NPM: An Anti-attacking Analysis Model of the MTD system Based on Martingale Theory

Xin Yang, Hui Li*, and Han Wang
Shenzhen Key Lab of Information Theory & Future Network Arch
Future Network PKU Lab of National Major Research Infrastructure
PKU Inst.of Big Data Technology
Shenzhen Engineering Lab of Converged Networking Technology
Huawei & PKU Jointly Engineering Lab of Future Network Based on SDN
School of Electronic and Computer Engineering, Peking University, Shenzhen, 518055, China
Email: yangxin2016@pku.edu.cn, * :corresponding author lih64@pkusz.edu.cn, wanghan2017@pku.edu.cn

***Abstract*—Moving target defense (MTD) techniques are effective solutions to improve the network security by continuously reconfiguring the system setting. On the other hand, continuously transforming also increase the cost of defenders, so it is important to analyze the effectiveness of MTDs compared with their cost. Current researches lack of analyzing the effectiveness by mathematical theory compared with analyzing by experiment. Motivated by the above, we propose a novel three-dimension model named NPM jointly use N-version programming, Poisson process, Markov chain and martingale theory to analyze the effectiveness of the proposed MTD model. Our analysis points out the difficulty for a successful adversary to defeat the MTD system, which is related to the system configuration, such as the number of executors and the judgment criterion in every node, the transforming period and rang of system MTD transformation. Finally, we give advices on the design of the system in the daily defense and the attacked defense, with the goal of guaranteeing security with minimal cost.**

***Index Terms*—Moving Target Defense, Security Analysis, Martingales, Markov Chains, Dynamic Heterogenous Network, N-version programming**

I. Introduction

Network security is an evolving issue for global enterprises and individuals. The traditional security approaches contain security risks for presenting attackers a static target. In traditional networks, attackers have enough time to reconnoiter a system and plot an attack. What is worse is that attackers can keep the acquired privilege for a long time without being removed [1], [2]. In other words, the static feature of current networks makes it easy for attackers to exploit the systems' vulnerabilities and penetrate through, putting defenders in an immediately disadvantaged situation.

To combat this disadvantage, a new security approach called Moving Target Defense (MTD) [3] applied in varied domain, is rapidly developing. Based on continuously randomizing circumstances of network's configuration, this approach obviously increase the difficulty and cost of launching attacks on the adversary. More and more research have focused on how to build an MTD system, which illustrate the great concern about MTD.

In addition, the feature of MTD that transforming from time to time also increases the cost of defenders. Therefore, it is important to assess the effectiveness of MTD and find out how to protect the system with minimum costs. Existing studies about this topic can be divided into two categories: 1) testing the effectiveness with experiment, 2) analyzing the safety with mathematical model.

There are many different experimental methods for measuring the effectiveness. MTD is proposed as a system that continuously reconfigures the system setting [4]. Then in [5] Zhuang et al. compared the effectiveness between a simple MTD system and an intelligent MTD system. Hong et al. classify MTD techniques into three categories: Shuffle, Diversity, and Redundancy in [6]. Furthermore, they used a security model named Hierarchical Attack Representation Model (HARM) and importance measures (IMs) to assess the effectiveness of MTD and improve the scalability. Colbaugh et al. [7] analyzed the robustness of MTD defenses applied in email application by developing game-based models within a machine learning (ML) framework. Sengupta et al. [8] propose a method to generate a MTD switching strategy for real-world web application, and find an efficient switching strategy by modeling the system as a repeated Bayesian game.

Compared with using simulation-based approaches, there are less researches that showing the effectiveness of MTD with mathematical methods. Maleki et al. [9] introduces a Markov-model-based method to analyze the relationship between the probability of a successful attack and the cost spent by the adversary in MTD system. Wu et al. [13] describes the security of the DHR architecture and analyzing the effectiveness of cyberspace mimic defense system with Peri nets [14].

