The evaluation of performance cost for network based moving target defense

Xinli Xiong1, 3, Kechang Li1 and Guangsheng Zhao2

1 Army Engineering University of PLA, Nanjing, China
2 National University of Defense Technology, Changsha, China
3 E-mail: xxlyx25@hotmail.com

Abstract. Network based moving target defense prevents attackers from reliably contacting a system by re-configuring network factors. Although limited researching has demonstrated it to be practical and feasible, little studies have been conducted to evaluate the performance cost appended to the original system. Without quantitative results, it is challenging to implement moving target techniques to typical information system broadly. This paper introduces a Queueing Petri Net model that can provide insight into the performance of network re-configurations. This model quantifies the system throughput, queue utilization and response time of requests regarding configurations of the information system, type of strategies for mutation and the frequency of networking transformation. Through logical deductions and simulations using Queueing Petri net Modelling Environment, results show that the network shifting method is an affordable defense technology compared to the performance loss in the processing of servers. Moreover, event-based strategies can get the trade-off between security and performance. These results guide the research deep into developing a more optimized method in moving target defense.

1. Introduction

The static and less-reconfigurable architecture of networks creates a fundamental asymmetry between adversaries and defenders. Network based Moving Target Defense (NMTD) continuously transforms network factors to increase efforts of reconnaissance and reduce the window of successful launching attacks [1].

Characteristics of moving target (MT) technologies in the network are dynamism and randomization against sniffers, hit-list worms, and DDoS attack, which have been investigated in several works of literature. NMTD technologies in IP address, such as RHM (Random Host Mutation) [2] is working on IP shifting schemes, and TPAH (TAP-based Port and Address Hopping) [3] is interested in randomization of ports, service hopping [4] is focused on mixed communication parameters. All those methods focus on dynamically randomizing network configurations to confuse adversaries, interrupt the cyber-attack chain and thwart probing attacks.

Many researchers make impressive contributions in evaluating the effectiveness of NMTD using Urn model [5] and Balls into Bins model [6], which gives quantitative results for several NMTD scenarios. However, the performance cost of MTD brought to the original system are deficient compared to other researches [7]. To our best knowledge, there are few works in evaluating the performance cost, and previous researches are focused on specific MTD techniques. Queueing theory is used to model Host-based MTD techniques such as platform rotation and self-cleaning [8]. By modeling the host resource with a limited reconfigurable resource, Connell et al. [8] added a
reconfigurable model to the queueing model of the primary system and provided a SimPy based simulation method to evaluate. However, this model is hard to extend to NMTD. Emulation methods taking for SCADA system are complicated and expensive, Leeuwen et al. [9] presented a test-bed with physical and virtual nodes to evaluate the performance cost by agents, which is difficult to provide timely evaluations.

Queueing Networks (QNs) and Petri Nets (PNs) are well-known formalism methods for the description and analysis in a variety of systems. PNs are used for qualitative analysis whereas QNs are employed for performance evaluation (quantitative analysis). From a practical point of view, system analysis should be pursued concerning both qualitative and quantitative aspects. Queueing Petri Nets (QPN) and Hierarchically-Combined Queueing Petri Nets (HQPN) [10] were introduced to make descriptions of complex systems easier, combined formalisms for the integrated specification of qualitative and quantitative aspects of a system also leads to new possibilities in the analysis. The widely-used software toolkits such as Queueing Petri nets Modeling Environment (QPME) [11] can be expediently employed for our simulations and performance analysis.

This paper introduces a place-transition model with queueing-service places to describe the performance cost of NMTD under various proactive strategies. QPN models are applied to provide a basic model for a typical information system as well as NMTD executions. These strategies are classified into two types, time-based and event-based, and two QPN models are presented to cover the difference. Evaluations are simulated by QPME, which provides the necessary tools to demonstrate effects that the frequency of shuffling, processing capability of information systems, and diverse type of strategies have on the system throughput, queue utilization of servers and response time of requests.

