IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 18, 2018, accepted January 4, 2019, date of publication January 9, 2019, date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891613

Effectiveness Evaluation Model of Moving Target Defense Based on System Attack Surface

XIN-LI XIONG[1], LIN YANG[2], AND GUANG-SHENG ZHAO[3]
[1]Collage of Command and Control Engineering, Army Engineering University of PLA, Nanjing 211101, China
[2]System Engineering Research Institute, Academy of Military Sciences PLA, Beijing 100141, China
[3]Collage of Computer Science, National University of Defense Technology, Changsha 410073, China

Corresponding author: Lin Yang (yanglin61s@yahoo.com.cn)

ABSTRACT Evaluation of moving target defense (MTD) effectiveness has become one of the fundamental problems in current studies. In this paper, an evaluation model of MTD effectiveness based on system attack surface (SAS) is proposed to extend this model covering enterprise-class topology and multi-layered moving target (MT) techniques. The model is focused on the problem of incorrect performance assessment caused by inaccurately characterizing the process of attacking and defending. Existing evaluation models often fail to describe MTD dynamically in a process. To deal with this static view, offensive and defensive process based on a player's move is presented. Besides, it converts all the attack and defense actions into the process, and interactivities are evaluated by system view extended attack surface model. Previously, the proposed attack surface models are not concerned about the links between nodes and vulnerabilities affected by topologies. After comprehensively analyzing the impact of interactions in the system, a SAS model is proposed to demonstrate how resources of the system are affected by the actions of attackers and defenders, thus ensuring the correctness of parameters for SAS in measuring MT technology. Moreover, by generating a sequence of those shifting parameters, a nonhomogeneous hierarchical hidden Markov model is used to find the possible sequence of attacking states by introducing the partial Viterbi algorithm. Also, a sequence of attacking states is defined to illustrate how adversaries are handled by MT technologies and how much additional consumption costs are increased by the system resource reconfiguration. Finally, the simulation of the proposed approach is given in a case study to demonstrate the feasibility and validity of the proposed effectiveness evaluation model in a systematic and dynamic view.

INDEX TERMS Information security, moving target defense, nonhomogeneous hidden Markov processes, performance evaluation.

I. INTRODUCTION

The development of computer science significantly extends the application of information system in our daily life. Security problems in cyber-space affect not only personal privacy but also political, economic, and military fields. Despite meaningful progress in cyber-security technologies, information systems are often challenged by attacks such as zero-day exploitation and advanced persistent threats (APT), which fall into a severe predicament of "easy to attack and hard to defend." [1]. Apart from inevitable vulnerabilities in software, deterministic and static nature of system architecture enables adversaries with enough time to launch detections and attacks. Moreover, the isomorphism of system architecture results in a successful implementation of malware that can easily sweep a large scale of information systems at a relatively low cost. At the same time, Deep Webs and black products facilitate hackers with different backgrounds and objectives to craft more intelligent and automated tools [2]. Existing defense methods cannot cope with these tools, which makes the deterioration of asymmetry in adversaries and defenders more and more serious.

To reverse the offense-defense state in cyber-space, a game challenging method, moving target defense (MTD) is proposed to change the attributes of systems uncertainly for adversaries, and make the protected system more random, dynamic and heterogeneous. Moving target techniques interrupt cyber kill chain (CKC) by changing configuration, topology, running environment and data format of system. Despite numerous moving target (MT) technologies proposed by researchers and engineers, only a few methods are employed practically in a wide range [3]. In addition to performance cost brought by MTD, lack of effectiveness assessment is another critical drawback. Therefore, how to evaluate diverse defense strategies employed by multi-layered MT techniques to achieve quantifiable results in a reasonable system topology has become the most fundamental and urgent question in current MTD studies.

Attack surface (AS) is a set of ways in which an adversary can enter the system and potentially cause damage [4]. The goal of AS models is to establish an assessment model to analyze the security level of the system, especially in the operating system combined with various softwares. Besides, confronting with MTD environment, attack surface generated by attacker and defender are different in illustrating how MT technologies can manipulate system resource and make attacking knowledge invalid frequently. With those differences, diverse defense strategies can be demonstrated by attack surface shifting, resizing and remodeling. Combined with AS and CKC, processes between adversaries and defenders can be analyzed by well-defined parameters and quantifiable results can be calculated by estimating the transition of attacking states.

Motivated by existing studies about evaluating framework and models, an MTD effectiveness assessment model focused on the offensive and defensive process is presented in this paper. This model expands the attack surface model to a system view and adopts nonhomogeneous hierarchical hidden Markov model (NHHMM) to solve the shortcomings in current studies. The main contributions of this paper are as follows:

(1) Regarding model construction, AS based on software is extended to a system view including nodes and links in system topology, which ensures the model universality. Furthermore, the evaluation focuses on offensive-defensive processes by an introduced sequence of shifting parameters rather than analyzing parameters statically, which makes the constructed model more in accordance with the features of attack-defense in MTD.

(2) In terms of attacking state analysis, judgments of transition based on prior and matching conditions are combined with the probability transition matrix in the hidden Markov model (HMM). At the same time, by taking the hierarchical model and nonhomogeneous features into account, HMM is extended to NHHMM. This extension enables getting more accurate conclusions of the transition of attacking states and making the assessment model more practical. Based on it, a partial Viterbi algorithm is proposed to enhance the model availability in solving estimation of the attacking state sequence in NHHMM.

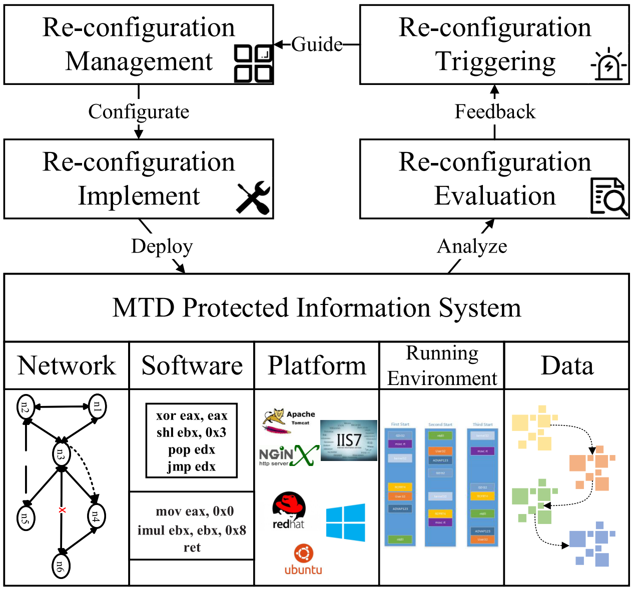
This paper is organized as follows: In Section II, the basic principles of MTD and attack surface are presented from the view of attack surface shifting. The related concepts of NHHMM and existing work are given. Section III analyzes the process between adversaries and defenders, and a time conversion model for the sequence of attacking states is given. After that, the system view of attack surface is given in Section IV and parameters of attack surface shifting are defined. In section V, the NHHMM for effectiveness evaluation is constructed and a partial Viterbi algorithm is designed. In Section VI, a case study illustrates the effectiveness of the proposed model and discusses quantification results under diverse MT strategies. Finally, we conclude our work and give future studies directions in Section VII.

II. BACKGROUND AND RELATED WORK

A. PRINCIPLES OF MOVING TARGET DEFENSE AND ATTACK SURFACE MODEL

The MTD is defined as ‘‘It enables to create, analyze, evaluate, and deploy mechanisms and strategies that are diverse and that continually shift and change over time to increase complexity and cost for attackers, limit the exposure of vulnerabilities and opportunities to attack, and increase system resiliency’’ [3]. The basic principle of MTD is shown in FIGURE.1. The MT techniques keep changing system resources by re-configurations in network, software, platform, running environment and data. All those changes in system resources are made by implementation of re-configuration, which is deployed in various layers of the information system. Moreover, those enhancements in security are analyzed by evaluation of re-configuration to give feedback to the triggering mechanism. Combined with timely feedbacks and a variety of shifting strategies, the management of MTD configures parameters of implementations in the system. Through implementations of MTD, the AS of each node continuously deforms, and the links between nodes are changeable as well. As a result, the MTD can deceive and confuse reconnaissance, interrupt the ongoing attacks and force attackers to repeat accomplished steps, due to which the efforts of attackers to successfully attain objectives will be remarkably increased.

Attack surface was first introduced by Howard *et al.* [5] to measure how likely the Windows operating system is vulnerable to attack based on the degree of exposure. In addition, their work [6] shows that an AS is promising while comparing two systems, such as Linux and Windows, regarding their security. In follow-up work, Manadhata and Wing [7] established a generalized formal notion of software AS based on system's entry point and exit point framework, which in turn identifies the relevant resources that contribute to the vulnerability exposure. Wang *et al.* [8] advanced the notion of AS to the complete network of computer systems, which focuses on the interfaces like remotely exploitable services. Sun and Jajodia [9] defined systems AS as a set of ways by which an adversary can enter into the system and compromise its security, which their work expended attack surface to external views from adversaries and internal views from defenders. Albanese *et al.* [10] formalized the notion of system view as well as the notion of distance between the views in attack surface, which can be thought as the subset of the internal view that would be exposed to potential attackers when no deceptive strategy is adopted.


Re-configuration Management
Guide
Re-configuration Triggering
Configurate
Feedback
Re-configuration Implement
Re-configuration Evaluation
Deploy
Analyze
MTD Protected Information System
Network
Software
Platform
Running Environment
Data
Apache Tomcat
IIS7
NGINX http server
redhat
ubuntu

```
xor eax, eax
shl ebx, 0x3
pop edx
jmp edx
```

```
mov eax, 0x0
imul ebx, ebx, 0x8
ret
```

FIGURE 1. The Basic Principle of MTD.

With the AS, existing works has introduced several models to demonstrate how attackers interacted with the MT techniques. Manadhata [11] extended the AS model with game theoretic approaches to describe the attack surface shifting in the MTD system. The extended model formalized how MT techniques affected the AS and provided a demonstration of interactions between the attacker and defender with a two-player stochastic extensive game. Zhuang *et al.* [12] introduced a theoretical framework that formalized the MTD system and extended the definition for AS. Their work [13] also presented a model of exploration surface to depict the effort that an MT technology imposes on adversaries, which inspired us to develop a framework that contains adversaries, system resources and MT techniques.

Essentially, the frameworks in existing references combined with the formalization of the MTD system and AS model are capable of describing the security model of a single software or a node under consideration. However, our focus is on the AS of an entire information system rather than a single isolated node in this article. Besides, inspired by this theoretical framework, we also present a framework based on a system attack surface (SAS) to describe interactions between players in the MTD system [14]. We inherit the description of external AS and internal AS from our previous framework.

B. EVALUATION MODELS FOR MTD

Although MT technologies are increasingly sophisticated and applied to protect the campus network and enterprise-class intra-net, assessment methods of effectiveness for diverse MT techniques still need to be researched. Therefore, to evaluate the efficiency of MTD more comprehensively, intuitively and accurately is a hot topic of current studies [3]. To analyze randomness, dynamism and uncertainty of MTD, various mathematical models including probabilistic model, State Machine, Markov model, Game theory, Attack Graph model, Testbed based simulation/emulation model and hybrid model are applied by various researchers.

