating the effectiveness of combining multiple MTD approaches in the time domain. For example, as two different MTD techniques can move the same resources [4], conflicts will happen if such two MTD techniques are deployed at the same time, or if one MTD operation is deployed while another MTD operation is unfinished. In addition, other events that occur in the network such as attacks, and job requests, can also affect the result of the simulation. To fully investigate the impact of different events occurring in the network and to find a solution to deal with conflicts, modeling discrete event simulation is needed [15].

1) Petri Net Model: Petri Net has been proposed by Carl Adam Petri in 1962, and it is now one of the most popular models used to simulate discrete event systems [29]. It can improve the understanding of different kinds of events that occurred in the system, allowing a joint analysis of their behavior and structure of them.

Cai et al. [30] designed a model to evaluate three shuffle-based MTD techniques based on Generalized Stochastic Petri Nets (GSPN). It categorised the state transitions of MTD events into two sub-types: immediate transitions and timed transitions. Immediate transitions can be fired randomly and without time delay, while timed transitions are associated with a random delay time between enabling and firing. They deployed their approach on a web server system and they found that the GSPN model they proposed is suitable for the three MTD techniques and the combination of them. While the work is more comprehensive than most of the existing approaches, it still has limitations. As the MTD techniques they proposed in the model have only been deployed on the web server system, some of the specific approaches may not be applicable for modeling generic networks.

Torquato et al. [25] developed a tool named *PyMTDEvaluator* to assess the effectiveness of time-based MTD techniques against availability attacks such as Denial of Service (DoS) attacks and Resource Starvation (RS) attacks. *PyMTDEvaluator* was designed based on Stochastic Petri Net (SPN) model, whose assumptions were drawn from empirical observations. The SPN model they proposed can represent the main events such as MTD action, attack progress, and system downtime. The tool applied three metrics to evaluate MTD: SA, ASP, and System Capacity (SC). Accordingly, this tool provides an inspiring idea for the design of time-based MTD state transitions and the modeling of discrete events occurring in the network. However, the MTD movement they designed was generic, without considering different types of MTD techniques.

2) Markov Chain Model: The Markov chain is another approach to modeling the state transitions of events that happened in the system. It describes each event as a stochastic process with discrete states, with time and the present state to be the parameters of the future state [29]. In this scenario, the Markov chain can be used to capture MTD behaviors in the time domain, which can be used to analyse the performance based on the data of state transitions for each MTD event. This allows for making adaptive strategies on deployments to improve the overall performance of MTD.

Chen et al. [21] proposed two types of Markov process-based models for time-based MTD and event-based MTD respectively to evaluate the effectiveness of MTD by investigating how MTD affects system performance while protecting the system. The time-based MTD was designed to be deployed periodically while the event-based MTD was designed to be deployed when the system is idle (no job being processed on the server). They used three metrics to evaluate MTD: mean number of jobs waiting in the system, mean job sojourn time, and ASP. Based on the analysis of Markov process-based models, They found that event-based MTD can have a less negative influence on the performance of the server than time-based MTD. However, the job request stream they designed came along with the attack stream, whereas cyberattack in the real world usually does not perform like this. Due to the lack of consideration in modeling the attack profile, the ASP in the scenario of event-based MTD is incalculable.

Zheng and Namin [28] proposed a Markov Decision Process (MDP) modeling-based approach. It aims to discover optimal policies for the implementation and deployment of MTD by analysing the variation of system security and action cost within the transition of four different states from a highly abstract level. Through the simulation experiments, they found that the optimal strategy changes associated with the action cost, which means the MDP approach they proposed can be used to analyse the impact of costs of possible action, which can determine the optimal MTD strategy. However, this work used a mathematical model that highly abstracts the MTD action, it cannot be utilized in assessing the effectiveness of multiple MTD techniques.

Connell et al. [22] proposed a quantitative analytic model for evaluating the resource availability and performance of MTD. They used Continuous Time Markov Chains (CTMC) to compute the probability distribution of the number of resources being reconfigured and thus determine resource availability and other performance metrics such as response time. They validated their approach by implementing a simulation using SimPy [24]. The simulation results show the trade-off between security and performance when deploying MTD.