This paper introduces an MTD framework for modeling the effectiveness of MTD schemes and giving approaches to ensure safety with minimum cost. There are numerous nodes in our network with many executors in each node. For each node, MTD performs the same function with multiple executors (e.g., operating systems, variant inputs and interpreters, variant software stack components) and makes decisions based on their results. As to the network, the system setting of some nodes is reconfigured every once in a while. Then we stand in the attacker's perspective to analyze the attack process which could be divided into three steps, and deduce the relationship

978-1-5386-6950-1/18/$31.00 ©2018 IEEE

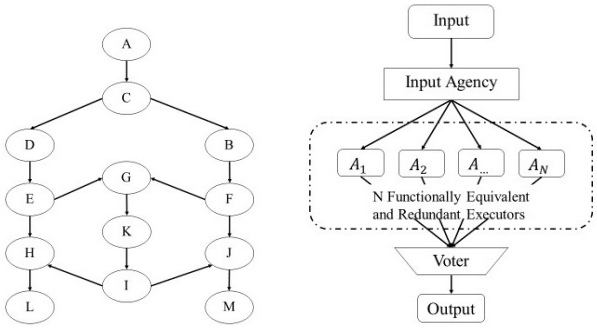
between successful attack and the system configuration. N-version programming [10], Poisson process [11], martingale theory [11] [15] are jointly used to analyze the effectiveness of the models corresponding to the three steps. Finally, we give advice on the system design based on the above results with the goal of guarantee security with minimal cost.

The rest of the paper is organized as follows. We present the MTD system model, assumption and symbols we used in Section II, followed by three-dimension security analysis model in Section III. Discussion and limitations of this paper are presented in Section IV. Finally conclusions are drawn in Section V.

II. System model and problem description

MTDs are built on the network with continuously randomizing so as to increase the difficulty and cost of attack on the adversary. We build a MTD system aiming at building a relatively secure system with the feature of dynamic, heterogeneous and redundant. In the proposed system dissimilar redundant structure is used to achieve the basic heterogeneous defense. Heterogeneous pools of every node are formed by N executors, which are functionally equivalent but implemented differently. When the system is running, the input agent assigns tasks to the set of executors (i.e., N equivalent components). N executors send their results to the multimode voter at the end of the operational cycle. The voter makes multiple-alternative decisions based on the received infomation and outputs the result.

Modules with dissimilar redundant structure between the network and the target, which are related to the attack, are regarded as attack nodes in the attack chain. The MTD architecture is shown in Figure 1.



(a) Network topology (b) The architecture of every node

Fig. 1. The MTD architecture.

In a local area network with U nodes, MTD system randomly chooses d nodes every fixed T transforming period. Then MTD system reconfigures the selected nodes and changes their heterogeneous pool with new executors. If the node being attacked encounters a reshuffle, attackers must monitor this node again to find out potential vulnerabilities to plan for a new attack, because the node information sniffed by the attacker is invalid. If the node from which the attack originates encounters a reshuffle, attackers must return to the previous node. After MTD transforming, the node setting is changed, which leds to the lost of the sniffed information, the replacement and stopping work of the defeated executors, and the cleaning on inserted backdoors, so attackers have to attack this node again. We disrupt the proliferation of attacks in the network through periodic MTD transformations.

Example 1: Here we give an example of attacking the node G. Attacks launched from the Internet are towards the Target G through the attack chain. Each attack is consisted of single step attacks from internet to A, A to C, C to B, B to F, and F to Target G. From the perspective of attackers, they need to break down the intermediate nodes A, C, B, F one by one before launching single step attacks to Target G. When node A controlled by attackers, node C becomes their new target according to the routing information from node A. Then attackers sniff the information (e.g., executors on work, and vulnerabilities in these working executors) of node C and launch attacks to node C. Firstly, when the attacker launch the single step attack to node C, they need to confuse more than M executors from N executors of node C, and they need to have them output the advantage result so that the node can be controlled. Then if the system does not reach the fixed-point of MTD transforming, the attackers can use the similar method to attack node C repeatedly according to the detected executors vulnerabilities, until the C node is compromised or attack forced back to internet because of the transformation at node A. The attack to node C can be defined as a success as long as at least one out of all the repeated attack succeeds in the MTD transforming period.Lastly, time goes to the fixed-point of MTD transforming, if the node A enconter the reshuffle, attackers go back to internet and attack the node A again. If node C has been compromised before MTD reshuffle, attackers can attack the next node B at the next time step. The attack continues until node G is compromised or the attackers are removed out the LAN.

According to the operation mechanism, the attack behavior in the proposed system can be divided into three dimensions with the concern of special event (attacks, transformation of the system). Firstly, the timeline can be sliced into many transforming periods in accordance with the system transformation. Secondly, each transforming period can be sliced into many single step attacks, because the attackers can launch attacks many times to a single node during each transforming period. The attack process can be divided into three steps: single step attacks, repeated attacks against the single node during a transforming, and attacks after MTD reshuffle. We analyze the effectiveness of MTD system in a three-dimension model named NPM: analysis the attack process jointly use N-version programming, Poisson process, Markov chain and martingale.