The contribution of this paper is introducing a QPN model to investigate the performance cost of network shifting methods in defending information system. The model gives an intuitional description of NMTD implementations and can be extended to multi-layered MT technologies. Results show that frequent mutations in the network influence the performance of a system in throughput and utilization. Different from the fixed periodic transformation in time-based strategies of NMTD, event-based strategies get the trade-off between security and consumption. Moreover, these results also investigate that costs brought by NMTD do not play a leading role compared to the bottleneck of processing in servers. This work is novel as related work performs a quantitative analysis of performance changes in the typical information system defending with network-based MT techniques.

This paper is organized as follows. In Section II, we discuss principles for network based moving target defense which describes various strategies, different implementations, and potential performance costs. In Section III, we introduce the model of Queueing Petri Nets for performance analyzing and use this model to evaluate the performance cost of NMTD quantitatively. Through QPME, a QPN editor and simulator, a case study and results analysis are given in Section IV. Section V summarizes the critical point of this paper and introduces some potential future works.

2. Principles of network based MTD
The foundation of NMTD is applying a virtual identity tag to replace the original network identity tag, such as IP addresses and port numbers, for connection and identification. Those virtual identity tags can be similar to original network tags or substituted by self-defined tags. Adopting diverse proactive strategies, NMTD can switch virtual identity tags from time to time, which provides dynamics and randomness to networks. To maintain availability, NMTD employs a synchronization mechanism for virtual identity tags updating applied in the centralized node or both sides of distributed nodes, and a translation mechanism for virtual/original tags converting provides a transparent channel for applications in the up-layer.

2.1. Proactive strategies in NMTD
In NMTD, proactive strategies of defenders refer to the method that transforms network configurations into several statuses at a proper time. Since the honeypot does not change the status of networks, strategies in disguise are not considering as proactive strategies in this paper. In IP address shifting
techniques, the strategy decides when to move the host from one IP address to another and how to choose the next available address. Therefore, NMTD can be modeled as a host $S$ transforms its status from $n_i$ to $n_{i+1}$ in the possible configurable space (PCS) and denoted as $S(n_{i+1})$, which is shown in Figure 1.

$$S(n_{i+1}) = R(S(n_i), t, \omega, \mu) \quad (1)$$

where $R(S, t, \omega, \mu)$ is the transition function changing network configurations of a host $S$ in PCS $\omega$ at the time $t$ with a randomness generation method $\mu$.

![Figure 1. Basic principles of network-based MTD.](image)

From Figure 1, several hosts connected by clients for different demands are continuously changing their configurations in networks. The attacker can detect and grasp one of the possible statuses for hosts after conducting some probes. After the status shifted by MT techniques, prior efforts of the adversary are invalid and a new round of probing need to be restarted. Moreover, the shifting time $t$ can be set by time-based strategies, specifically the mutation period $\tau$. However, in event-based strategies, the shifting time $t$ is entirely determined by security-related events, such as alerts and audits from IDS.

2.2. Performance cost in NMTD

Deployed network-based MTD technologies, the traditional information system must include additional modules to process, which leads to potential performance reductions. Apart from the additional processing time, the mechanism of synchronization and translation also introduce performance cost to the information system to ensure that legitimate users can finish their operations expectedly and transparently. Necessary modules of NMTD is shown in Figure 2.

As depicted in Figure 2, the primary process of NMTD can be divided into four parts. The Translation completes basic operations, which processes the virtual-to-original translation to ensure that both clients and servers are connected transparently. The Synchronization updates network parameters, which sends the network configuration to legitimate clients after mutations of servers. If there is a controller, as shown in dashed lines, clients and servers must connect to the controller to authenticate and get the configuration of NMTD. The Mutation conducts the shifting of network configurations for servers, and Reconnection assists clients to re-build connections after the synchronization with servers.
To evaluate the performance cost of NMTD in different implementations, our approach catalogs the main cost into two parts, Translation, and Mutation-Synchronization-Reconnection. The former is the regular cost among every connection through NMTD, which can be viewed as an additional process of network transmissions. The latter is the primary cost when servers mutate network configurations, which forces clients to stop on-going transmissions and reduces the total throughput of information systems. Various implementations of NMTD are not only determined the performance cost brought by Mutation-Synchronization-Reconnection but also affected by diverse strategies of NMTD. In this paper, we introduce a QPN model to give qualitative and quantitative analyses.