3) Discussion: Although the Petri Net model and Markov Chain Model are good at modeling and analysing the state transition of the discrete event, most of the existing approaches were focused on analytical modeling in which MTD techniques and attackers were highly abstracted. Thus, the implementation of discrete events in this proposed project will not be based on specific modeling strategies in the simulation. Nevertheless, It will draw on the details of the design of state transitions for specific discrete events presented in the literature.

C. Security Metrics

Appropriate security metrics need to be considered in order to measure the effectiveness and efficiency of multiple MTD techniques. Researchers usually evaluate MTD from two perspectives: from the attacker's perspective or the defender's perspective.

1) Attack Cost: Attack cost (AC) is one of the most popular security metrics in assessing the effectiveness of MTD by measuring the cost spent by the attackers when they exploit the vulnerabilities on a host machine. Alavizadeh et al. have combined AC with other metrics such as Return on Attack (RoA) and system risk to evaluate HARM-based MTD they proposed from both attackers' and defenders' perspectives [10] [11] [14]. In addition, AC, RoA, and system risk were also applied to the *MTDSim* as metrics for evaluating the effectiveness of combinations of multiple shuffle-based MTD techniques [19]. However, these metrics still have limitations on the current *MTDSim* as it is not sufficiently designed and developed in terms of the time domain.

2) Mean Time to Compromise: Mean Time to Compromise (MTTC) indicates how long an attacker takes to compromise a target host machine running on the network [4]. MTTC can also be recognized as Mean Time to Failure (MTTF) from the perspective of the defender.

Hong et al. [27] proposed a set of dynamic security metrics based on Temporal Hierarchical Attack Representation Model (T-HARM). The metrics can capture the security changes in the network and evaluate the effectiveness of MTD techniques. Based on that, Lee [20] applied Attack Path Variation (APV) and Attack Path Exposure (APE) to measure the effectiveness of MTD based on the shift of security posture in the *MTDSim*. In the aspect of the time domain, the authors proposed a metric called Attack Compromise Duration (ACD) which computes the MTTC of an attack on the target host in a given network state. ACD is also computed based on the T-HARM model, and it consists of a set of network states (S) and the time taken to exploit each vulnerability of each stepping stone in the attack path ($t\left(ap_i\right)$). The ACD is dynamic because of the changes in S and ($t\left(ap_i\right)$) caused by MTD operations.

3) Discussion: The security metrics of this project will focus on implementing time-based metrics. Since AC was already implemented in the previous work *MTDSim* but has limitations, this project will update the implementation of AC and introduce MTTC to further assess the performance of MTD.

D. Simulating MTD

Cho et al. [4] surveyed four types of existing evaluation methods applied to assess the performance of MTD techniques, including analytical models, emulation models, real test-bed environments, and simulation models.

For analytical models, the researchers found that most of them can provide a certain level of insights based on behaviors of a general system with a low evaluation cost but are incapable of capturing some deviations and effects in real application scenarios. For emulation models and real test-bed environments, although they can provide higher validity of the performance of MTD techniques, the researchers found that most of them are having difficulty evaluating large-scale networks. In terms of the simulation models, although the results of the performance evaluation of MTD techniques have less validity than emulation models or real test-bed environments due to the uncontrollable uncertainty issue, the researchers pointed out that it can provide high flexibility in modeling attacks, system components, network scales, and MTD techniques without much restriction, which means it will be a good choice for the proposed project.

MTDSim is a Python-based MTD simulator first designed and developed by Brown [19]. The purpose of this simulator is to evaluate the effectiveness of combinations of multiple MTD techniques. However, due to time constraints, only shuffle-based MTD techniques were implemented in his work. Based on that, Lee implemented a new attack profile and diversity-based MTD techniques with additional security metrics to improve the overall functionalities and performance of the simulator [20]. However, the current simulator still falls short in simulating 'time'. Firstly, in terms of modeling the time-consuming actions of MTD, *MTDSim* only modeled time in the *MTD interval* (Section II-A2) and ignores the execution time of MTD. This means whenever an MTD is triggered, it will be instantly executed, which can be a problem as all types of MTD operations need a certain amount of time to be deployed and executed [21] [22]. On the contrary, the attacker's movements in the simulator were designed relatively comprehensive. The attacker's actions were adaptively changed based on a given amount of simulation time and certain events occurred. However, due to the lack of consideration in the time delay of each MTD event and the design of a flawless attack profile against security strategies, the overall experiment results of MTD were still not so desirable. Therefore, in order to further improve the performance of the simulator and obtain more constructive outcomes from it, the time domain should be integrated into the current simulator.