Several assumptions are made to simplify the analysis:

- Attackers attack one node at most during every transforming period;
- MTD transformations are only applied at a specified time interval;
- Attacks outside the link will be identified and captured by the system, so we only analyze the attacks inside the

link;
- The information of nodes is changed completely after transforming, so attackers attack this node again with the same cost.
- The probabilities of each in break against the single node are completely independent and identically distributed.

III. Security analysis

We analyze the effectiveness of MTD system in three-dimension model as follows. Firstly, our analyzing approach for the single attack process against the single node comes from the N-version programming, the details of which will be shown in the first part of this section. Secondly, every successful single step attack appears to be a small probability event over a long period of time (i.e., a system transforming period), so the number of successful attacks in every system MTD transforming period is Poisson distribution.The analysis of the repeated-attack-model will be shown in the second part of this section. Finally, the state of the system after transforming has three possible directions compared with the last period: going to the next node, going back to the last node and staying at the same node, which is only relevant to that during the present state but not to the state during the past states. In another word, the process donates states between different transforming periods is a Markov chain. Therefore, in the third part of this section, a Markov chain is set up to and transformed to Martingale to calculate the probability of the successful attack overall system during different transformation cycles.

A. Single step attacks

In an single step attack cycle, an N-version programming model is used to analyze the attack effect. N functionally equivalent redundant executors respectively obtain from the agent their input data, whose results are collected by the voter as A_1, A_2, $\cdots$, A_N. In practical applications, executors normally have different execution time with each other. Define the arrival sequence of executors' results to the voter as A_{q_1}, A_{q_2}, $\cdots$, A_{q_N}. Define the judgment criterion as follows. If the voter receives M identical results, the voter will consider them right and pass to the output device. At the same time, the executor set finishes its computing task.

The probability of the event that the executor set finishes computing right after the voter receives A_{q_j} with the output which attackers want is deduced in [10]. The probability is shown as Eq.(1).

$r(A_i)$ means the probability that the result of executor A_i is tampered with what attackers want.

Considering the independence between events of different j, the probability that the voter passes results that attackers want , regarded as a successful attack of the single node is Eq.(2).

Time required by a successful attack of the single node is a discrete random variable of parameters P_j and $t(q_j)$, whose conditional expectation is

$$E[t_A] = \frac{\sum_{j=M}^{N} P_j t(q_j)}{\sum_{j=M}^{N} P_j}, \tag{3}$$

where $t(q_j)$ means the average execution time of executor A_{q_j}.

B. Repeated attacks against one node before transforming

As same as in the static network, attackers can launch multiple attacks to a node in every system transforming period. It is stochastic whether the attack succeeds with p_A probability. According to the analysis in the previous section, p_A is much smaller than the transforming period, so successful attack is a small probability event over a long period. As a result, the number of successful attacks is Poisson distribution. In this section, we calculate the probability of an attacker defeating a single node during the transforming period.

Note the transforming period as T. The attacker launches attacks consecutively, and the shortest interval is denoted by $\triangle\ t$. In every period, the attacker launches $\frac{T}{\triangle t}$ attacks at most. According to the previous section, it takes t_A at least to attack a single node, so the maximum number of valid attacks is $\frac{T-t_A}{\triangle t}$. From the previous section, the probability of attacking a single node is p_A. Easy to infer, the number of successful attacks every period is $\frac{T-t_A}{\triangle t(T-t_A)} p_A = \frac{p_A}{\triangle t}$, in another word, the number of successful attacks is approximate Poisson distributed with the parameter $\lambda = \frac{p_A}{\triangle t}$.

Note the number of the compromised nodes since the start of this cycle until the time t as $No(t)$, where $0 \leqslant t \leqslant T$. According to the previous section, the attacks from t_A at the end of the cycle can not be successful, so $No(t), 0 \leqslant t \leqslant T - t_A$ is Poisson distributed.