Urn model [15] and Balls into Bins model [16] are classical probabilistic models for analyzing the randomness of network-based MT technologies, such as IP address shuffling and port hopping. Carroll *et al.* [17] established an evaluation model for the effectiveness of network address shuffling under two extreme conditions, perfect hopping and static none-defense. This model can analyze the influence of the network size, the number of vulnerable systems, scanning frequency, and number against the success probability of attacks. Crouse *et al.* [18] extended the Urn model to include deception methods and analyzed the success probability of attacks in the none-defense model, honey-pot model, and MTD model. Luo *et al.* [19] illustrated the effectiveness of port hopping under extreme scenarios, perfect and static, which quantified the influence of different port pool size, scanners, vulnerable services, and port hopping frequency against the success probability of attacks. Evans *et al.* [20] evaluated various diversity defenses against different attack strategies, and a Ball into Bins model was employed to assess incremental attacks in re-randomization of XOR keys for instruction-set randomization. Okhravi et al. [21] developed a generalized model of dynamic platforms and deduced a probability model to predict the expected time required for an attacker to compromise a system that is protected by MTDs.

The state machine is a widely used and accepted computational model in computer science. Describing the change happening in a system when attacks and MTDs occur, the ability of state machine can help evaluate the system status. Xu *et al.* [22] proposed a three-layered evaluation method that filled the gap between low-level and high-level methods, such as ASLR, ISR, and software diversification. The first layer captures low-level contexts in separate programs; the second layer models damage propagation between different programs; the third layer works as a user interface to explicitly expresses evaluation results.

Markov model can illustrate the process of system reconfiguration under MTD. Maleki *et al.* [23] proposed a Markov chain model to analyze the effectiveness of various MT technologies and strategies. By introducing the security capability that converts the attack time into its cost, a general model for the probability of a successful attack and the total attacking time and cost is given under single and multi MTD environments. Nguyen and Sood [24] used Semi-Markov Chains to analyze the resilience of services protected with Self-Cleansing Intrusion Tolerance mechanism, where states capture the behaviors of both the attacker and the service being studied. By analyzing mean time to security failure (MTTSF) and mean time to recover (MTTR) of services, this model provided a mathematical foundation for compensating the expansion of a service's attack surface by tuning SCIT system parameters.

Game theory is used to analyze the relationship between attackers and defenders. Parakash and Wellman [25] defined an abstract network attack and defense scenario and analyzed diverse attacking and defensive strategies through empirical game theory. Through analyses of 72 examples of games defined by different targets, costs and defenders' ability, a group of strategies with distinct differences under different environmental conditions was obtained, and the MTD defense strategy was evaluated quantitatively. Jones *et al.* [26] proposed a game theory model containing probabilistic learning attacker and dynamic defender (PLADD) to evaluate MT strategies, which proves that a defensive strategy causes the rational attacker to withdraw from the attack explaining the limitations of attacker's strategy in PLADD. Ben-Asher *et al.* [27] introduced a quantitative evaluation model of the effectiveness of MT technologies aiming at platform migration defense (PMD) technology. Through simulation experiments, the effectiveness of different platform migration strategies and the impact of different skill levels of attackers are evaluated. Moreover, Zhu and Başar [28] developed a game-theoretic framework for guiding the quantitative design of MTD as a trade-off between security and usability. Based on the model, the authors proposed a feedback mechanism that allows the system to monitor its current system state and update its randomized strategy based on its observation.

Attack graph (AG) model is used to describe the potential attack path by graphically presenting vulnerability dependency and state transition relationship. Lei *et al.* [29] aimed at the network-based MT techniques and presented a change-point based assessment method containing hierarchical network resource graph to establish the relationship between resource vulnerability change and node security state transition. Hamlet and Lamb [30] extended AG to resource dependency graph and employed this model to evaluate the effectiveness of address space layout randomization and data execution prevention.

Simulation and emulation methods are intuitive in evaluation and analysis, which provide a uniform approach to evaluate MTD techniques with limited resources and efforts. Zhuang *et al.* [31] presented a preliminary design of a network moving-target defense system and conducted simulation-based experiments on an existing network security simulator called NeSSi2 to study the effects of randomly changing one aspect of the system in reducing attacker's success likelihood. Bardas *et al.* [32] analyzes the costs and security benefits of MTD in cloud based IT systems (CBITS) using a practical attack window model and show how a system managed using MTD CBITS will increase attack difficulty.

Hybrid model combines several evaluating methods to provide a more comprehensive view. Zaffarano *et al.* [33] presented an overall approach to metric design for moving target defense technology for network defense, which utilized a cyber-testbed Siege's cyber quantification framework that can be rapidly configured and reconfigured to gather and analyze vast quantities of data. Taylor *et al.* [34] described the results of several experiments designed to test two dynamic network moving target defenses against a propagating data ex-filtration attack. The authors designed a collection of metrics to assess the costs of mission activities and benefits in the face of attacks and evaluated the impacts of the moving target defenses in both areas.

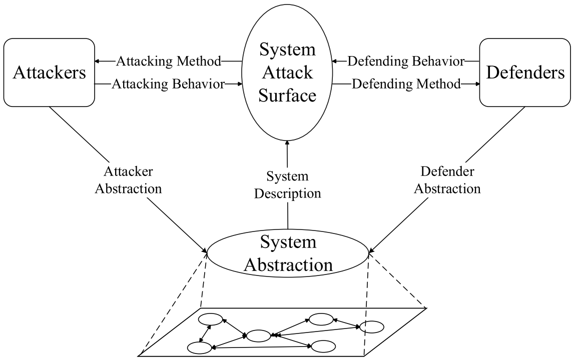


FIGURE 2. The Framework for Evaluating the Effectiveness of MTD.

From the above discussion, we can conclude that to evaluate the effectiveness of multi-layered MTD in a large-scale network topology, an assessment model for integrating multiple methods is indispensable. Although the evaluation of effectiveness for MTD has made certain advances, some problems still exist that are as follows:

(1) In terms of applicability, most models are either highly abstracted or fully simulated/emulated. Selections of parameters to depict MTD in the existing literature do not correspond to the strategy in real systems. Moreover, metrics used in evaluations of the models to demonstrate the security level are counterintuitive and improper in the MTD system, which may lead to misleading results and apply to MTD trade-off.

(2) In terms of extensibility, most models mainly focus on single-layered MT techniques. These models cannot be extended to multiple MTD combined systems, which leads to defects in assessing multi-layered MTDs.

(3) In terms of comprehensiveness, a few of the existing models evaluate the effectiveness in a system view. This drawback leads to a lack of analysis for the security of large-scale information systems.

III. OFFENSIVE AND DEFENSIVE PROCESS

To accurately describe characteristics of MTD technologies and evaluate its effectiveness, our approach extends the framework for evaluation in MTD [14], and the framework is shown in FIGURE.2. Moreover, the offensive and defensive process (ODP) is refined in this paper to include more details between adversaries and defenders. Analyses of this process are the key to effectiveness assessment for MTD techniques. The basic model of interactive process is shown in FIGURE.3

As shown in FIGURE2, the framework for evaluating the effectiveness of MTD includes three parts which are system attack surface, players and its behavior, and offensive & defensive processes to give an assessing for MT technologies. This framework using ODP to describe how adversaries and defenders interacted with system resources to assess.

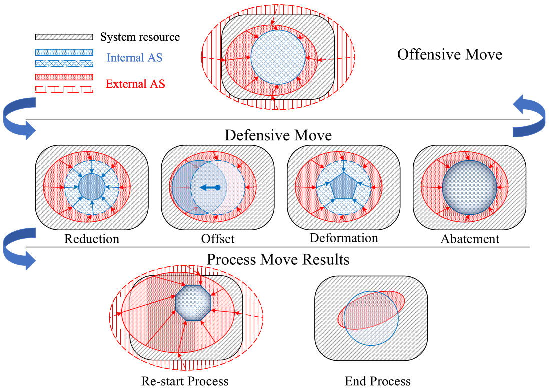


FIGURE 3. The Basic Model of Offensive And Defensive Process.

The ODP starts from offensive moves and ends in the objective accomplished by attackers. Those offensive moves illustrate how adversaries prepare and conduct attacks. From the perspective of AS model, external AS approaches internal AS to acquire information, craft malware and conduct exploitation in those offensive moves. At the same time, defensive moves demonstrate how defenders protect information system by traditional security methods or MTD. In the point of attack surface, those defensive moves change internal AS by reduction, offsetting, deformation and abatement, which frustrate adversaries in different stages. The interaction between offensive and defensive moves leads to two types of results: the defenders winning the game and attackers restarting offensive moves, and the attackers winning the game and end of the whole ODP.

For ODPs under MTD environment, adversaries are prohibited by defender manipulating the system configurations, which results in knowledge acquisition of attackers different from static protection. Therefore, the offensive process is analyzed by the model of offensive knowledge acquirement under MT techniques and the defensive process is illustrated by system resources transition. Moreover, logic-time and real-time conversions in ODP are presented to employ this method in practical situations.

A. ADVERSARIES KNOWLEDGE ACQUISITION MODEL

Protecting by MT techniques, attacking knowledge of targets explored by adversaries about targets are frequently changing. This characteristic makes adversaries adjust their strategies to regain advantages, and this repeated interaction leads to three representative attacking knowledge models (AKM) which are, *waiting for change*, *step by step* and *probabilistic probing*.

Taking *waiting for change* model as an example, adversaries explore target system with constant knowledge, expecting that the MT shifting matches the knowledge. This model is only effective against high-frequency shifting. With *step by step* model, adversaries detect knowledge of target system over a sequence of interactions to exclude inaccurate information. When system resource is shifted by MT techniques, all previous knowledge becomes invalid, and adversaries start detection from the very beginning. This method can be adopted against most strategies of MT techniques, which is more effective against relatively limited space and infrequent shifting. Employing *probabilistic probing* model, adversaries probe real systems with random sequences to eliminate uncertainty, which is effective against relatively large space and frequent shifting.

To cover those three representative models in AKM, adversaries explore the knowledge of targets can be denoted as:

$$K_{att}(t) = \sum_{t=0}^{t}\sum_{d=1}^{N_0} R_{ast}(k_{obj}^{d}(t), t). \tag{1}$$

where $R_{ast}(*, t)$ is the method that adversaries employ to detect targets and acquire knowledge, $k_{obj}^{d}(t)$ is the knowledge of d-th target at time t, N_0 is the number of targets that adversaries can connect with.

To simply the evaluation and cover most scenarios of MTD, only the model of *step by step* is used in AKM to describe how offensive moves make external attack surface approach internal attack surface.

B. CONVERSION IN LOGIC-TIME AND REAL-TIME

Assessment of ODP is based on logic-time[1] to illustrate consumptions of adversaries. The basic logic-time τ stands for a time-step in atomic moves of an attacker or a defender, such as probing of targets or crafting of malware. For single-layer MT techniques, the logic-time and real-time conversion can be done by adding a coefficient γ in a universal time unit (e.g., seconds or milliseconds). However, how do we measure an ODP made of several layers of MTD with different time scales? Let us measure different time steps in layered-moves given as the layered logic-time τ_j at different layers j. To convert those layered logic-time steps to a confluent real-time T_r, a layered coefficient γ_j is defined for each layer.