IV. METHODOLOGY

To extend the simulator in the time domain, discrete events that occurred in the system need to be modeled and simulated appropriately. This involves identifying the potential events that will happen in the system, the relations and interactions of each event with other events, and the time duration of each time-consuming action in each event. Accordingly, Section IV-A will introduce the action flow of both the MTD event and the attack event. Section IV-B will illustrate the solution for simulating the time duration of each time-consuming action in each event and the evaluation methods for assessing the effectiveness of multiple MTD techniques in the time domain.

A. Discrete Event Simulation for MTDSim

An instantaneous occurrence that could alter the status of a system is known as an event [31]. Two main event types are existing in the simulation environment that can be modeled: MTD events and attack events. The MTD operations are deployed either in the application layer (AL) or the network layer (NL). These two layers are treated as two different types

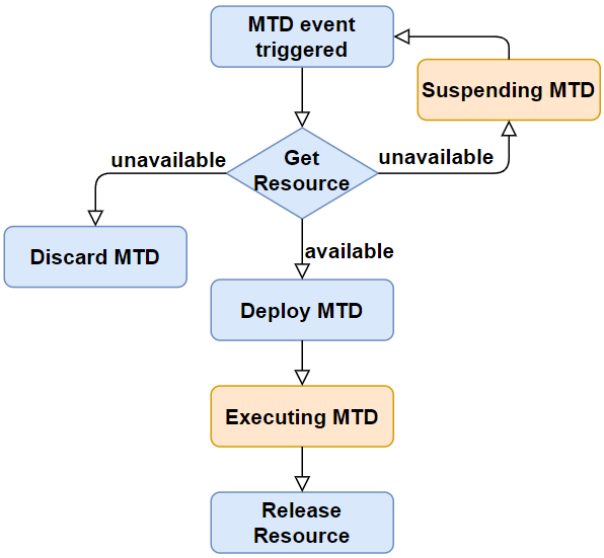

Fig. 1: Action flow diagram of an MTD event

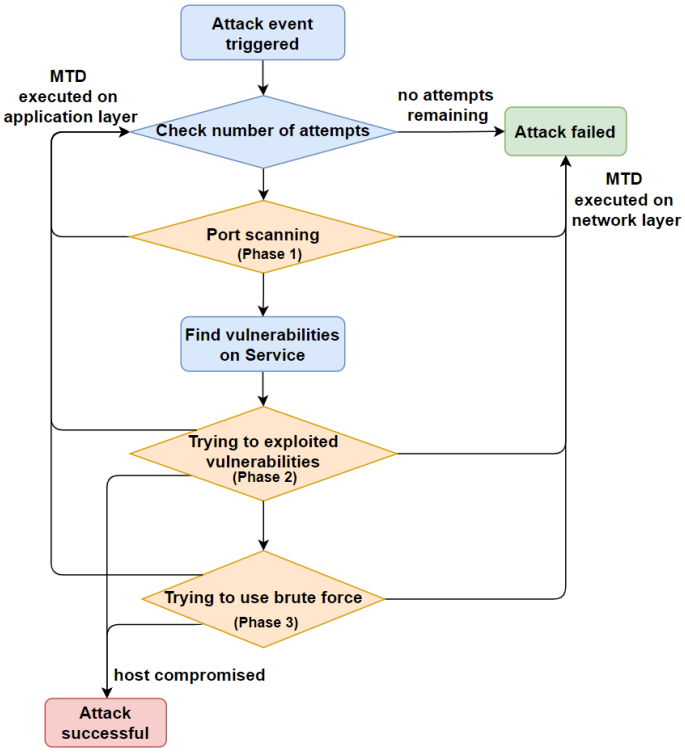

Fig. 2: Action flow diagram of an attack event

of resources in the system. In addition, the proposed attack profile also focuses on these two layers. For adversaries that attack a system following the processes of Cyber Kill Chain (Section II-B1), they will perform the Reconnaissance step on the NL, and perform Exploitation step on the AL.

1) MTD Event Simulation: Fig. 1 is the action flow diagram of an MTD event. An MTD event is triggered periodically with a certain time interval. After the MTD operation is triggered, it needs to get the resource it serves. As multiple MTD techniques may serve the same resource in the system, the first thing for the triggered MTD operations to do is to identify whether the resource they serve is available or not. If the resource is available, the triggered MTD operation will be deployed and occupy that resource until its execution completes. If the resource is not available, the triggered MTD operation will be either suspended until the resource is available or discarded.