That is for all $t, s \geqslant 0$,

$$P\{No(t+s) - No(s) = n\} = e^{-\frac{p_A}{\triangle t} \cdot t} \frac{(\frac{p_A}{\triangle t} t)^n}{n!}. \tag{4}$$

We have the probability of compromising no node is:

$$P\{No(T) - No(t_A) = 0\} = e^{-\frac{(T-t_A)}{\triangle t} p_A}. \tag{5}$$

Hence, the probability of successfully attacking one node in a transforming period is:

$$\mu = 1 - P\{No(T) - No(t_A) = 0\} = 1 - e^{-\frac{(T-t_A)}{\triangle t} p_A}. \tag{6}$$

C. The attack process after MTD transforming

As the previous analysis, we adopt Markov chains to analyze the transforming process. The probability of attacking a single node successfully is μ, if the configuration manager does not reconfigure the node attacking organized from during transform. There are W nodes in the attack chain and U nods in the LAN. The defender chooses d nodes from U nodes to reconfigure their system setting in every transforming period with ω probability of reshuffling this node. Assuming that the

$$P_j = r(q_j)\prod_{i=1}^{j-1}(1-(q_i))[\sum_{i_1=1}^{j-M+1}\frac{r(q_{i_1})}{1-r(q_{i_1})}\sum_{i_2=i_1+1}^{j-M+2}\frac{r(q_{i_2})}{1-r(q_{i_2})}\cdots\sum_{i_{M-1}=i_{M-2}+1}^{j-1}\frac{r(q_{i_{M-1}})}{1-r(q_{i_{M-1}})}] \tag{1}$$

$$p_A = \sum_{1}^{N}P_j = \sum_{1}^{N}r(q_j)\prod_{i=1}^{j-1}(1-(q_i))[\sum_{i_1=1}^{j-M+1}\frac{r(q_{i_1})}{1-r(q_{i_1})}\sum_{i_2=i_1+1}^{j-M+2}\frac{r(q_{i_2})}{1-r(q_{i_2})}\cdots\sum_{i_{M-1}=i_{M-2}+1}^{j-1}\frac{r(q_{i_{M-1}})}{1-r(q_{i_{M-1}})}] \tag{2}$$

attack now stays at the i_{th} node (i.e., the attacker has attacked i nodes successfully), the Markov chain of attacking is shown as fig.2:

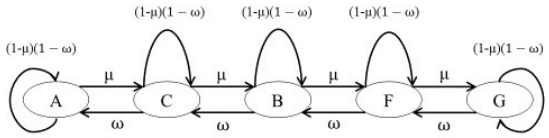


Fig. 2. Markov chain of attacking

We describe the attacker transfer with a $M^{W\times W}$ matrix where the W columns (and rows) denote the W nodes in the attack chain. The cell $M_{i,j}$ denotes the transfer probability from the i_{th} node to the j_{th} node. The successful state of attackers is represented by W (i.e., node G shown in the example). And the state 0, 1,$\cdots$, $W-1$, correspond to the different position the attack at after the last transformation. Attackers launch attacks to the nodes one by one along the attack chain. The attack have three possible moving directions after transformation: going to the next node, going back to the last node, or staying at the same node. Correspondingly, the transforming probability is shown as follows:

- $M_{i,i-1} = \omega$
  The node from which the attack originates encounters the reshuffle with probability ω. Thus the attack can not carry on, and attackers must return to the previous node, once the node from which the attack originates encounters the transformation, i.e. $M_{i,i-1} = \omega$.
- $M_{i,i+1} = (1-\omega)\mu$
  No transformation has occurred before the attacker successfully attack the node i, so the attack move to the next node (i.e., node $i+1$). The probability of the situation mentioned above is $M_{i,i+1} = (1-\omega)\mu$, with probability μ the attacker launchs a successful attack and probability $(1-\omega)$ meaning that the node from which the attack originates does not encounter the transformation.
- $M_{i,i} = (1-\omega)(1-\mu)$
  The adversary does not attack the next node successfully, and no effective MTD transformation is done, with probability $(1-\mu)$ and $(1-\omega)$ respectively.

Firstly, the probability of the node from which the attack originates being reconfigured can be expressed as follows:

$$\omega = \frac{\binom{U-1}{d-1}}{\binom{U}{d}} = \frac{d}{U} \tag{7}$$

Let $X_0, X_1, X_2, , X_n$ be random variables, where X_i denotes the node position at which the attack stays after the beginning of the i_{th} transforming period. Every element has a value range of $[0, W]$, and $X_0 = 0$ indicating that the initial position of the attack is the position of entering the attack chain. When the position at time n is k, the state at next period is inferred as follows:

$$P\{X_{n+1} = k+1 | X_n = k\} = (1-\omega)\mu, \tag{8}$$

$$P\{X_{n+1} = k | X_n = k\} = (1-\omega)(1-\mu), \tag{9}$$

$$P\{X_{n+1} = k-1 | X_n = k\} = \omega. \tag{10}$$

Hence

$$E[X_{n+1} | X_n = k] = k + (1-\omega)\mu - \omega. \tag{11}$$

Theorem 1: Let $M_0, M_1, M_2, \cdots, M_n$ be independent random variables, where $M_i = X_i - [(1-\omega)\mu - \omega]\cdot i$,
i.e.