Assessment of ODP is based on logic-time to illustrate consumptions of adversaries. The basic logic-time τ stands for a time step in atomic moves of an attacker or a defender, such as a probing of targets or crafting of malware. For single layer MT techniques, the logic-time, and real-time conversion can be done by adding a coefficient γ in a universal time unit (e.g., seconds or milliseconds). However, how do we measure an offensive and defensive process made of several layers of MTD with different time scales? Let us measure different time steps in layered moves as the layered logic-time τ_j at different layers j. To convert those layered logic-time steps to a confluent real-time T_r, a layered coefficient γ_j is defined for each layer and the conversion can be denoted as:

$$T_r = \sum_{j=1}^{J} \gamma_j \cdot \tau_j. \tag{2}$$

In this paper, two types of coefficients, γ_{craft} and γ_{other}, are taken into consideration. Intuitively, adversaries use much

[1] Without special notations, time steps used in evaluations of this paper are logic-time steps.

more time with crafting malware than acquiring information or installing a back-door. Based on those distinctions, we set those coefficients to a different magnitude of time, $\gamma_{craft} = 10$ minute per logic-time and $\gamma_{other} = 1$ second per logic-time.

C. BASIC ASSUMPTION FOR EVALUATION

Several assumptions are given to limit the scope of evaluations and simplify the analysis process.

- Both attackers and defenders are rational, which means that they only choose the most advantageous strategies to accomplish their goals.
- Adversaries are aware of changes in system configurations and resources, which means that when SAS shifted by MT techniques adversaries will discard completed attacking steps and re-start from the very beginning.
- Original working and linking states of nodes in the system will not be disturbed by MT shifting, which means that performance cost and state-switching influence are not included in effectiveness evaluation.
- Only MT techniques are deployed in our evaluating environments, which means that adversaries can compromise all nodes with vulnerabilities at the end of ODP.
- Non-MTD changes are prohibited in our evaluation environments to exclude the effect of non-MTD features, which means that only MT techniques can manipulate the system resource and force SAS to shift.
- The ability of adversaries is entirely determined by the attacking knowledge model, which means that the adaptive adversaries are excluded from our evaluation.

IV. PARAMETERS OF SYSTEM ATTACK SURFACE

A. SYSTEM ATTACK SURFACE MODEL

In cyber-space, a typical information system includes servers and users in external and internal networks, and the topology of a typical system is shown in FIGURE.4. These nodes and interconnections make the system resource can be described in a graph model rather than a set of vulnerabilities. Therefore, the system resource can be defined as a directed weighted graph and can be denoted as G_{sys}.

$$G_{sys} =< N_{sys}, E_{sys}, W_{sys} > . \tag{3}$$

where N_{sys} is the set of resource in nodes, E_{sys} is the link between nodes, and W_{sys} is the weight standing for the connection matrix between nodes affected by link status and firewall policies.

This paper defines the system attack surface (SAS) as the abstraction of the system resource, which is a subset of the system resource, and can be denoted as a directed weighted graph $G_{sas}(t)$.

$$G_{sas}(t) =< N_{sas}(t), E_{sas}(t), W_{sas}(t) > . \tag{4}$$

where $N_{sas}(t)$ is the set of AS for internal and external nodes, $E_{sas}(t)$ is the connection between nodes, and $W_{sas}(t)$ is the connection matrix between nodes. Under MT techniques, the SAS is varying from time to time, and t stands for a time variable.

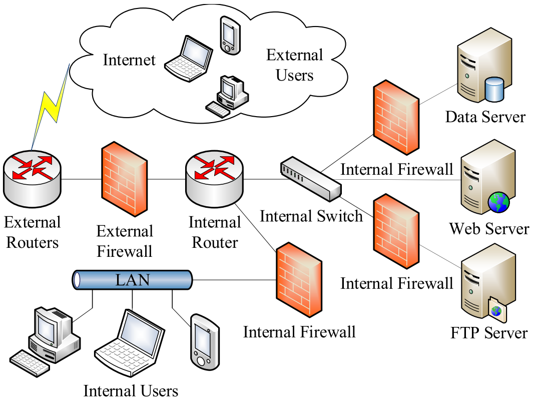


FIGURE 4. **The Typical topology of an information system.**

For the set of node's AS, $N_{sas}(t)$ is the abstraction of node resources, and is defined as a tuple:

$$N_{sas}(t) = \{N^d_{as}(t), d = 1, 2, \cdots, N\}.$$
$$N^d_{as}(t) = < P^t_d, V^t_d > . \tag{5}$$

where P is the attack surface parameters, V is the range of the parameters corresponding to the configuration properties, d is the index of nodes, and N is the total number of nodes in the system.

In this paper, an external system attack surface (ESAS), G_{esas}, is defined as an attack surface abstracted from the perspective of an attacker, and an internal system attack surface (ISAS), G_{isas}, represents an attack surface abstracted from the perspective of the defender. ESAS and ISAS are different in sizes, shapes, and positions indicated different abilities in accessing system resources. Both ESAS and ISAS are specific resources in the system that can be manipulated by attackers or defenders. Moreover, it is impossible for defenders to grasp the zero-day vulnerability in the system thoroughly, and attackers may have the ability to find several zero-day vulnerabilities. Therefore, zero-day vulnerabilities are also a subset of system resource and can be defined as N_{zd}. The relation between N_{zd} to ESAS, ISAS and system resource can be denoted as:

$$N_{zd} = \{n^d_t, \forall n \subset G_{sys}, n \in G_{esas} \cap n \notin G_{isas}\}. \tag{6}$$

B. SHIFTING PARAMETERS OF SAS

With the model of SAS, the AS model is extended to system view, which contains factors of nodes and links. Moreover, Multi-layered MT technologies that manipulate various system resource can be demonstrated by the shifting of SAS.

Definition 1: The shifting of SAS refers to the change of system resources in the process of attack and defense. For a given system, the SAS at time t is $G_{sas}(t)$ and the SAS shifts to $G_{sas}(t + \Delta t)$ at time $t + \Delta t$ by MT techniques. The shifting of SAS must satisfy any one of the following three conditions:

- $\exists r \in G_{sas}(t)$, but $r \notin G_{sas}(t + \Delta t)$

- $\exists r \in G_{sas}(t) \bigcap r \in G_{sas}(t+\Delta t)$, but $r(t+\Delta t) \neq r(t)$
- $\exists r_1, r_2 \in G_{sas}(t) \bigcap r_1, r_2 \in G_{sas}(t+\Delta t)$ and $r_1(t) = r_1(t+\Delta t) \bigcap r_2(t) = r_2(t+\Delta t)$, but $W_{sas}(r_1, r_2, t) \neq W_{sas}(r_1, r_2, t+\Delta t) \bigcup W_{sas}(r_2, r_1, t) \neq W_{sas}(r_2, r_1, t+\Delta t)$

By given the definition of SAS shifting, MT technologies from different layers can be cataloged into various types of SAS shifting. Therefore, the evaluation of multi-layered MT techniques can be converted to evaluate metrics of SAS shifting. Compared to previous studies in AS, shifting parameters of SAS is defined to describe the relative change of SAS rather than give the assessment of security directly.

Taking the SAS and MT techniques into consideration, the shifting parameter of SAS can be quantified by a 5-tuple:

$$M_{sas}(t) =< L(t), \Omega(t), S(t), \Psi(t), \Phi(t) > . \tag{7}$$

where $L(t)$ is the position of SAS, $\Omega(t)$ is the shape of SAS, $S(t)$ is the size of SAS, $\Psi(t)$ is the intensity of SAS and $\Phi(t)$ is connections in SAS.

The position of SAS shifting is a 2-tuple that describes entrances where adversaries intrude into the system and can be divided into two part, network position, and physic position, which can be denoted as:

$$L(t) =< L_{np}(t), L_{pp}(t) > . \tag{8}$$

where system network position $L_{np}(t)$:

$$L_{np}(t) =< L_{sip}(t), L_{spn}(t), L_{spt}(t) > . \tag{9}$$

including node IP address $L_{sip}(t) = \{L^d_{nip}(t), d = 1, 2, \cdots, N\}$, node port number $L_{spn}(t) = \{L^d_{npn}(t), d = 1, 2, \cdots, N\}$ and node protocol type $L_{spt}(t) = \{L^d_{npt}(t), d = 1, 2, \cdots, N\}$. And system physic position $L_{spp}(t) = \{L^d_{npp}(t), d = 1, 2, \cdots, N\}$ is the set of memory address for vulnerabilities.

The shape of SAS shifting describes the set of services provided by nodes that contain potential vulnerabilities. Those services are determined by Os types, platforms and supporting environments, which can be denoted as:

$$\Omega(t) = \{\omega^t_1, \omega^t_2, \cdots, \omega^t_N\}.$$
$$\omega^t_d = \omega^d_{os}(t) \otimes \omega^d_{pt}(t) \otimes \omega^d_{sp}(t). \tag{10}$$

where $\omega^d_{os}(t)$ is the Os type, $\omega^d_{pt}(t)$ is the platform type and $\omega^d_{sp}(t)$ is the supporting environment.

The size of SAS shifting is a 3-tuple that depicts the number of potential vulnerabilities, the difficulty of exploitation and the existence of *zero-day* vulnerabilities, which can be denoted as:

$$S(t) =< S_n(t), S_i(t), S_{zd}(t) > . \tag{11}$$

where $S_n(t)$ is the amount of system potential vulnerabilities and each $S_n(t)$ contains the set of vulnerabilities in nodes, $S^d_{nn}(t)$, that can be denoted as $S_n(t) = \sum_{d=1}^{N} S^d_{nn}(t) \cdot W_{sas}(n^d, *, t)$, and $W_{sas}(n^d, *, t)$ is the weight[2] between node n^d to other nodes at time t. Moreover, the difficulty of exploitation $S_i(t)$ and the existence of zero-day vulnerabilities $S_{zd}(t)$ can be denoted as $S_i(t) = \sum_{d=1}^{N} S^d_{ni}(t) \cdot W_{sas}(n^d, *, t)$ and $S_{zd}(t) = \sum_{d=1}^{N} S^d_{nzd}(t) \cdot W_{sas}(n^d, *, t)$.

The intensity of SAS shifting contains the potential damage of vulnerabilities, and the minimal privilege to cause the damage, which can be denoted as:

$$\Psi(t) =< \Psi_e(t), \Psi_p(t) > . \tag{12}$$

where $\Psi_e(t)$ is the system potential damage and each $\Psi_e(t)$ includes node potential damage $\Psi_{ne}(t)$ that can be denoted as $\Psi_e(t) = \sum_{d=1}^{N} \Psi^d_{ne}(t) \cdot W_{sas}(n^d, *, t)$. And $\Psi_p(t)$ is the system minimal privilege to cause damages, which can be denoted as $\Psi_p(t) = \min[\Psi^d_{np}(t) \cdot W_{sas}(n^d, *, t), d = 1, 2, \cdots, N_{min}]$ and N_{min} is the minimal number of nodes than can cause damages.