For instance, there are three shuffle-based MTD techniques: Port Shuffle, OS Shuffle, and IP Shuffle. The first two are designed to be deployed on the AL while IP Shuffle is deployed on the NL. If OS Shuffle is triggered first and no other MTD operations occupy the AL, then OS Shuffle will be deployed and occupy the AL. However, if Port Shuffle is triggered before OS Shuffle finished its deployment, Port Shuffle will be suspended or discarded since the AL is still occupied by OS Shuffle, and vice versa. However, IP Shuffle can be deployed at any time in this case as it is deployed on the NL.

2) Attack Event Simulation: The overall steps of the attack event are similar to the Cyber Kill Chain [16]. Fig. 2 is the action flow diagram of an attack event. An attack event is triggered after the adversary scans the network and discovers accessible hosts. There are three time-consuming phases in an attack event. Firstly, the attacker needs to perform port scanning on the host to look for services that have vulnerabilities that can be exploited. This action is denoted as Phase 1. After locating vulnerabilities in the service, the attacker starts trying to exploit vulnerabilities, which is denoted as Phase 2. If the vulnerabilities can be successfully exploited by the attacker, this attack event will be successful. If the vulnerabilities cannot be exploited, the attacker will start trying to brute force a user login, which is denoted as Phase 3. If the brute force is successful, this attack event will also be successful.

However, different MTD techniques can either interrupt or terminate the attack event in any of these three attack phases. Firstly, the attack event will be failed immediately if any MTD techniques are deployed on the NL. It is because the attack event is triggered after the attacker discovered a host, NL-based MTD strategies such as IP Shuffle and Host Topology Shuffle can block the attacker's action by changing the IP address or the connection path of each host. As a result, the adversary loses connection to the target host and the attack event thus fails. On the other hand, AL-based MTD strategies such as Port Shuffle and OS Shuffle cannot block attack actions as they do not affect the overall network topology. However, attack actions will still be interrupted as services on the target host are deployed on another port or reconfigured. Hence, the adversaries at this condition have to restart the attack event from Phase 1 whatever their progress was before. In addition, the limited number of attempts will be set for each attack event. This means if the number of attack actions of the adversary being interrupted by AL-based MTD reached a certain threshold value, this attack event will fail and the adversary will shift the target to attack other hosts in the network.

B. Time Simulation and Evaluation Methods

In Discrete Event Simulation (DES), moving the simulated time from one value to the next is one of the key elements [31]. Thus, simulating time for each time-consuming action in each event is an important step for integrating the time domain to *MTDSim*.

1) Generating Probabilistic Simulation Time Value: Since the real-world system have a high level of randomness, it is crucial for the simulation study to use notions from probability and statistics. Finding the probability distribution functions that describe the input data is important in this regard. Accordingly, the exponential distribution is selected to be the main probability distribution method as it is commonly used in simulating the distribution of elapsed times of an action [32]. The probability density function (pdf) is shown in Eq. 1.

$$f(x) = 1/\mu \mathrm{e}^{-\frac{1}{\mu}x} \tag{1}$$

Generally, the random variable *x* equals (a) the time between two successive events, such as the *MTD interval*, or (b) the passage of time to complete an action. μ is the historical average elapsed time, which will be obtained from empirical study and sensitivity analysis. Based on the above probability density function, the corresponding cumulative density function (cdf) is shown in Eq. 2.

$$F(x) = \int_0^\infty \left[\frac{1}{\mu} e^{-\frac{x}{\mu}}\right] = 1 - e^{-\frac{x}{\mu}} \tag{2}$$

For instance, With the cdf, the probability that a specific time-consuming action finishes execution before time $x = a$ will be set to be $P(x < a) = 1 - e^{-\frac{a}{\mu}}$. Based on that, the Probabilistic value of the simulation time of each time-consuming action can be generated.

2) Time Simulation for Attacker: As it is shown in Fig. 2, three time-consuming phases coloured in orange need to be considered in an attack event. The time consumption of Phase 1 and Phase 3 are assumed to be no external interference. For Phase 1, the speed of the port scanning is usually constant based on the same scan strategy while the brute force is set to have a certain time limit in Phase 3. The time needed for Phase 2, however, varies based on several factors, which are either on the server side or the attacker side.