$$M_n = X_n - [(1-\omega)\mu - \omega]n, \tag{12}$$

$$M_{n+1} = X_{n+1} - [(1-\omega)\mu - \omega](n+1). \tag{13}$$

Then the sequence M_n is a martingale with respect to $X_0, X_1, X_2, \cdots, X_n$

Proof:

$$\begin{aligned} E[M_{n+1} | X_0, X_1, X_2, \cdots, X_n] &= E[M_{n+1} | X_n] \\ &= E[X_{n+1} - [(1-\omega)\mu - \omega](n+1) | X_n] \\ &= E[X_{n+1} | X_n] - [(1-\omega)\mu - \omega](n+1) \\ &= X_n + (1-\omega)\mu - \omega - [(1-\omega)\mu - \omega](n+1) \\ &= X_n - [(1-\omega)\mu - \omega]n \\ &= M_n. \end{aligned} \tag{14}$$

In order to derive the number of steps to attack the node W, we introduce the martingale stopping-time theorem (i.e., Lemma 1), which is presented the proof in [11]

Lemma 1: when S is a stopping time , and satisfy:
A. $P\{S < \infty\} = 1$;
B. $E[|M_S|] < \infty$;
C. $lim_{n\to\infty} E[|M_S| I_{\{S>n\}}] = 0$,

Thus

$$EM_S = EM_0. \tag{15}$$

Theorem 2: For a MTD game with the probability μ attacking a single node successfully and probability ω performing MTD transformation at this node. And there are W nodes in the attack chain with transforming period T. The expected time until the attackers win the game (i.e., attacking the node W successfully) is $E[T_A] = \frac{WT}{[(1-\omega)-\omega]}$.

Proof: The condition of arrival time S is $X_S = W$. Obviously, whether n is equal to S is easily known based on the result of the previous n rounds, so the time S is the stopping time of martingale. According to the property of Markov chain [12], there exist $c < \infty$, $\rho < 1$, s.t.

$$I\{S > n\} \leq c\rho^n. \tag{16}$$

Notice that $|M_n| = |X_n - [(1-\omega)\mu - \omega]n| \leq W + n$, and thus

$$E[|M_S|] \leq W + S \leq \infty, \tag{17}$$

$$E[|M_S|I_{\{S>n\}}] \leq c\rho^n(W+n), \tag{18}$$

$$lim_{n\to\infty}E[|M_S|I_{\{S>n\}}] \leq lim_{n\to\infty}c\rho^n(W+n) = 0. \tag{19}$$

$lim_{n\to\infty}E[|M_S|I_{\{S>n\}}] \geq 0$, so it satisfies the lemma 1. The number of steps required to reach node W can be calculated based on the lemma 1:$EM_S = EM_0 = EX_0 = 0$.

$$\begin{aligned} E[M_S] &= E[X_S - [(1-\omega)\mu - \omega]S] \\ &= E[X_S] - [(1-\omega)\mu - \omega]E[S] \\ &= 0, \end{aligned} \tag{20}$$

and

$$E[M_S] = W, \tag{21}$$

thus

$$W - [(1-\omega)\mu - \omega]E[S] = 0, \tag{22}$$

$$E[S] = \frac{W}{[(1-\omega)\mu - \omega]}, \tag{23}$$

$$E[T_A] = \frac{WT}{[(1-\omega)\mu - \omega]}. \tag{24}$$

D. Conclusion

For given system parameters: W, U, d, T, t_A, $\triangle\, t$, M, N and $r(q_i)$, we have the time $E[T_A]$ to break the system

$$E[T_A] = \frac{WT}{(1-\frac{d}{U})[1 - e^{-\frac{(T-t_A)}{\triangle t}(\sum_{j=M}^{N} P_j)}] - \frac{d}{U}}, \tag{25}$$

where, $P_j$is shown in Eq.(1).

In the following section, we analyze each of these parameters, and give advice on the design of the system, with the goal of guaranteeing security with minimal cost.