The connection of SAS shifting extends from the link matrix of SAS, which working state of nodes is taking into consideration as well, and can be denoted as:

$$\Phi(t) =< \Phi_w(t), \Phi_s(t) > . \tag{13}$$

where $\Phi_w(t)$ is the link matrix, $\Phi_s(t)$ is the matrix of working state $\Phi_s(t) = \{\Phi^d_{ns}(t), d = 1, 2, \cdots, N\}$, and for each node working state can be denoted as:

$$\Phi^d_{ns}(t) = \begin{cases} 0 & down \\ 1 & working \end{cases} \tag{14}$$

With the definition of shifting parameters, a corollary can be deduced to illustrate how MT techniques affect SAS.

Corollary 1: The necessary and sufficient condition for SAS shifting is changes in shifting parameter of SAS

Proof 1: First, the sufficiency can be deduced as follows. Under MTD environment, $\forall t \in (0, T)$, $\exists \Delta t$ make $M_{sas}(t) \neq M_{sas}(t+\Delta t)$. So $\exists m \in M_{sas}(t), \exists m' \in M_{sas}(t+\Delta t)$ and *typeof(m)* = *typeof(m′)* (the type of m and m' is same) make $m \neq m'$. And $\exists r \in G_{sas}(t), \exists r' \in G_{sas}(t+\Delta t)$ make $m = f_{sp}(r), m' = f_{sp}(r')$ and $f_{sp}(*)$ is the function to acquire shifting parameter from system resource. $\because m \neq m', \therefore r \neq r'$ and using the definition of SAS shifting, the SAS is shifting at time $t + \Delta t$.

Then the necessity can be deduced as follows. Under MTD environment, $\forall t \in (0, T)$, $\exists \Delta t$ make $G_{sas}(t) \neq G_{sas}(t+\Delta t)$. So $\exists r \in G_{sas}(t), \exists r' \in G_{sas}(t+\Delta t)$ make $r' = \varnothing \bigcup r \neq r' \bigcup W_{sas}(*, r, t) \neq W_{sas}(*, r', t+\Delta t) \bigcup W_{sas}(r, *, t) \neq W_{sas}(r', *, t+\Delta t)$. And $\exists m \in M_{sas}(t), \exists m' \in M_{sas}(t+\Delta t)$ make $m = f_{sp}(r), m' = f_{sp}(r')$. $\because r' = \varnothing \bigcup r \neq r' \bigcup W_{sas}(*, r, t) \neq W_{sas}(*, r', t+\Delta t) \bigcup W_{sas}(r, *, t) \neq W_{sas}(r', *, t+\Delta t)$, $\therefore m' = \varnothing \bigcup m \neq m'$. Using the definition for shifting parameter of SAS, the shifting parameter is changing at time $t + \Delta t$.

[2] The weight between n^d to other nodes is unified to $W_{sas}(n^d, *, t)$, and the weight between other nodes to node n^d is unified to $W_{sas}(*, n^d, t)$.

From the view of SAS, multi-layered MT techniques deployed in system can manipulate different system resource at each shifting period. With the corollary about SAS, the changes make by diverse MT techniques eventually reflect in the shifting parameters of SAS. For example, a network-based MTD can be described by network parameters of SAS, such as IP addresses, port numbers, and communication protocols. When NMTD shifts the IP address to a new one, the network position of shifting parameter is changes according to the shifting method of NMTD. By analyzing shifting parameter, our approach can describe how multi-layered MT technologies affect system resource and hinder adversaries.

C. SHIFTING PARAMETERS FOR MTD EFFECTIVENESS EVALUATION

Considering APT based on CKC under MTD environment, two sequences of shifting parameter, $Q_{esas} = \{M_{esas}(t), t \in (0, T)\}$ and $Q_{isas} = \{M_{isas}(t), t \in (0, T)\}$, can be generated by defender and attacker using AKM and MT configurations to demonstrate the ODP.

The sequence of ISAS is entirely determined by configurations of MT techniques and resources of the system, which describes how defenders manipulate configurations in the system. The manipulation of MTD can be modeled as a node N transforms its status from s_i to s_{i+1} in the possible configurable space (PCS) and denoted as $N(s_{i+1})$, and can be denoted as:

$$N(s_{i+1}) = R_{ns}(N(s_i), t, \omega, \mu) \tag{15}$$

where $R_{ns}(S, t, \omega, \mu)$ is the transition function changing configurations of nodes N in PCS ω at time t with shifting method μ.

Furthermore, to contain adaptive strategies in MT techniques, our approach divide these strategies into several simple constant strategies. Such as the adaptive shifting period, an adaptive method includes a set of constant shifting period. By modeling each constant shifting method, the whole adaptive strategy can be evaluated through the shifting parameters.

Moreover, The generation of ISAS is based on various shifting strategies of MTD, and can be denoted as:

$$Q_{isas} = G_{qis}(M_{isas}, \tau, \omega, \mu) \tag{16}$$

where $G_{qis}(*)$ is the generating function for ISAS sequence, τ is the shifting period, ω is the shifting space, and μ is the shifting method.

To evaluate the effectiveness of MTD, the sequence of ESAS can indicate the phase that adversaries step into. This approach not only gives qualitative conclusions to illustrate how MTD prevents APT but also can provide quantitative results by given specifical parameters of MTD and AKM.

Similar to the generation of Q_{isas}, the sequence of ESAS can be generated as denoted:

$$Q_{esas} = G_{qes}(M_{isas}, \delta) \tag{17}$$

where $G_{qes}(*)$ is the generating function for EASA sequence, δ is the model of AKM. The pseudo code of the EASA sequence generating algorithm, taking IP probing as an example, is given in Algorithm.1.

Algorithm 1 $Q_{esas}.L_{np}$ Generation Algorithm

```
Data: $M_{isas}$, $\delta$ (step by step), $t$, $d$
Result: $Q_{esas}.L_{np}$
$L_{np}.initial \leftarrow \min(M_{isas}.\omega.L_{np})$;
$probes(0, 1).L_{np} \leftarrow L_{np}.initial$;
for $t \in (0, T), d \in (1, N_0)$ do
    if $probes(t, d).L_{np} \notin \omega.L_{np}$ then
        $probes(t, d).L_{np} \leftarrow L_{np}.initial$;
    end
    if $t = \tau$ then
        $probes(t, d).L_{np} \leftarrow L_{np}.initial$;
    end
    if $probes(t, d).L_{np} = M_{isas}(t).L_{np}(d)$ then
        $Q(t, d) \leftarrow M_{isas}(t).L_{np}(d)$;
    else if $Q(t - 1, d) = M_{isas}(t - 1).L_{np}(d)$ &
    $Q(t - 1, d) = M_{isas}(t - 1).L_{np}(d)$ then
        $Q(t, d) \leftarrow M_{isas}(t).L_{np}(d)$;
    else
        $probes(t + 1, d).L_{np} \leftarrow probes(t, d).L_{np} + 1$;
        $Q(t, d) \leftarrow -1$;
    end
end
$Q_{esas}.L_{np} \leftarrow Q$;
```

Although the sequence generating algorithm can reveal the extent that ESAS approaches ISAS, the step of weaponization, which adversaries craft malware, can't reflect in the algorithm straightforwardly. To solve this problem, we convert the step of malware crafting to the step of vulnerabilities analyzing, which attackers analyze several vulnerabilities of the target step by step based on the attacker's capability. When all vulnerabilities in the target are analyzed, adversaries fully grasp all weak points and finish the crafting step. Coding and compiling of malware are neglected in the conversion, whereas the period of analysis is the critical part when attackers craft a malware.

Even though the sequence shifting parameter for ESAS reveals how SAS is shifting in the perspective of attackers, the extent that adversaries prevented by MT technologies are not discovered. A promising method to solve those problems is establishing a model to evaluate the attacking state changed under MTD environment.

V. NONHOMOGENEOUS HIERARCHICAL HIDDEN MARKOV MODEL FOR MTD EFFECTIVENESS EVALUATION

A. BASIC HIERARCHICAL HMM IN MTD EFFECTIVENESS EVALUATION

Existing assessment models either focus on measuring AS that how likely adversaries can exploit the vulnerabilities; or employ Attack Graph and Resource Graphs to exam-

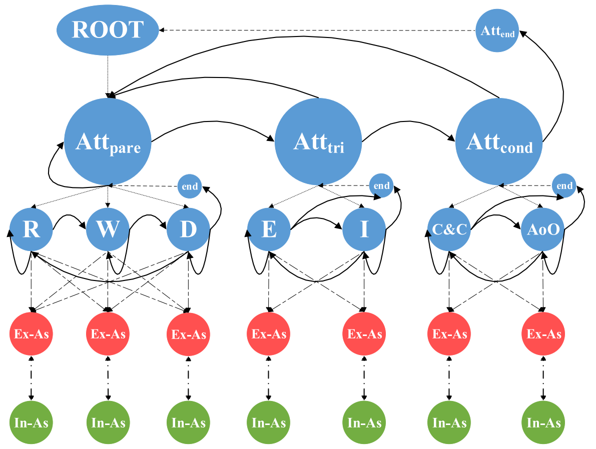


FIGURE 5. A Basic HHMM model for effectiveness assessment of MTD.

ine the difference between AS shifted by MT techniques. Those methods are either lack of system view or deficient in dynamic analysis, which may cause a loss of accuracy in enterprise-level systems and failure to evaluate the attacking state.

An HHMM [35] based effectiveness assessment for MTD (HHMMEA-MTD) is proposed in this paper to overcome these defects, which combines shifting parameter of SAS with attacking procedure of CKC. Its underlying model is shown in FIGURE.5. In HHMMEA-MTD, states of a attack are deduced by transition matrix and observation sequences, and the evaluation is given by analysis of most possible attacker's state sequence under MT techniques.

The basic HHMMEA-MTD contains three hidden level and an observation sequence as illustrated in FIGURE.5. The first hidden level is the root node where the model starts and ends. Every attack and defense process begins with a root node and goes through vertical and horizontal transformations, and finally returns to the root. The second hidden level is an abstract model of cyber-attack, which includes preparation, trigger, and conduction. Moreover, specific attack steps generated from the abstract model are in the third level. Taking APT and CKC as examples, each step from CKC forms a hidden state in the third level and belongs to a state in the second level. Reconnaissance, weaponization, and delivery are classified into preparation, where adversaries collect information about the target, and craft and deliver malware to the target. Exploitation and installation are parts of trigger. At the state of trigger, malware is triggered to exploit vulnerabilities and install back-doors or other tools into the target system. Command and control (C&C) and attack on objective (AoO) belong to conduction. Since C&C is the foundation of several attacks, adversaries carry out their objectives in conduction. For the observation sequence, the sequence of ESAS is used as the indicator to reveal how attacking states transit.

To evaluate multi-step ODPs that adversaries must compromise multiple targets to accomplish objectives, the basic HHMMEA-MTD can be extended to multi-step HHMMEA-MTD. In the extended model, multiple targets can be divided into two types, serial and parallel, which represent the typical attack of springboards and botnets. In serial multiple targets scenarios, a successful cyber-attack needs to compromise target in series, and the HHMMEA-MTD contains multiple sets of preparation, trigger, and conduction with corresponding specific attacking steps in the third hidden level to describe those attacking states of adversaries. The observation sequence also consists of ESAS that illustrate shifting parameters of adversaries about targets. In parallel multiple targets scenarios, adversaries must compromise multiple targets concurrently, and the HHMMEA-MTD can analyze those steps separately with a basic model, and then aggregate these results to evaluate.