3. Evaluating performance cost in NMTD using Queueing Petri Nets

The main idea of the QPN and HQPN is to extend queueing and timing aspects to the places of Colored Generalized Stochastic Petri Nets (CGSPNs) [12]. A QPN is defined as an 8-tuple and can be denoted as:

$$QPN = (P,T,C,I^-,I^+,M_0,Q,W)$$

where $P = \{p_1,p_2,\cdots,p_n\}$ is a finite and non-empty set of places; $T = \{t_1,t_2,\cdots,t_m\}$ is a finite and non-empty set of transitions, $P \cap T = \emptyset$; $C$ is a color function that assigns a finite and non-empty set of colors to each place, and a finite and non-empty set of modes to each transition; $I^-$ and $I^+$ are the backward and forward incidence functions defined on $P \times T$; $M_0$ is a function defined on $P$ describing the initial marking such that $M_0(p) \in C(p)_{MS}$; $Q = (\hat{Q}_1,\hat{Q}_2,\cdots,\hat{Q}_p)$, $\hat{Q}_i \subseteq P$ is the set of timed queueing places, $\hat{Q}_2 \subseteq P$ is the set of immediate queueing places, $\hat{Q}_1 \cap \hat{Q}_2 = \emptyset$, and $q_i$ denotes the description of a queue taking all colors of $C(p_i)$ into consideration; $W = (\hat{W}_1,\hat{W}_2,\cdots,\hat{W}_p)$, $\hat{W}_i \subseteq T$ is the set of timed queueing transitions, $\hat{W}_2 \subseteq T$ is the set of immediate queueing transitions, $\hat{W}_1 \cap \hat{W}_2 = \emptyset$, $\hat{W}_1 \cup \hat{W}_2 = T$, $w_i \in C(t_i \rightarrow \mathbb{R}^+)$ such that $\forall c \in C(t_i) \cap w_i(c) \in \mathbb{R}^+$ is interpreted as a rate of a negative exponential distribution specifying the firing delay due to color $c$. Also, a basic QPN model depicted with notions in QPME is shown in Figure 3.

---

**Figure 2.** Basic modules in NTMD.
As depicted in Figure 3, tokens, when fired into a queueing place by any of its input transitions, are inserted into the queue according to the queue’s scheduling strategy. Tokens in the queue are not available for output transitions of the place. After completion of its service, a token is immediately moved to the depository, where it becomes available for output transitions of the place. Moreover, the subnet of an HQPN has a dedicated input and output place, which are ordinary places of a QPN. Tokens being inserted into a subnet place after a transition firing are added to the input place of corresponding HQPN subnet. Every HQPN subnet also contains an actual-population place used to keep track of the total number of tokens fired into the subnet place. Each transition includes incidence functions to describe the firing rules of tokens.

3.1. Performance analysis of typical information system using QPN
A typical information system can be modeled as three significant parts, clients among the Internet, web server(s) and application server(s) in the front and database server(s) in the back [13]. These clients send several requests to the system, and servers in the front, Front End (FE), process those requests according to the application. When requests of clients need to manipulate data, FEs establish connections with database servers in the back, BE (Back End), to CRUD data and send results back to the clients through FEs. Modeling the typical information system in QPN, Client is modeled by Infinite Server (IS) queue, FEs are modeled by Processor Sharing (PS) queue, and BEs are modeled by using the First Come First Service (FCFS) queue. The basic model of a typical information system using QPN (TISQPN) in QPME is shown in Figure 4.

As described in Figure 4, the TISQPN model depicts a classical C/S architecture that FEs hold the logical processing of clients and BEs are used for persisting data, which is widely applied in E-Businesses, Office Automation, and SCADA system. Also, the places and tokens in the model are demonstrated as followed.