IV. Simulation result and analysis

The security of our system is related to parameters in Eq.(25). However, numerous parameters are beyond control, for example, the difficulty of tampering with executor ω_i and the length of the attack chain. There are also many parameters having obvious impact on the system security such as the security of a single node. So we analyze the conclusion with giving some parameters, as well as jointly analyzing other parameters.

We develop the two-step simulation to verify the effect of each parameter on the system safety and look for a relatively efficient configuration. We investigate the most secure configuration of every node on the first step by adjusting the number of executors and the judgment criterion (i.e., M). Then we look for the most secure link configuration in the second step by adjusting the time period and range of reshuffle in the LAN.

We initialize the uncontrolled elements in the network with assuming that the number of nodes in the LAN is $U = 500$, the attack chain size is $W = 10$. There are 15 different executors. For every executor, the attack time and the probability of successful attack with the arrival sequence of executors' results to the voter as $A_{q_1}, A_{q_2}, \cdots, A_{q_N}$ are as Table 1.

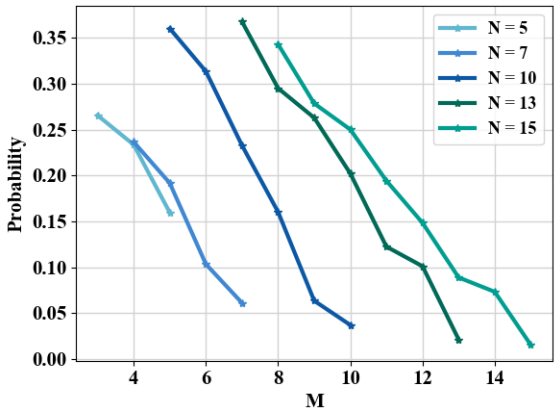
1) Single step attack: Observe the Eq.(25), it is easy to find that N and M should be adjusted to make p_A as small as possible and $E[t_A]$ as large as possible. We choose the $1_{th}, 5_{th}, 9_{th}, 10_{th}, 14_{th}$ value from the 15 executors for $N = 5$, and the other value for $N = 10$ and the whole value for $N = 15$. Similarly, we choose values for $N = 7$ and $N = 13$ with control for the average time and the average probability.If $M < 1/2N$, the correctness of the system output is unable to judge, so we choose $M \geq 1/2N$.

We compute the single step attack success rate and attack time corresponding to different N and M values. Figure 3 shows the change of attack success rates and attack times with different N and M.

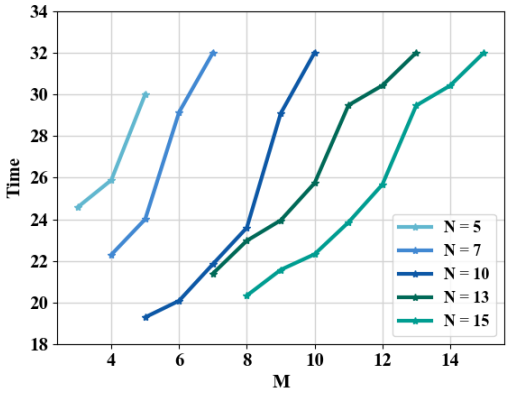
As shown in fig.3 (a), we can reduce the probability of compromising single node by select the M and N to share the pressure on the entire system. The difficulty of compromising a single node increases along with M, and the value of N reduces the minimum success rate. In the perspective of defenders, the large M and N mean huge overhead. With the same attack success rate, we would like to build the system with the smaller M and N. We can use MTD transformation

TABLE I
15 EXECUTORS WITH THEIR RESPECTIVE ATTACK TIME AND THE PROBABILITIES OF BEING SUCCESSFULLY ATTACKED

q_j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$r(q_j)$	0.2	0.3	0.4	0.1	0.2	0.4	0.11	0.2	0.1	0.21	0.2	0.3	0.1	0.4	0.15
$t(q_j)/(min)$	3	4	6	9	11	11	14	15	15	17	18	20	25	30	32



(a) Attack time map



(b) Probability map

Fig. 3. Different effects brought by M and N.

to protect the system, so we need not select the biggest N and M. For example we can select a middle $M = 10$, and $N = 8$ with 0.160277 probability of compromising a single node and $t_A = 23.6004$

2) The security of network: There are two ways to prevent the spread of attacks: increasing the frequency of transformations and increasing the range of transformations. We can reduce T or increase ω, both of which mean the consumption of resources. Therefore, we make them work together to find the minimum cost while ensuring safety. Base on the result of the first step analysis, we take $\triangle t = 1$ minute, and then take $t_A = 23.6004$ minutes and $P_A = 0.160277$, for the Eq.(25).