To assess the effectiveness of MTD generally, this paper focuses on the basic HHMMEA-MTD, which is the foundation for both single-step and multi-step scenarios.[3] The HHMMEA-MTD is entirely determined by an initial matrix, a transition matrix, and an observation matrix, and can be denoted as:

$$\lambda = \{\{\Pi^{q^h}\}_{h\in(1,H-1)}, \{A^{q^h}\}_{h\in(1,H-1)}, \{B^{q^h}\}\}. \tag{18}$$

where q_i^h is the i-th state at h-th hierarchy, and the hierarchy index of the root is 1 and of production states is H. The Π^{q^h} is initial distribution vector of substate q^h and $\Pi^{q^h} = \{\pi^{q^h}(q_i^{h+1})\}$, which is the probability that state q^h will initially activate the state q_i^{h+1}. The state transition matrix denoted by $A^{q^h} = (a_{ij}^h)$, and $a_{ij}^h = P(q_j^{h+1}|q_i^{h+1})$ is the probability of making a horizontal transition from the i-th state to the j-th. Each production state q^H is solely parameterized by its output vector $B^{q^h} = \{b^{q^H}(k)\}$, and $b^{q^H}(k) = P(M_{esas}(t)|q^H)$ is the probability that the production state will interact with the Q_{esas}.

Since Π^{q^h} and CKC steps decide initial states that cannot be skipped, the initial probability matrix is entirely determined by the offensive process, and a successful cyber-attack has to follow a particular order of offensive move. Therefore, the initial matrix in HHMMEA-MTD is set to make sure that each initial state is the first state in sub-states.

As shown in FIGURE.5, the state transformation matrix contains two level. The first level is the internal state transition which includes *Preparation*(p), *Triggering*(t), *Conduction*(c) and a end state, and can be denoted as:

$$A_{root}^2 = \begin{bmatrix} a_{pp} & a_{pt} & 0 & 0 \\ a_{tp} & 0 & a_{tc} & 0 \\ a_{cp} & 0 & 0 & a_{cend} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

Production states are covered in second level, which are specific steps in CKC, such as reconnaissance(r), weaponization(w), delivery(d), exploitation(e), installation(i), command&control(c), and attack on objective(a), and a end state

[3] Without special notations, HHMMEA-MTD used in this paper refer to the basic model for concision.

TABLE 1. Relationship between shifting parameters and attacking states.

State	Shifting Parameter							
	L		Ω	S	Ψ		Φ	
	L_{np}	L_{pp}			Ψ_e	Ψ_p	Φ_s	Φ_ω
R	*	-	*	-	-	-	1	-
W	-	*	1	*	*	-	-	-
D	1	-	-	-	-	-	1	-
E	-	*	1	1	1	*	1	-
I	-	-	1	1	1	*	1	-
C&C	1	-	-	-	-	1	1	*
AoO	-	-	1	-	-	1	1	1

"1" stands for the prior conditional parameter required in the state, "*" stands for the parameter to be determined by this state and "-" stands for an independent parameter of the state.

for vertical transition, which can be denoted as:

$$A^3_{prep} = \begin{bmatrix} a_{rr} & a_{rw} & 0 & 0 \\ 0 & a_{ww} & a_{wd} & 0 \\ a_{dr} & 0 & a_{dd} & a_{dend} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$A^3_{tri} = \begin{bmatrix} a_{ee} & a_{ei} & a_{eend} \\ a_{ie} & a_{ii} & a_{iend} \\ 0 & 0 & 0 \end{bmatrix}$$
$$A^3_{cond} = \begin{bmatrix} a_{cc} & a_{ca} & a_{cend} \\ a_{aa} & a_{ac} & a_{aend} \\ 0 & 0 & 0 \end{bmatrix} \tag{20}$$

For production states, the observation matrix determines the relationship between Q_{esas} and attacking state. Moreover, the relation between CKC steps and shifting parameters of SAS is concluded in Table.1. Although those parameters can determine some attacking states, the condition is too coarse to align every observation to different attacking states, which may result in state overlapping. To solve this problem, a prior conditional based equitable methods to allocate various attacking states is proposed in this paper to generate the observation matrix. For overlapped elements in the production matrix, this method first generates the probability of i-th state align with j-th observation, $b^{q^H}_{i,j} = 1$, according to shifting parameter of SAS in Tabel.1, then normalizes overlapping probability as:

$$b^{q^H}_{i,j} = \frac{b^{q^H}_{i,j}}{\sum\limits_{j=0}^{J} b^{q^H}_{i,j}}. \tag{21}$$

B. NONHOMOGENEOUS EXTENDED HHMM IN MTD EFFECTIVENESS ASSESSMENT

Deployed by multi-layered MT technologies, the system shifts configurations frequently in the possible configurable space (PCS). In extreme condition, MT strategy reconfigures after each probing of adversaries, which leads to the perfect protection that makes knowledge collected by attackers invalid from time to time. However, the perfect shifting strategy is not acceptable regarding the performance cost. Therefore, the possible strategy employed by MT techniques is shifting in a variable or constant PCS periodically or aperiodically. Those shifting strategies in networks, hosts, and data finally map in the state transition matrix, which makes states transition nonhomogeneous. Therefore, a nonhomogeneous extended HHMMEA-MTD model (NHHMMEA-MTD) is presented to accommodate diverse strategies in multi-layered MT techniques.

Meanwhile, the nonhomogeneous characteristic is mainly reflected in the state transition matrix at the second and third hidden levels where the sequence of attacking states interrupted by MT techniques. For Network-based MTD (NMTD), strategies in NMTD can be cataloged into the variable or constant PCS, and periodical or aperiodic shifting. Moreover, the probability of a successful probe can be deduced by the Balls into Bins model and Generating Function [36]. Although the PCS of OS and service platform are limited in host-based MTD (HMTD), adversaries are impacted by rotation of diverse platform and self-cleaning techniques, which hinders adversaries at every shifting time. Moreover, data-based MTD (DMTD) technologies force adversaries to collect extra data to accomplish objectives, which results in spending extra time-steps in C&C and AoO.

To illustrate the practical strategies in the evaluation of effectiveness for MTD, probabilities in the transition matrix at the second and third hidden level are varying every logic-time or atom-operation of adversaries, which introduces non-homogeneous to the HHMM. The transition matrix in the second hidden level is an abstraction level where each state is activated only when attacking state enters or leaves, which does not generate sequences of observations. Except for the time, τ_0, when MT techniques manipulate system resource, the probability to step forward is set to 1, and the probability in transition matrix can be rewritten as:

$$a^2_{i,j}(t) = \begin{cases} 0, & otherwise. \\ 1, & j - i = 1. \end{cases}, \quad t \neq \tau_0 \tag{22}$$

At every shifting time, the probability to step back from other states to the initial state is set to 1, and the probability in the transition matrix can be rewritten as:

$$a^2_{i,j}(\tau_0) = \begin{cases} 0, & otherwise. \\ 1, & j = 1. \end{cases} \tag{23}$$

For the transition matrix in the third hidden level, both AKM and MT shifting strategies make those matrices varying after atom-operations of adversaries or manipulations of system resources. Same as the matrix at the second level, different stages of attacks are forced to step back to the initial state at every shifting time. At *Preparation* stage, a Balls into Bins model can be used to summarize the knowledge acquisition under MT techniques, and the probability can be denoted as:

$$a_{rw}(L_p) = 1 - (1 - P(T_\tau)^J)(1 - P(\sigma))$$
$$P(L_p) = \sum_{i=b}^{L_p} \frac{N(i)}{s \cdot b}$$

$$N(L_p) = \sum_{r=0}^{\lfloor \frac{L_p-b}{s} \rfloor} (-1)^r C_b^r C_{L_p-b-sr}^{L_p-sr-1} \tag{24}$$

where $L_p = J \cdot T_\tau + \sigma$ stands for probing times adversaries initiated, $s \cdot b$ stands for PCS with b blocks and s space in each block, and $\lfloor * \rfloor$ is the round down function.

Moreover, crafting and delivery of malware, exploitation of vulnerabilities and installation of backdoors, and command&control of targets can be modeled by an Urn model without replacement. Taking crafting of malware and c&c of targets as examples, the former can be used to demonstrate the AKM against MT strategies that adversaries exhaust the possible space gradually, and the later can be used to illustrate MT techniques against different attacking objectives that need certain times to accomplish, which can be denoted as:

$$a^*_{akm} = \frac{N^{probe}_{done}}{N^{pcs}_{min}} \tag{25}$$

$$a^*_{aoo} = \frac{N^{c\&c}_{done}}{N^{dem}_{min}} \tag{26}$$

where N^{probe}_{done} stands for the number of probes that have done by the attacker, N^{pcs}_{min} is the minimal space in PCS, $N^{c\&c}_{done}$ stands for the times that attack have command & control the target, and N^{dem}_{min} is the minimal demand of accomplishing a attack.

Since end states at every production matrix make vertical transitions, the probability to transition into the end state, except for A^3_{prep}, is set to 1 after every shifting time of MTD.

C. PARTIAL VITERBI ALGORITHM FOR NHHMM IN MTD EFFECTIVENESS ASSESSMENT

To calculate the most probable sequence of attacking state from shifting parameter of ESAS sequence, Generalized Viterbi Algorithm (GVA) is proposed to solve the problem of hierarchical transition for HHMM [35]. Generalized Viterbi Algorithm defines $\delta_{glo}(t, t+k, q_i^h, q^{h-1})$ to be the likelihood of the most probable (hierarchical) state sequence generating observation sequence $o_t \cdots o_{t+k}$ given that q^{h-1} was entered at time t, its substate q_i^h was the last state to be activated by q^h, and control returned to q^h at time $t+k$. By two additional variables, $\psi(t, t+k, q_i^h, q^{h-1})$ is the index of the most probable state to be activated by q^{h-1} before activating q_i^h, and $t' = \tau(t, t+k, q_i^h, q^h), t \leq t' \leq t+k$ is the time when q_i^h was activated by q^h. Given these two variables the most probable hierarchical state sequence is obtained by scanning the lists ψ and τ from the root state to the production states. However, extended by time-varying transition matrix, the GVA is not workable to calculate the state sequence in NHHMM by global optimal variables.

In consideration of the specific model in NHHMMEA-MTD, a Partial Viterbi Algorithm (PVA) is presented to solve the problem by calculating local optimum. The second hidden level consists of general cyber-attacking states which guide the production level generate the observation sequence at a proper time. Since internal states do not yield observations, each internal state can be regarded as a separator to split production states. Therefore, optimal global variables can be divided into several local optimums that only record the likelihood of the most probable attacking state sequence under its parent internal state. The local optimum can be denoted as:

$$\delta^j_{loc} = (t_j, t_j + k_j, q_i^h, q_j^{h-1}) \tag{27}$$

where q_j^{h-1} is the parent state that initiate q_i^h, t_j is the time that q_i^h is activated by q_j^{h-1} and $t_j + k_j$ is the time returned to q_j^{h-1}. To simply the calculation of δ^j_{loc}, a *MAX* function, which parameters are a function f and a finite set S, is defined as:

$$MAX_{l \in S}\{f(l)\} \triangleq \left(\max_{l \in S} f(l), \arg\max_{l \in S} f(l) \right) \tag{28}$$

By limiting the optimum in the local autonomous region between upper abstract state and its sub-states, PVA ignores the optimum global changing brought by time-varying transitions and calculates the sequence through every autonomous model ignoring the influence of the trans-neighboring states.