In terms of the server side, the time duration of Phase 2 can be affected by the attack complexity (AC_v) of the service running on the target host. In *MTDSim*, the services are randomly generated with a version number from 1 to 99 on each host. The AC_v would be produced at random from a range of [0,1] to reflect the likelihood that the attack would succeed, with 0 denoting the unexploitable vulnerability and 1 denoting the easiest exploitable vulnerability. Since the service is simulated, time duration data from empirical observation is not applicable. Alternatively, sensitivity analysis will be used to assign different values of time duration for different levels of AC_v to get a reasonable impact of each level of AC_v will affect the time duration of the attack.

For the attacker side, Phase 2 can be affected by the learning capabilities and confusion of the adversary. The adversary in *MTDSim* is designed to take a time penalty to simulate the confusion caused by MTD. The time penalty will be assigned each time the attack event is interrupted or stopped by MTD. However, each time the adversaries will learn from the changes that hinder their actions and thus decrease the time penalty next time. The range of reduction of the time penalty will be based on the learning capabilities of the attacker. So in this case, sensitivity analysis will also be used to determine the proper learning rate of the attacker.

The mean time to compromise (MTTC) will be used as the evaluation method to assess the effectiveness of MTD from the attacker's side. The time duration for compromising a host is the total time spent for a successful attack event, which is calculated by summing up the time spent by all time-consuming actions in this event. For instance, if the attacker successfully compromises a host in the first attempt of Phase 2 of an attack event. The total time to compromise the host will be time duration in port scanning plus time duration in exploiting vulnerabilities ($T_{A_{total}} = T_{A_{phase1}} + T_{A_{phase2}}$). This metric can reflect the effort spent by the attacker to compromise the host, thus indicating the effectiveness of MTD.

3) Time Simulation for MTD: As mentioned in Section IV-A1, the MTD operations were deployed periodically in a certain amount of time interval in the *MTDSim*. The time interval between the triggering of two MTD operations (*MTD interval*) was designed to be uniformly distributed. While uniform distribution is often used when the characteristics of the random variables are not well known, the new proposed approach will simulate the *MTD interval* using exponential distribution and explore other types of distribution if necessary in order to find a more appropriate distribution to perform simulation.

When modeling the combination of multiple MTD techniques in the time domain, it is necessary to take the difference in time spent on executing different MTD techniques into consideration as different MTD techniques will take a different amount of time to be executed in the real-world scenario. However, there is still a lack of research that measures the time spent by specific MTD techniques. For instance, Chen et al. [21] set a default value of the time duration of executing MTD techniques to their testbed environment, but the MTD technique they used was generic. Due to that, the proposed solution will pick various time values for each MTD technique in a reasonable range based on the existing empirical data. Then perform sensitivity analysis to pick up the appropriate value.

The frequency of MTD deployment (FMD) will be used to evaluate the effectiveness of MTD from the defender side. It is calculated by recording the number of occurrences of any executed MTD events (N) in a given amount of time (T), which means the discarded MTD will not be taken into consideration in this case. For instance, in the time range T, the FMD will thus be $FMD = N/T$. Since deploying MTD can take time and computation resources, FMD will imply the cost from the defender side. In this case, By analysing the result of combining FMD with MTTC, a reasonable indication of trading-off between cost and security can be obtained.