Figure 4 shows the change of the successfully attack time with different T and $\omega = \frac{d}{U}$.

The transformation period $T > t_A$, because it is possible to make transformation when the arbiter has not received enough execution results if the period is too short, which disturbs the normal function. Based on the Eq. (25), for T larger than t_A, the probability of compromising one node increase and the attack time decrease with the increase of T. If the T is bigger, the probability of compromising one node in Eq. (25) (i.e., $[1 - e^{-\frac{(T-t_A)}{\Delta t} * P_A}]$) tends to be 1 as T increases. At this moment, the T of the molecular part of Eq.(25) appears to be more influential. In this case, the attack succeeds with fewer steps, but long succeed time because of the long period of transformation. Obviously, this is unreasonable and departs from our assumption, so the value of T should be close to t_A.

Observing Eq. (25), ω denotes the up-going probability of the system. Starting from 0, the theoretical system failure time increases as ω increasing. As ω approaches $\mu/(\mu + 1)$, it reaches infinity. When it exceeds $\mu/(\mu + 1)$, attack time becomes negative, and increases from negative infinity to 0 with ω increasing. At this moment, the downlink probability of the attack in the attack chain is less than the uplink probability because of the large ω. As a result the attack can not only go down the attack chain, but also be moved upwards because of transformation. The attacker is further and further away from the attack target, and is cleared out of the attack chain finally. The negative value means the time that attack being cleared out W steps along the attack chain takes. The larger the ω is, the shorter time to make attack back to W steps.

In summary, we take $23 < T < 40$, and $0 < \omega < 0.55$. Besides, we only intercept the part between $[-2500, 3500]$ for discussion because there will be a critical value to attack time tends to infinity.

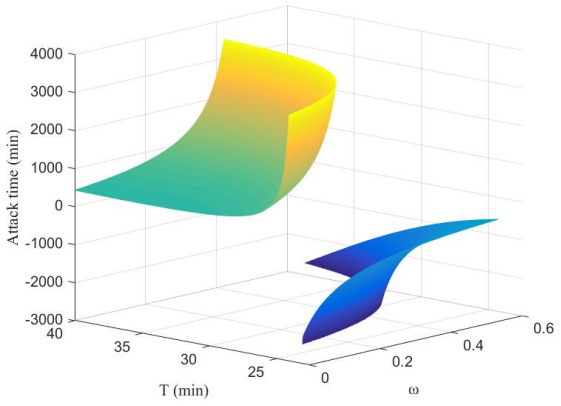


Fig. 4. Attack time with T-ω

Our goal is to ensure safety under the premise of reducing

overhead, which means ensuring long enough attack under the premise of reducing the conversion rate and transform the scope. In another word, defenders need a high attack time with big T and small ω.

We obtain the top view (shown in Figure 5) of the figure 4 above by rotating. The green and yellow part means the positive dot representing attack time. The closer the dot is to the curve edge, the more taken overhead and theoretical break time it represents. Under the same security, the larger T is, the larger ω is, so there is a trade-off between T and ω. We can set the attack time according to system requirements, and then select the T, ω.

The absolute values of the positive (negative) values in Figure 5 respectively represent the time it takes for the attack to move uplink (downlink) W step. In usual defense, we can select the T, ω value from the yellow part to ensure a long time success attack. When there are attacks been detected, we can choose the value of T, ω value from the blue part to clear attacks out of the system.

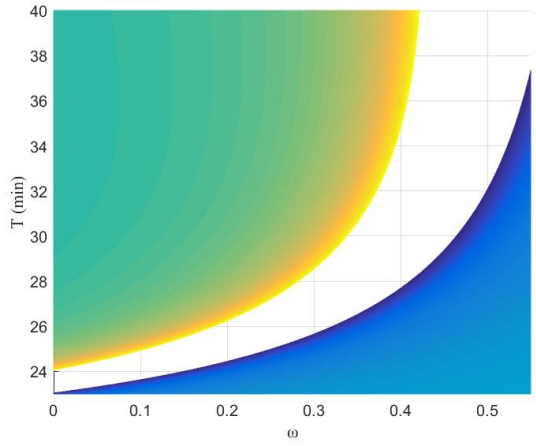


Fig. 5. Attack time with T-ω (the top view)

For example, when there is no attack, we take the attack time as 2000 mins, which is the edge of the yellow part of the curve. Then the function of the curve is:

$$\omega = \frac{1 - e^{-(T-23.6004)*0.160277} - T/200}{2 - e^{-(T-23.6004)*0.160277}}. \quad (26)$$

After obtaining the analytic formula, the system can flexibly adjust T, ω values according to their design requirements such as $T = 30$ and $\omega = 0.2994$, and with Eq.(7) $d = 90$.