Unfortunately, the PVA is not a generalized algorithm that solves all NHHMMs in finding the most probable state sequence. Two prior conditions determine the scope of application for PVA. Firstly, the production matrix between production states and observations have to be partially corresponding. This condition ensures that even though the transition matrix changes over time, its production states will not produce the corresponding observations. Secondly, the vertical transition has to be non-negligible. Therefore, the vertical transition is used as the separator between the autonomous region.

Similar to GVA, two additional variables, $\psi^j_{loc}(t_j, t_j + k_j, q_i^h, q_j^{h-1})$ and $\tau^j_{loc}(t_j, t_j + k_j, q_i^h, q_j^{h-1})$, are use to record prior index and time. Given these two variables the most probable hierarchical state sequence is obtained by scanning the lists $\sum_{j=1}^{J} \psi^j_{loc}$ and $\sum_{j=1}^{J} \tau^j_{loc}$ from the root state to the production states. If a breadth-first-search is used for scanning, then the states are listed by their level index from top to bottom. If a depth-first search is used, then the states are listed by their activation time. Moreover, the pseudo-code of PVA is shown in the appendix.

VI. CASE STUDY AND RESULTS ANALYSIS

A. CASES DESCRIPTION

The case study is presented to validate the feasibility of the proposed assessment model of effectiveness for MTD and advantages of SAS shifting parameters compared to traditional methods in a MATLAB simulation. The typical network topology used in our simulation is shown in FIGURE.4. There are three servers in the internal network, including Database server, Web server, and FTP server, and their basic configurations are shown in TABLE.2 and TABLE.3, which factors can be collected by a probe or agent working in

TABLE 2. Configurations in typical information system.

Node	Network Parameter			Host Parameter			MTD Deployment		
	IP	Port	Protocol	OS	Platform	Support	Net-based	Host-based	Data-based
Database Server(DbS)	192.168.1.*	3306	SQL	Linux	MySql	-	-	-	Format
Web Server(WbS)	192.168.1.* 10.155.33.*	80	HTTP	Linux*	Apache*	JavaVM	IP	OS/Ser-Plat	-
FTP Server(FtpS)	192.168.1.*	21	FTP	Linux	vsftpd	-	IP	-	-

"*" stands for multiple configurations,"-" stands for uncovered configurations.

TABLE 3. Versions in information system.

Type	Software	Version
OS	Ubuntu Server	12.04
	Redhat Enterprise	6.0
	Window Server	2012
Web	Apache	2.4.3
	Nginx	1.0.15
	IIS	8.0

TABLE 4. Parameters of offense and defense in information system.

Role	Type	Parameter
Defense	IP hopping	τ_{hf}:1/50(t) τ_{lf}:1/100(t) ω:200
	OS rotation	τ_{hf}:1/100(t) τ_{lf}:1/200(t) ω:3
	Service mutation	τ_{hf}:1/150(t) τ_{lf}:1/300(t) ω:3
	Data re-formatting	τ:- ω:5
Offense	Detect-ability	1 target per t
	Craft-ability	50 parts of malware per t
	Avg. C&C	$\approx 50t$

"t" stands for the unit of logic-time steps

the node in practice. The strategies of IP hopping, OS rotation and Service-Platform mutation are purely random and triggered by shifting time without shifting space resizing. The connectivity limits external users to connect other nodes but the Web server and only the Web server can exchange data to the Database server by configuring the access control policy in internal firewalls. Besides, TABLE.5 illustrates vulnerabilities in those configurations obtained by an automated analyzer of vulnerabilities that connecting with CVSS and CVE database. Moreover, settings of MTD and ability of adversaries are demonstrated in TABLE.4. From these descriptions, the objective of adversaries in this paper is to obtain confidential data from the Database server in the internal network and transmit these data to the client controlled by the attacker in the external network.

The NHHMMEA-MTD model is constructed and the assessment of effectiveness is evaluated as follows:

1) ACQUIREMENT OF SHIFTING PARAMETER

The typical topology of information system shows that internal connections between servers and clients are limited by internal firewalls, and external users are only allowed to connect the Web server. Therefore, the initial connection weight can be acquired as:

$$W_{as}(t_0) = \begin{bmatrix} 1 & B_{DW} & B_{FW} & B_{IW} & B_{EW} & B_{AW} \\ B_{WD} & 1 & 0 & 0 & 0 & 0 \\ B_{WF} & 0 & 1 & 0 & 0 & 0 \\ B_{WI} & 0 & 0 & 1 & 0 & 0 \\ B_{WE} & 0 & 0 & 0 & 1 & B_{AE} \\ B_{WA} & 0 & 0 & 0 & B_{EA} & 1 \end{bmatrix}$$

where B_* is the bandwidth of the connection between servers and clients.[4] And in this case, the internal bandwidth is set to 1000Mbps, and the external bandwidth is set to 100Mbps.

According to the definition of shifting parameter for each nodes in SAS, parameters are acquired from configurations of system. For basic configurations in the system, these parameters, such as network, host and data parameters, can be collected by a probe or agent working in the node, which is shown in TABLE.2 and TABLE.3. For vulnerabilities in the system, these parameters can be obtained from CVSS and CVE database shown in TABLE.5, which we develop an automated analyzer in python to collecting those factors according to configurations of the system. To contains characteristics of MT techniques, the initial states of those parameter are choosing from those configurations and versions randomly.

2) GENERATION OF SAS SHIFTING SEQUENCE

With initial states of SAS, the sequence of shifting parameter for ISAS can be generated by diverse strategies of MT technologies. For systems protected by multiple MT techniques, the sequence of shifting parameters for ISAS is varying at every shifting time by multi-layered MT techniques. Q_{isas} shows the capability of MTD to manipulate system resource, which is generated according to configurations of MTD listed in TABLE.4. Meanwhile, using the generation algorithm, Q_{esas} is generated to demonstrate how adversaries gather information, craft malware and conduct exploitation against multiple protection of MTD, which is generated according to TABLE.4. Those sequence with details in several shifting points for this case is shown in TABLE.6.

3) CALCULATION FOR THE MOST PROBABLE SEQUENCE OF ATTACKING STATE

According to the Q_{esas} and transition matrix at every probe, the most probable sequence of attacking state is calculated by PVA. As shown in pseudo-code, each part of NHHMM can be solved by GVA, and the coalescence is done by PVA. Results of the most probable sequence of attacking state under different MT technologies are shown in FIGURE.6 to FIGURE.8. Quantitative results got from 50 replicated evaluations are shown in TABLE.7 under two typical settings.

[4] The capital character strands for different nodes. W=Web-server, D=Db-server, F=Ftp-server, I=Internal-user, E=External-user and A=Attacker.

TABLE 5. Information about vulnerability and threat.

Software	Num.	Vulnerability Score					
		Base(Sum.)	Base(Avg.)	Impact (Sum.)	Impact(Avg.)	Exploitability(Sum.)	Exploitability(Avg.)
Ubuntu	658	3727.3	5.7	3408.9	5.2	5270.9	8.0
Redhat	153	995.6	6.5	938.4	6.1	1283.1	8.4
Windows	533	3350.1	6.3	3931.9	7.4	3215.3	6.0
Apache	16	82.2	5.1	55.4	3.5	151.6	9.5
Nginx	1	4.3	4.3	2.9	2.9	8.6	8.6
IIS	1	5.1	5.1	6.4	6.4	4.9	4.9

Time span: 2012 to 2017; Metric: CVSS2.0

TABLE 6. Examples for M_{sas} in several shifting times.

Nodes	Shifting Parameter										
	L		Ω			S			Ψ		Φ	
	L_{np}	L_{pp}	ω_{os}	ω_{pt}	ω_{sp}	S_n	S_i	S_{zd}	Ψ_e	Ψ_p	Φ_s	Φ_ω
Web@(τ_1^-)	192.168.1.100	Oxc380c0	Ubuntu	Apache	Java-VM	658;16;-	5.2;3.5;-	-	8.0;9.5;-	admin/guest	1	$W_{as}(\tau_1^-)$
Web@(τ_1^+)	192.168.1.123	Oxc380c0	Ubuntu	Apache	Java-VM	658;16;-	5.2;3.5;-	-	8.0;9.5;-	admin/guest	1	$W_{as}(\tau_1^+)$
Web@(τ_2^-)	192.168.1.123	Oxc380c0	Ubuntu	Apache	Java-VM	658;16;-	5.2;3.5;-	-	8.0;9.5;-	admin/guest	1	$W_{as}(\tau_2^-)$
Web@(τ_2^+)	192.168.1.123	Oxa2356b	Redhat	Nginx	Java-VM	153;1;-	6.1;2.9;-	-	8.4;8.6;-	admin/guest	1	$W_{as}(\tau_2^+)$
Web@(τ_3^-)	192.168.1.123	Oxa2356b	Redhat	Nginx	Java-VM	153;1;-	6.1;2.9;-	-	8.4;8.6;-	admin/guest	1	$W_{as}(\tau_3^-)$
Web@(τ_3^+)	192.168.1.165	Oxb568d2	Windows	IIS	-	658;1;-	5.2;6.4;-	-	8.0;4.9;-	admin/guest	1	$W_{as}(\tau_3^+)$

"τ_*^-" stands for the time before MT shifting and "τ_*^+" stands for the time after MT shifting

TABLE 7. Quantitative results for MTD.

MTD		Attacking states results									
Type	Settings	Min. time to success			Max. time to success			Avg. time to success			Conversion
		Crafting	Other	Sum	Crafting	Other	Sum	Crafting	Other	Sum	(second)
None-MTD	-	6	66	72	12	264	273	7.68	167.22	174.9	4775.22
Net-MTD	High-SFR	6	73	79	20	773	793	8.88	337.14	346.02	5665.14
	Low-SFR	6	85	91	27	1573	1600	12.4	699.9	712.3	8139.9
Multi-MTD	High-SFR	6	62	68	30	1226	1256	10.36	339.24	349.6	6555.24
	Low-SFR	6	90	96	42	1952	1994	22.78	802.14	824.92	14470.14

"High-SFR" stands for high space-frequency ratio which τ_{lf} is used and "Low-SFR" stands for low space-frequency ratio which τ_{hf} is used.

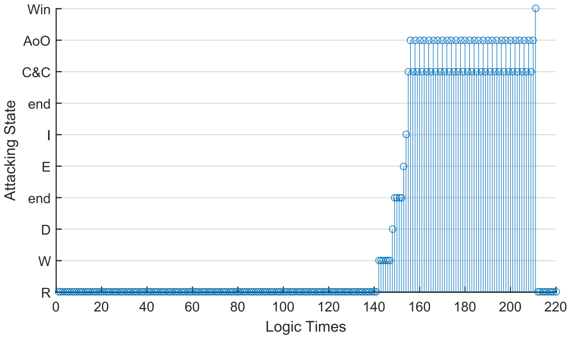


FIGURE 6. The Sequence of Attacking States for none-MTD Environment.