The following types of places are used in the TISQPN model:

- Clients Queueing place with IS scheduling strategy represents clients sending requests to the system.
- FE-CPU-n Queueing place with PS scheduling strategy represents the CPU of FEs, where n stands for the mark of CPU using in FEs.
- BE-1/O Queueing place with FCFS scheduling strategy represents the disk subsystem of BEs.
- Thread pool Ordinary place represents the thread pool of FEs.
- Connection pool Ordinary place represents the database connection pool of BEs.
• FE and BE Ordinary places use to hold incoming requests for FEs and BEs.

The following types of tokens are used in the TISQPN model:
• r represents a request sent by a client for execution, which is contained only in places Client, FE-CPU-n, BE-I/O, FE and BE.
• t represents an FE thread, which can be contained only in the place Thread pool.
• c represents a BE connection which can be contained only in the place Connection pool.

The life-cycle of a client request in TISQPN model is demonstrated as followed. Every request is initially at the queue of place Client where it waits for a user-specified interval to partition requests. After elapses of intervals, the request moves to the Client depository and transmits to the place FE waiting for a thread to be allocated to it before its processing can begin. Once a thread is allocated, the request moves to the queuing place \( FE-CPU-n \), where it receives service from the CPU of FEs. Then, it moves to the depository of the place, which links to place BE and waits for a database connection to be allocated to it. The database connection is used to connect to the database servers and make any CRUD required by the respective transaction. Once the connection is available, the request receives service at the place BE-I/O, which completes the processing of the request, and then sent back to place Client releasing the held thread of FE and the connection of database.

![Figure 4. Model of typical information system in QPN.](image)

3.2. Performance analysis of NMTD using QPN
Adding a QPN model of NMTD to the system, our approach is able to measure the performance cost of NMTD including different implementation in a typical information system. Simplifying the QPN model to avoid the problem of state explosion and cover various implementations, the QPN model of NMTD can be divided into two major parts, Translation of NMTD and Transformation of NMTD, which extends the TISQPN to NMTD-TISQPN. In the model of Translation of NMTD, modules for maintaining transparent connections and other pre-conditional operations are included, which can be modeled by a PS queue. The later, Transformation of NMTD, contains Mutation-Synchronization-Reconnection processes to ensure that clients maintain connections with servers in FEs, which can be described as an FCFS queue.

Meanwhile, performance costs bring by different proactive strategies are modeled by two methods. Time-based strategies are modeled by transition incidence functions, which firing weight is using as the period of mutation. However, event-based strategies are modeled by a generator of event-token, which generates event-tokens according to prior knowledge or pre-set distribution. The model of NMTD in QPME is shown in Figure 5.
As shown in Figure 5, the model of NMTD is adding to the typical information system as a pre-processing. Moreover, the places and tokens in the model are demonstrated as followed.

The following types of places and specific transitions are used in the NMTD-TISQPN model:
- **NMTD_n** Ordinal place represents the *Translation* of NMTD.
- **NMTD_m** Queuing place with FCFS scheduling strategy represents the *Transformation* of NMTD.
- **NMTD_e** Queuing place with normal distribution represents the generator of tokens for event-based NMTD strategies.
- **NMTD_redo** Ordinal place represents the redo-method of NMTD to regain tokens.
- **NMTD_in1** and **NMTD_in2** Ordinal place represents the input of NMTD services.
- **NMTD_rc** Ordinary place represents the mutual exclusion method of translation and transformation in NMTD.
- **NMTD_out** Ordinary place represents the output of NMTD services.
- **t_{\text{in}}** Transition has the incidence functions representing time-based strategies of NMTD.
- **t_{\text{rodo}}/t_{\text{1}}** Transition that links to a typical information system.