When there are attacks, we take the attack time from the blue part as -1000, which means attacks are pushed away 10 nodes within 1000 minues, then the function of the curve is:

$$\omega = \frac{1 - e^{-(T-23.6004)*0.160277} + T/100}{2 - e^{-(T-23.6004)*0.160277}} \quad (27)$$

At this time if we want to increase the defense, a balanced approach is to reduce the conversion cycle while increasing the transformation range, such as $T = 25$, $\omega = 0.3755$, and $d = 112$.

Above all, T, ω, N, M can be chosen based on the above analysis and the attack time we want.

Conclusion

This paper proves an upgrade MTD system, and divides an attack process into three stages: attacking once against a single node, attacking in a transformation period against a single node, and attacking along the network. Corresponding to the three stages above, we build a three-dimension model named NPM with N-version programming, Poisson process, Markov chain and martingale to analyze the effectiveness of MTD. As a result, we make a description of relationship between the security and cost of the MTD system. For the system designers, we give advice on how to adjust the system configuration making the system reliable with the minimum cost in the daily defense and the attacked defense.

Acknowledgment

This work is supported by the National Keystone R&D Program of China (No. 2017YFB0803204, 2016YFB0800101), Natural Science Foundation of China (NSFC) (No. 61671001), GuangDong Key Program (GD2016B030305005), Shenzhen Research Programs (JSGG20150331101736052, ZDSYS201603311739428, JCYJ20170306092030521).

References

[1] Mandiant Intelligence Center, "APT1: Exposing one of China's cyber espionage units," Mandiant, Tech. Rep., 2013.
[2] D. Barrett, "Hackers penetrate NASDAQ computers," http://online.wsj.com/article/, Feb. 2011.
[3] Jajodia S, Ghosh A K, Swarup V, et al." Moving Target Defense[M]." Springer New York, 2011.
[4] R. Zhuang, S. Zhang, S. DeLoach, X. Ou, and A. Singhal, "Simulation-based Approaches to Studying Effectiveness of Moving-Target Network Defense," in Proc. of National Symposium on Moving Target Research, 2012.
[5] R. Zhuang, S. Zhang, A. Bardas, S. DeLoach, X. Ou, and A. Singhal, "Investigating the Application of Moving Target Defenses to Network Security," in Proc. of the 6th International Symposium on Resilient Control Systems (ISRCS 2013), 2013, pp. 162169.
[6] Hong J B, Dong S K. "Assessing the Effectiveness of Moving Target Defenses Using Security Models[J]." IEEE Transactions on Dependable & Secure Computing, 2016, 13(2):163-177.
[7] Colbaugh, Richard, Kristin, "Predictive Moving Target Defense."
[8] Kambhampati S, Kambhampati S, Kambhampati S, et al. "Moving Target Defense for Web Applications using Bayesian Stackelberg Games:" (Extended Abstract)[C]// International Conference on Autonomous Agents & Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2016:1377-1378.
[9] Maleki H, Valizadeh S, Koch W, et al. "Markov Modeling of Moving Target Defense Games"[C]// ACM Workshop on Moving Target Defense. ACM, 2016:81-92.
[10] Levitin G. "Optimal structure of fault-tolerant software systems." Reliability Engineering & System Safety, vol. 89, no. 3, pp.286-295, 2005.
[11] Ross S M. "Stochastic Processes[M]." Wiley, 1983.
[12] Serfozo R. "Basics of Applied Stochastic Processes[M]"// Basics of applied stochastic processes /. Springer, 2009:xiv+443.
[13] WU Jiangxing,"Research on Cyber Mimic Defense, Journal of Cyber Security,Vol.1,No.4, Oct,2016.
[14] German, Reinhard. "Markov regenerative stochastic Petri nets with general execution policies: supplementary variable analysis and a prototype tool." Performance Evaluation 39.14(2000):165-188.
[15] Li, Shuo Yen Robert. "A Martingale Approach to the Study of Occurrence of Sequence Patterns in Repeated Experiments." Annals of Probability 8.6(1980):1171-1176.