B. RESULTS ANALYSIS

1) THE ANALYSIS OF SHIFTING PARAMETER UNDER DIVERSE MTD TECHNIQUES

From TABLE.6, shifting parameters of SAS reflect changes in the information system, and diverse MTD deployed in multi-layer can manipulate several system resources. Taking τ_* as the transition time, the superscript indicates that the time after or before the transition time, and our results show the distinction of shifting parameters between τ_*^- and τ_*^+. There are three typical shiftings shown in examples, such as IP hopping, OS Rotation, and Service Platform Mutation. Therefore, shifting parameters of SAS illustrate the extent of attack surface shifting, and results show that different MT technologies affect SAS shifting in various parts. Such as IP hopping in the network and ASLR in the memory, those methods randomly change the position of attack surface and reduce the attack window that adversaries can connect with. Meanwhile, OS Rotation and Service Platform Mutation in the hosts reform the shape, size, and intensity of SAS, and increase the cost of crafting malware and limit the range of impact. Although, Data Randomization is not explicitly reflected in shifting parameters, the effect of Data Randomization will finally reflect in the evaluation of MTD.

2) THE ANALYSIS OF EFFECTIVENESS ASSESSMENT FOR MTD

Since initial states of our assessments are generated randomly according to TABLE.2, TABLE.3 and TABLE.4, and adversaries are set to start from the very beginning of the PCS, the result of evaluation for MT techniques are random as well. Moreover, typical results under diverse MTD environment are given to compare and analyze the effectiveness. Quantitative results are shown in TABLE.7, which evaluated three types of defense, static none-MTD, single-layered NMTD, and

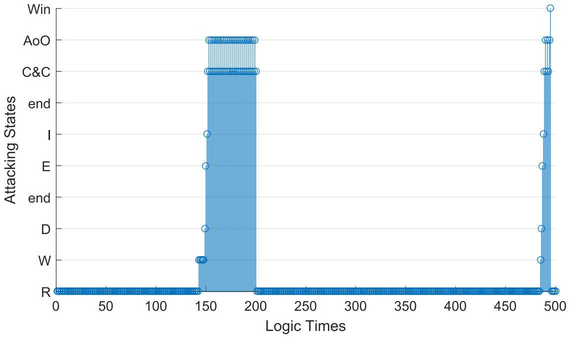


FIGURE 7. The Sequence of Attacking States for network based MTD Environment.

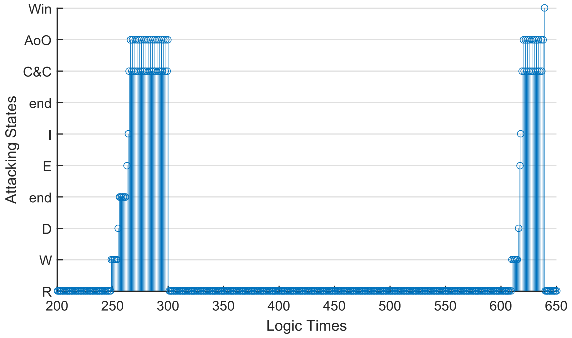


FIGURE 8. The Sequence of Attacking States for multi-layered MTD Environment.

multi-layered MTD (NMTD + HMTD) in two options of the parameter set, high space-frequency ratio and low space-frequency ratio, under the typical topology of the information system.

FIGURE.6 to FIGURE.8 illustrate the sequence of attacking state where adversaries reconnoiter a target, craft malware, exploit vulnerabilities, install backdoors, and establish controlled channels. Moreover, the sequence reflects how MT techniques interrupt the process of attack: IP hopping disturbs reconnaissance and break channels, OS and service transition force attackers to re-craft, re-exploit, and re-install, and data randomization prolongs the necessary times of connections.

FIGURE.6 gives a baseline of evaluation compared to various MT technologies. When adversaries are against nome-MTD environment, attacking states are transited step by step. Attackers are unimpeded in gathering information, conducting exploitation and accomplishing objectives. Therefore, our evaluation shows that fallbacks and re-doings are not contained in the sequence of attacking states.

Assessment results of network-based MT technologies are illustrated in FIGURE.7. From the aspect of transition, attacking states are forced to backspace and re-do the process of exploration. Moreover, probing times spent in reconnaissance are significantly increased compared to the none-MTD scenario. Those changes in evaluation stand for promotion in security and extra costs spent by adversaries. Moreover, configurations in network-based MT techniques also influence adversaries. When the shifting period shrinks, the location of SAS is more mutative, and adversaries are interrupted more frequently. At the same time, an enlarged possible configuration space forces adversary to dedicate more efforts in information acquisition. These results confirm our intuition and previous studies.

Evaluation of effectiveness when deploying multi-layered MT techniques to the system is demonstrated in FIGURE.8. As multi-layered MT techniques shift multiple dimensions of SAS, adversaries are impeded in multiple stages. Different from the shifting of single parameter, multi-layered MTD introduces a sophisticated and comprehensive combination of diverse methods to manipulate system resource. Those manipulations promote the efficiency to stop ongoing attacks which are hindered in stages of preparation, triggering and conduction. Meanwhile, the logic-time spent by attackers to accomplish objectives is longer when compared to single-layered MTD. Moreover, compared to network-based MT technologies, OS Rotation and service platform mutation not only force attacking states to re-start information gathering, but also make malware and back-doors invalid, which prevents the spreading of malware and increases efforts by adversaries to re-craft, re-exploit, and re-install.

Quantitative results for diverse MT technologies under two settings are shown in TABLE.7. To compare diverse technologies and different settings, we introduce an unprotected result as the baseline. Hindered by single network-based MT technologies, adversaries have to spend more time before achieving goals. The more frequent the shifting is, the more time the attacker has to pay in re-reconnaissance. Although from the view of logic-time, multi-layered MT techniques do not promote the security of systems dramatically. Converted to real-time scales, with multi-layered techniques deployed into the system, average times consumed by adversaries to accomplish objectives are increased significantly. The results in TABLE.7 illustrate that multi-layered deployment of MT techniques set in high shifting frequency can tremendously increase the effectiveness of MTD.

VII. CONCLUSION AND FUTURE WORK

Moving target defense is a game-changing technique to overturn attackers' advantages of time, space, cost and price in cyber-space. How to evaluate the effectiveness of MTD is a fundamental problem for deploying defense, selecting strategies, and achieving trade-off. An effectiveness evaluation model for moving target defense based on system attack surface is proposed to extend the evaluations of inaccurately describing the MTD confrontation by existing models and cope with difficulties in analyzing multi-layered MT technologies in enterprise-class topologies effectively. This model focuses on the offensive and defensive process in cyberspace to demonstrate additional consumption of adversaries brought by MT techniques. Regarding model construction, attack surface model is extended to system view and a group of shifting parameters is defined to illustrate the manipulation of system resources. Generating the sequence of those shifting parameters, the offensive and defensive

Algorithm 2 Partial Veterbi Algorithm

```
Data: $Q_{esas}, \lambda = \{\Pi, A, B\}, P_{mtd}, MaxT$
Result: $Q_{state}$
Initialization: $Att.win = False, Att.vcf = False, t = 1, s = 1, N_{porbe}$;
while $Att.win \neq True$ do
    if $t > MaxT$ then
        /* we will break loops if $t$ beyond $MaxT$.                                  */
        $\delta_{glo} = \sum_{s=1}^{S} \delta_{loc}^{s}$;
        break;
    end
    /* we update transition matrix in NHHMM.                                          */
    $\lambda = update_M(\lambda, t, N_{porbe}), P_{mtd}$;
    /* Firstly, we calculate at the production states.                                */
    if $t = 1$ then
        $\delta_{loc}^{s}(t, q_i^H, q_s^{H-1}) = [\pi^{q_s^{H-1}}(q_i^H)][b^{q_i^H}(Q_{esas}^t)]$;
        $\psi_{loc}^{s}(t, q_i^H, q_s^{H-1}) = 0$;
        $\tau_{loc}^{s}(t, q_i^H, q_s^{H-1}) = t$;
    else
        $[\delta_{loc}^{s}(t, q_i^H, q_s^{H-1}), \psi_{loc}^{s}(t, q_i^H, q_s^{H-1})] = MAX_{1<j<|q_s^{H-1}|} \left\{ \delta_{loc}^{s}(t-1, q_i^H, q_s^{H-1}) a_{ji}^{q_s^{H-1}} [b^{q_i^H}(Q_{esas}^t)] \right\}$;
        $\tau_{loc}^{s}(t, q_i^H, q_s^{H-1}) = t$;
        /* if reaching productive end states, we transit to parent states.            */
        if $s \in S_{end}^{Production}$ then
            $\delta_{loc}^{s}(t, q_i^h, q_s^{h-1}) = \max_{1<j<|q_i^h|}[\pi^{q_s^{h-1}}(q_i^h)\delta(t, q_i^h, q_j^{h+1}) a_{jend}^{q_i^h}]$;
            $\psi_{loc}^{s}(t, q_i^h, q_s^{h-1}) = 0$;
            $\tau_{loc}^{s}(t, q_i^h, q_s^{h-1}) = t$;
            $Att.vcf = True$;
        end
    end
    /* Then, we calculate at internal states.                                         */
    if $Att.vcf$ then
        $R = \max_{1<r<|q_s^h|} \left\{ \delta_{loc}^{s}(t, q_s^h, q_r^{h+1}) a_{rend}^{q_s^h} \right\}$;
        $[\delta_{loc}^{s}(t, q_s^h, q^{h-1}), \psi_{loc}^{s}(t, q_s^h, q^{h-1})] = MAX_{1<j<|q^{h-1}|} \left\{ \delta_{loc}^{s}(t-1, q_s^h, q^{h-1}) a_{ji}^{q^{h-1}} R \right\}$;
        $\tau_{loc}^{s}(t, q_s^h, q^{h-1}) = t$;
        if $s = S_{end}^{parents}$ then
            /* if reaching parent end states, attacker wins.                          */
            $Att.win = True$;
            $\delta_{glo} = \sum_{s=1}^{S} \delta_{loc}^{s}$;
        else
            /* else going back to production states.                                  */
            $\delta(t, q_{s+1}^H, q^{H-1}) = [\pi^{q^{H-1}}(q_{s+1}^H)][b^{q_{s+1}^H}(Q_{esas}^t)]$;
            $\psi(t, q_{s+1}^H, q^{H-1}) = s$;
            $\tau(t, q_{s+1}^H, q^{H-1}) = t$;
            $Att.vcf = False$;
            if $\delta(t, q_{s+1}^H, q^{H-1}) > 0$ then
                $s = s + 1$;
            end
        end
    end
    /* updating $N_{probe}$ according to $t$, $s$ and $\delta$.                       */
    $N_{probe} = update_N[N_{probe}, t, s]$;
    $t = t + 1$;
end
$Q_{state} = search(\delta_{glo}, t - 1)$;
```

TABLE 8. Abbreviations used in this paper.