The following types of tokens are used in the NMTD-TISQPN model:
- **n** represents an execution of NMTD which can be contained in places NMTD_n, NMTD_m, NMTD_redo, NMTD_in1, NMTD_in2, and NMTD_out.
- **s** represents a mutual exclusion of execution in NMTD, which can be contained only in place NMTD_rc.
- **e** represents a mutation of NMTD triggered by security-related incidents, which can be contained only in place NMTD_e.

The procedure of network mutation in NMTD-TISQPN model is described as followed. As a pre-processing to the typical information system, the NMTD_in1 and NMTD_in2 in the NMTD-TISQPN model hold the initial token *n* for an execution. To ensure that *Translation* of NMTD and *Transformation* of NMTD are mutually exclusive, the NMTD_rc holds the initial token *s* for a selection. Both *n* and *s* are tokens that send to NMTD_n and NMTD_m from the very beginning, and when the request from clients arrive at the FE, it waits for both the thread of FE and the execution of NMTD. To demonstrate diverse strategies of NMTD, only when tokens in *t_{\text{no}}* meet the firing condition of incidence function or the NMTD_e sends the event-token, the token *n* enters the FCFS queue of NMTD_m and leaves from the NMTD_out representing a mutation of NMTD. In other conditions, the

![Figure 5. Model of Network-based MTD in QPN.](image-url)
token \( n \) enters the PS queue of NMTD \( n \) and output a normal execution. With the thread and the execution, the request moves to the place FE-CPU-\( n \), and the following procedure is identical to the TISQPN model.

In the QPN model of NMTD, clients’ requests are interrupted by the re-configuration of networks. Both the shifting time \( t \) and randomness generation method \( \mu \) influence the connection of TCP for requests. Intuitively, the more frequent the shifting is, the less throughput is in the FEs. Although the PCS has a significant influence on security, the performance cost of the system is unaffected by the size of PCS from the point of clients’ requests.

To quantify the performance cost of NMTD, two metrics, throughput and response time, are used in this paper to evaluate the influence that diverse strategies of NMTD brought to the typical information system. The throughput of the system is measured by the number of tokens arrived or departed per time at the FEs, which can be denoted as:

\[
\Psi^+ = \frac{\sum_{t=1}^{T} \psi_{FE}^t}{T}
\]

where the superscript of \( \Psi \) stands for the arriving \( \Psi^+ \) and departure \( \Psi^- \), \( \psi_{FE}^t \) is the number of tokens in FEs at the time \( t \), and \( T \) stand for the whole processing times.

Moreover, the response time of a request is the sum of all individual residence times of tokens in queues and depositories at a simulation model without the interval of requests from a client, which can be denoted as:

\[
R_{req} = \sum_{v=1}^{P} R_{queue}^v + \sum_{v=1}^{P} R_{depository}^v
\]

where \( R_{queue}^v \) is the residence times of tokens in queues, \( R_{depository}^v \) is the residence times of tokens in depositories, and \( P \) is the max of none-NMTD places holding tokens.

However, in NMTD-TISQPN model, the execution of NMTD are selective in Translation and Transformation. The average response time of a request with NMTD in a specific simulation time-span can be denoted as:

\[
R_{req}^{avg} = R_{req}^{avg} + \sigma \cdot R_{trf}^{avg} + (1 - \sigma) \cdot R_{rel}^{avg}
\]

where \( \sigma = \left[ \frac{T_s}{T_{rel}} \right] \cdot \left( \frac{T_s}{T_{int}} \right)^{-1} \) is the ratio of NMTD transformation for time-based strategies and

\[
\sigma = \left[ \frac{T_e}{T_{trf}} \right] \cdot \left( \frac{T_e}{T_{int}} \right)^{-1} \] is for event-based strategies, \( T_s \) is the simulation time-span, \( T_{rel} \) is the request interval, \( T_{trf} \) is the average transformation period of event-based strategies, \( R_{avg} \) is the average response time of requests in system, \( R_{trf}^{avg} \) is the average process time of NMTD transformation, and \( R_{rel}^{avg} \) is the average process time of NMTD translation.