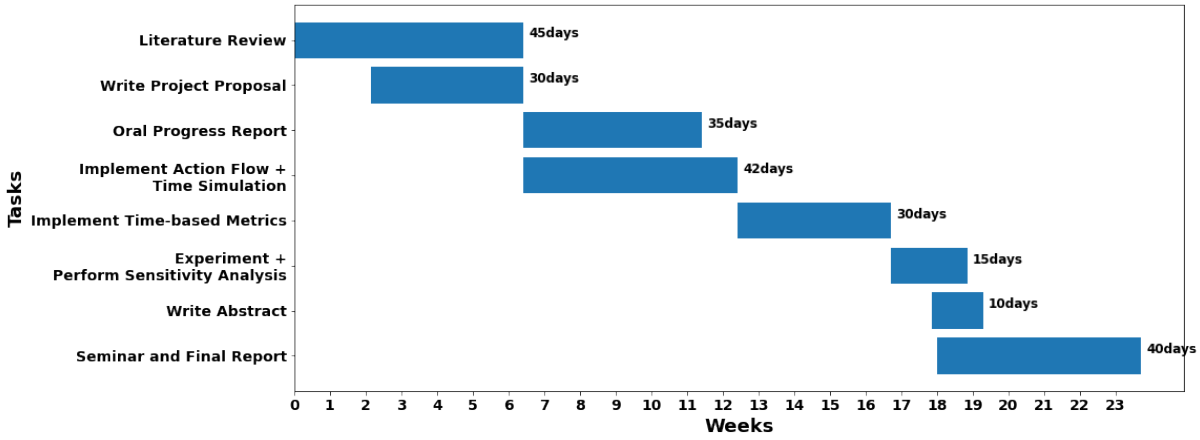

Fig. 3: Timeline

V. TIMELINE

As it is shown in Fig. 3, several tasks need to be done in order to complete the project. These tasks can be allocated into four phases. The first phase is to review the related work in literature and write the project proposal to gain a brief overview of the project and acquire the necessary knowledge to deal with it. The subsequent phase is the implementation phase, which consists of implementing the simulation model and the evaluation method in the *MTDSim*. Implementations of the simulation model include action flow and time simulation for both attack events and MTD events. Implementations of the evaluation method are focused on time-based metrics. In addition, an oral progress report will be delivered near the end of this phase. The third phase is to evaluate the implemented MTD model by performing sensitivity analysis and experiments on it. The final phase is to complete the abstract, final report, and seminar about the project that had been conducted.

REFERENCES

[1] C. Ruhl, D. Hollis, W. Hoffman, and T. Maurer, "Cyberspace and geopolitics: Assessing global cybersecurity norm processes at a crossroads." Carnegie Endowment for International Peace., 2020.

[2] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, Moving target defense: creating asymmetric uncertainty for cyber threats. Springer Science and Business Media, 2011.

[3] A. Alshamrani, S. Myneni, A. Chowdhary and D. Huang, "A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities," in IEEE Communications Surveys and Tutorials, vol. 21, no. 2, pp. 1851-1877, Secondquarter 2019, doi: 10.1109/COMST.2019.2891891.

[4] J. -H. Cho et al., "Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense," in IEEE Communications Surveys and Tutorials, vol. 22, no. 1, pp. 709-745, Firstquarter 2020, doi: 10.1109/COMST.2019.2963791.

[5] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang and S. Kambhampati, "A Survey of Moving Target Defenses for Network Security," in IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 1909-1941, thirdquarter 2020, doi: 10.1109/COMST.2020.2982955.

[6] G.-L. Cai, B.-S. Wang, W. Hu, and T.-Z. Wang, "Moving target defense: State of the art and characteristics," Front. Inf. Technol. Electron. Eng., vol. 17, no. 11, pp. 1122–1153, Nov. 2016

[7] J. Hong and D. S. Kim, "HARMs: Hierarchical attack representation models for network security analysis," Proceedings of the 10th Australian Information Security Management Conference, AISM 2012, pp.74–81, 2012.

[8] 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 International Conference on Information Security Practice and Experience, 2017: Springer, pp. 539-548.

[9] H. Alavizadeh, J. B. Hong, D. S. Kim, and J. Jang-Jaccard, "Evaluating the effectiveness of shuffle and redundancy mtd techniques in the cloud," Computers and Security, vol. 102, p. 102091, 2021.

[10] H. Alavizadeh, J. B. Hong, J. Jang-Jaccard, and D. S. Kim, "Evaluation for combination of shuffle and diversity on moving target defense strategy for cloud computing," in Proc. 17th IEEE Int. Conf. Trust Security Privacy Comput. Commun. (TrustCom), 2018, pp. 573–578.

[11] H. Alavizadeh, D. S. Kim, and J. Jang-Jaccard, "Model-based evaluation of combinations of shuffle and diversity MTD techniques on the cloud," Future Generation Computer Systems, vol. 111, pp. 507-522, 2020.

[12] A. Laszka, W. Abbas, Y. Vorobeychik and X. Koutsoukos, "Synergistic Security for the Industrial Internet of Things: Integrating Redundancy, Diversity, and Hardening," 2018 IEEE International Conference on Industrial Internet (ICII), 2018, pp. 153-158, doi: 10.1109/ICII.2018.00025.