Abbreviations	Description
MTD	moving target defense
MT	moving target
NMTD	network-based MTD
HMTD	host-based MTD
DMTD	data-based MTD
APT	advanced persistent threat
CKC	cyber kill chain
ODP	offensive and defensive precess
AKM	attacking knowledge model
AS	attack surface
SAS	system attack surface
ESAS	external system attack surface
ISAS	internal system attack surface
HMM	hidden Markov model
HHMM	hierarchical HMM
NHHMM	nonhomogeneous HHMM
HHMMEA-MTD	HHMM based effectiveness assessment for MTD
NHHMMEA-MTD	NHHMM based effectiveness assessment for MTD
PCS	possible configurable space
GVA	general Viterbi algorithm
PVA	partial Viterbi algorithm
SFR	space-frequency ratio

process is modeled. Moreover, the sequence of attacking states is defined to describe how MT technologies handle adversaries. The nonhomogeneous hierarchical hidden Markov model evaluates the effectiveness in depicting the status-changing of adversaries under system resources manipulated by multi-layered MT techniques. The model splits different attacking states into three abstracted stages to refine transition of states, introduces attempts-varying variables into transition matrices to instantiate procedures of attack, and employs the partial Viterbi algorithm to calculate the most probable sequence of attacking state. Finally, a method of logic and real-time conversion is presented to apply our model to practical information systems.

Despite all the endeavors having been made, employing our evaluation results in deployment and optimization requires further theoretical derivation and reproducible experiments. Besides, we need to have further studies on how to combine MTD defense with other means of defense to figure out a more comprehensive defensive method. This work will be carried out in the future.

APPENDIX A PSEUDO-CODE OF PARTIAL VITERBI ALGORITHM

The detailed pseudo-code of Partial Viterbi Algorithm is shown in Algorithm.2.

APPENDIX B ABBREVIATIONS AND NOTATIONS

Main abbreviations used in the paper are listed in TABLE.8.

ACKNOWLEDGMENT

The authors would like to thank for the valuable review comments from all experts and editors.

REFERENCES

[1] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, 1st ed. New York, NY, USA: Springer-Verlag, 2011.

[2] L. Turner, "Searching the deep Web," *J. Comput. Sci. Colleges*, vol. 17, no. 1, p. 258, Oct. 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=772488.772530

[3] H. Okhravi, T. Hobson, D. Bigelow, and W. Streilein, "Finding focus in the blur of moving-target techniques," *IEEE Security Privacy*, vol. 12, no. 2, pp. 16–26, Mar. 2014.

[4] P. K. Manadhata and J. M. Wing, *A Formal Model for a System's Attack Surface*. New York, NY, USA: Springer, 2011, pp. 1–28.

[5] M. Howard, J. Pincus, and J. M. Wing, *Measuring Relative Attack Surfaces*. Boston, MA, USA: Springer, 2005, pp. 109–137.

[6] P. Manadhata and J. M. Wing, "Measuring a system's attack surface," *Adv. Inf. Secur.*, vol. 54, pp. 1–28, Jan. 2004.

[7] P. K. Manadhata and J. M. Wing, "An attack surface metric," *IEEE Trans. Softw. Eng.*, vol. 37, no. 3, pp. 371–386, May 2011.

[8] 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 Trans. Dependable Secure Comput.*, vol. 11, no. 1, pp. 30–44, Jan. 2014.

[9] K. Sun and S. Jajodia, "Protecting enterprise networks through attack surface expansion," in *Proc. Workshop Cyber. Secur. Anal., Intell. Automat. (SafeConfig)*, New York, NY, USA, pp. 29–32, doi: 10.1145/2665936.2665939.

[10] M. Albanese, E. Battista, S. Jajodia, and V. Casola, "Manipulating the attacker's view of a system's attack surface," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2014, pp. 472–480.

[11] P. K. Manadhata, "Game theoretic approaches to attack surface shifting," in *Moving Target Defense II*, S. Jajodia, A. K. Ghosh, V. Subrahmanian, V. Swarup, C. Wang, and X. S. Wang, Eds. New York, NY, USA: Springer, 2013, pp. 1–13.

[12] R. Zhuang, S. A. DeLoach, and X. Ou, "Towards a theory of moving target defense," in *Proc. 1st ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2014, pp. 31–40, doi: 10.1145/2663474.2663479.

[13] R. Zhuang, A. G. Bardas, S. A. DeLoach, and X. Ou, "A theory of cyber attacks: A step towards analyzing MTD systems," in *Proc. 2nd ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2015, pp. 11–20, doi: 10.1145/2808475.2808478.

[14] X. Xiong, G. Zhao, and X. Wang, "A system attack surface based MTD effectiveness and cost quantification framework," in *Proc. 2nd Int. Conf. Cryptogr., Secur. Privacy (ICCSP)*, New York, NY, USA, 2018, pp. 175–179, doi: 10.1145/3199478.3199487.

[15] B. Friedman, "A simple urn model," *Commun. Pure Appl. Math.*, vol. 2, no. 1, pp. 59–70, 2010.

[16] P. Berenbrink, K. Khodamoradi, T. Sauerwald, and A. Stauffer, "Balls-into-bins with nearly optimal load distribution," in *Proc. 25th Annu. ACM Symp. Parallelism Algorithms Archit. (SPAA)*, New York, NY, USA, 2013, pp. 326–335, doi: 10.1145/2486159.2486191.

[17] T. E. Carroll, M. Crouse, E. W. Fulp, and K. S. Berenhaut, "Analysis of network address shuffling as a moving target defense," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2014, pp. 701–706.

[18] M. Crouse, B. Prosser, and E. W. Fulp, "Probabilistic performance analysis of moving target and deception reconnaissance defenses," in *Proc. 2nd ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2015, pp. 21–29, doi: 10.1145/2808475.2808480.

[19] Y.-B. Luo, B.-S. Wang, and G.-L. Cai, "Effectiveness of port hopping as a moving target defense," in *Proc. IEEE Int. Conf. Secur. Technol.*, Dec. 2014, pp. 7–10.

[20] D. Evans, A. Nguyen-Tuong, and J. Knight, *Effectiveness of Moving Target Defenses*. New York, NY, USA: Springer, 2011, pp. 29–48, doi: 10.1007/978-1-4614-0977-9_2.

[21] H. Okhravi, J. Riordan, and K. Carter, "Quantitative evaluation of dynamic platform techniques as a defensive mechanism," in *Research in Attacks, Intrusions and Defenses*, A. Stavrou, H. Bos, and G. Portokalidis, Eds. Cham, Switzerland: Springer, 2014, pp. 405–425.

[22] J. Xu, P. Guo, M. Zhao, R. F. Erbacher, M. Zhu, and P. Liu, "Comparing different moving target defense techniques," in *Proc. 1st ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2014, pp. 97–107, doi: 10.1145/2663474.2663486.

[23] H. Maleki, S. Valizadeh, W. Koch, A. Bestavros, and M. van Dijk, "Markov modeling of moving target defense games," in *Proc. ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2016, pp. 81–92, doi: 10.1145/2995272.2995273.

[24] Q. L. Nguyen and A. Sood, "Improving resilience of SOA services along space-time dimensions," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN)*, Jun. 2012, pp. 1–6.

[25] A. Prakash and M. P. Wellman, "Empirical game-theoretic analysis for moving target defense," in *Proc. 2nd ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2015, pp. 57–65, doi: 10.1145/2808475.2808483.

[26] S. T. Jones *et al.*, "PLADD: Deterring attacks on cyber systems and moving target defense," Sandia Nat. Lab., Albuquerque, NM, USA, Tech. Rep. SAND2017-0412C 650438, Jan. 2017.

[27] N. Ben-Asher, J. Morris-King, B. Thompson, and W. J. Glodek, "Attacker skill, defender strategies, and the effectiveness of migration-based moving target defense in cyber systems," in *Proc. Int. Conf. Cyber Warfare Secur. (ICCWS)*, 2016, pp. 21–30.

[28] Q. Zhu and T. Başar, "Game-theoretic approach to feedback-driven multi-stage moving target defense," in *Decision and Game Theory for Security*, S. K. Das, C. Nita-Rotaru, and M. Kantarcioglu, Eds. Cham, Switzerland: Springer, 2013, pp. 246–263.

[29] C. Lei, D.-H. Ma, H.-Q. Zhang, and L.-M. Wang, "Moving target network defense effectiveness evaluation based on change-point detection," *Math. Problems Eng.*, vol. 2016, no. 11, pp. 1–11, 2016.

[30] J. R. Hamlet and C. C. Lamb, "Dependency graph analysis and moving target defense selection," in *Proc. ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2016, pp. 105–116, doi: 10.1145/2995272.2995277.

[31] R. Zhuang, S. Zhang, S. A. DeLoach, X. Ou, and A. Singhal, "Simulation-based approaches to studying effectiveness of moving-target network defense," in *Proc. Nat. Symp. Moving Target Res.*, vol. 246, 2012, pp. 1–12.

[32] A. G. Bardas, S. C. Sundaramurthy, X. Ou, and S. A. DeLoach, "MTD CBITS: Moving target defense for cloud-based it systems," in *Computer Security—ESORICS*, S. N. Foley, D. Gollmann, and E. Snekkenes, Eds. Cham, Switzerland: Springer, 2017, pp. 167–186.

[33] K. Zaffarano, J. Taylor, and S. Hamilton, "A quantitative framework for moving target defense effectiveness evaluation," in *Proc. 2nd ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2015, pp. 3–10, doi: 10.1145/2808475.2808476.

[34] J. Taylor, K. Zaffarano, B. Koller, C. Bancroft, and J. Syversen, "Automated effectiveness evaluation of moving target defenses: Metrics for missions and attacks," in *Proc. ACM Workshop Moving Target Defense (MTD)*, New York, NY, USA, 2016, pp. 129–134, doi: 10.1145/2995272.2995282.

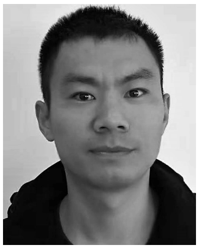
[35] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," *Mach. Learn.*, vol. 32, no. 1, pp. 41–62, Jul. 1998, doi: 10.1023/A:1007469218079.

[36] X. Xiong, W. Xu, and G. Zhao, "The effectiveness assessment for network based MTD strategies," in *Proc. 8th Int. Conf. Commun. Netw. Secur. (ICCNS)*, New York, NY, USA, 2018, pp. 7–11, doi: 10.1145/3290480.3290485.

XIN-LI XIONG received the B.S. degree in radar engineering and the M.S. degree in information and communication engineering from the PLA University of Science and Technology, Nanjing, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer science with the College of Command and Control Engineering, Army Engineering University of PLA, Nanjing. His research interest includes the development of moving target defense techniques, evaluation of system security, network security, and proactive defense techniques.

LIN YANG was born in Hefei, Anhui, China, in 1970. He received the B.S. and M.S. degrees in automation from the National University of Defense Technology, Changsha, Hunan, in 1995, and the Ph.D. degree in communication and electronic system from the National University of Defense Technology, in 1999.

He is currently a Senior Researcher with the System Engineering Research Institute, Academy of Military Sciences PLA, China. His research interests include computer security, information system security, network security, trusted computing, moving target defense, security protocol analysis, and big-data security.



GUANG-SHENG ZHAO received the B.S. degree in computer science from Shandong University, Shandong, China, in 2007, and the M.S. degree in computer science from the National University of Defense Technology, Hunan, China, in 2009, where he is currently pursuing the Ph.D. degree in computer science.

His research interest includes moving target defense techniques, network security, and blockchain techniques in security.