[13] M. C. Lucas-Estañ, B. Coll-Perales and J. Gozalvez, "Redundancy and Diversity in Wireless Networks to Support Mobile Industrial Applications in Industry 4.0," in IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 311-320, Jan. 2021, doi: 10.1109/TII.2020.2979759.

[14] H. Alavizadeh, J. B. Hong, J. Jang-Jaccard, and D. S. Kim, "Comprehensive security assessment of combined MTD techniques for the cloud," in Proc. 5th ACM Workshop Moving Target Defense (MTD), 2018, pp. 11–20

[15] "Discrete-event simulation - Wikipedia", En.wikipedia.org, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Discrete-event_simulation. [Accessed: 14- Aug- 2022]

[16] L. Martin. (2017). Cyber Kill Chain (CKC). URL: https://www.lockheedmartin.com/enus/capabilities/cyber/cyber-kill-chain.html

[17] MITRE ATT&CK Framework: Adversary Tactics and Techniques. 2015. URL: https://attack.mitre.org/

[18] J. Livingston. (2022). "What is MITRE ATT&CK? The Definitive Guide." 2022. URL: https://verveindustrial.com/resources/blog/what-is-mitre-attack-framework/

[19] A. Brown (2021) MTDSim[Source Code] https://github.com/Ccamm/MTDSim

[20] T. Lee (2022) MTDSimTze[Source Code] https://github.com/tzewenlee99/MTDSimTze

[21] Z. Chen, X. Chang, J. Mišić, V. B. Mišić, Y. Yang and Z. Han, "Model-based Performance Evaluation of a Moving Target Defense System," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9322609.

[22] W. Connell, D. A. Menascé, and M. Albanese, "Performance modeling of moving target defenses," in Proceedings of the 2017 Workshop on Moving Target Defense, 2017, pp. 53-63.

[23] W. Connell, D. A. Menascé and M. Albanese, "Performance Modeling of Moving Target Defenses with Reconfiguration Limits," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 1, pp. 205-219, 1 Jan.-Feb. 2021, doi: 10.1109/TDSC.2018.2882825.

[24] SimPy (2002-2020). SimPy: Simulation framework in Python. URL: https://simpy.readthedocs.io/en/latest/index.html

[25] M. Torquato, P. Maciel and M. Vieira, "PyMTDEvaluator: A Tool for Time-Based Moving Target Defense Evaluation: Tool description paper," 2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE), 2021, pp. 357-366, doi: 10.1109/ISSRE52982.2021.00045.

[26] X. Xiong, L. Ma, and C. Cui, "Simulation Environment of Evaluation and Optimization for Moving Target Defense: A SimPy Approach," in Proceedings of the 2019 the 9th International Conference on Communication and Network Security, 2019, pp. 114-117.

[27] J. B. Hong, S. Y. Enoch, D. S. Kim, A. Nhlabatsi, N. Fetais, and K. M. Khan, "Dynamic security metrics for measuring the effectiveness of moving target defense techniques," Computers & Security, vol. 79, pp. 33-52, 2018.

[28] J. Zheng and A. Siami Namin, "A markov decision process to determine optimal policies in moving target," in Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, 2018, pp. 2321-2323.

[29] C. L. Belusso, S. Sawicki, F. Roos-Frantz, and R. Z. Frantz, "A study of Petri Nets, Markov chains and queueing theory as mathematical modeling languages aiming at the simulation of enterprise application integration solutions: a first step," Procedia Computer Science, vol. 100, pp. 229-236, 2016.

[30] G. Cai, B. Wang, Y. Luo, and W. Hu, "A model for evaluating and comparing moving target defense techniques based on generalized stochastic Petri net," in Conference on Advanced Computer Architecture, 2016: Springer, pp. 184-197.

[31] D. Weitz, "Introduction to Simulation with SimPy Part 1: The Basics", Towards Data Science. [Online]. Available: https://towardsdatascience.com/introduction-to-simulation-with-simpy-b04c2ddf1900. [Accessed: 22- Aug- 2022].

[32] D. Weitz, "Introduction to Simulation with SimPy Part 2: Measures of Performance for Queuing Systems", Towards Data Science. [Online]. Available: https://towardsdatascience.com/introduction-to-simulation-with-simpy-322606d4ba0c. [Accessed: 22- Aug- 2022].