Moreover, an approximate result of the average response time of a request with NMTD can be denoted as:

\[
R_{req}^{avg} \approx R_{req}^{avg} + \frac{T_{int} (\lambda_n - \lambda_m)}{\tau \lambda_n \lambda_m} + \frac{1}{\lambda_n}
\]

where \( \lambda_n \) is the parameter of exponential service time for the translation and \( \lambda_m \) is the parameter of exponential service time for the transformation. Once the system configuration and implementation are set, the approximate average response time of a request with NMTD is entirely determined by the processing capability of the system and the frequency of NMTD shifting. In practice, the service time of translation is short compared to other processing time. Therefore, the average process time of
NMTD translation can be ignored in the evaluation. Then the average response time of a request with NMTD and the approximate value of the average response time of a request with NMTD can be rewritten as:

\[ R_{\text{req}}^{\text{avg}} = R_{\text{req}}^{\text{avg}} + \sigma \cdot R_{\text{req}}^{\text{avg}} \approx R_{\text{req}}^{\text{avg}} + \frac{1}{\lambda_m} \cdot \frac{T_{\text{int}}}{\tau} \]  \hspace{1cm} (7)

For connectionless requests, a ratio of interruption is taking into consideration, and can be denoted as:

\[ R_{\text{int}} = \left[ \frac{T_x}{\tau} \right] \cdot T_{\text{int}} \approx \frac{1}{\tau} \cdot \lambda_m \]  \hspace{1cm} (8)

where \( T_{\text{int}} \) is the processing time of network reconfiguration.

4. Case Study and Simulations Analysis

In the case study, a typical information system protected by NMTD with C/S architecture is modeled by QPN and simulated in QPME. The typical information system modeled by QPN in the case study takes the Tak's work [12] as a reference. To reduce the repetitive work in QPN modeling, only time-based and event-based strategies in NMTD are taking into consideration. Parameters of the typical information system, strategies of NMTD, and settings in QPME are shown in Table 1.

| Parameter Category | Parameters | Type | Value | Description |
|--------------------|------------|------|-------|-------------|
| QPME Queue         | FE         | Type | PS    | CPU scheduling |
|                    | BE         |      | FCFS  | DB-I/O scheduling |
|                    | Client     |      | IS    | Request scheduling |
|                    | Transformation |  | FCFS  | IP mutation scheduling |
|                    | Event-generator |  | Random | Event scheduling |
|                    | FE         | lambda | 1.4  | Exponential |
|                    | BE         | lambda | 7.5   | Exponential |
|                    | Client     | lambda | 0.06  | Exponential |
|                    | Transformation | lambda | 0.03 | Exponential |
|                    | Network-reconfig | lambda | 0.05 | Exponential |
|                    | Event-generator | mu, sigma | Normal |
|                    | FE         |       | 1 node | 1 CPU |
|                    | BE         |       | 1 node | 1 DB-I/O |
| TIS                | Thread     |       | 30 per FE's node | initial tokens |
|                    | Connection |       | 40 per FE's node | initial tokens |
|                    | Client     |       | 100 | initial tokens |
| NMTD               | Time-based |       | \( \tau = 5-50 \) | Interval 5 |
|                    | Event-based |       | \( \mu = 15, \sigma^2 = 9 \) | Expectation, Deviation |

In results collecting, the throughput of the system is collected as the departure throughput of queues in FEs, and the queue utilization of FE-CPU is also gathered as an additional metric. Moreover, the performance cost of NMTD is assessed under two significant scenarios, time-based, and event-based NMTD. Results are shown in Figure 6, Figure 7, and Table 2.
From Figure 6, both the throughput and the queue utilization of system are increased when the period of mutation is prolonged. Since the transformation of network configurations interrupts ongoing transmission and forces clients re-connect after synchronization with the controller or servers, the more mutations that NMTD makes, the less throughput and queue utilization are in the system. For event-based strategy, our evaluations show that the throughput and queue utilization of FE is similar to basic TIS, 1.4 for throughput and 100% for queue utilization. Therefore, the event-based strategy is a trade-off for specific security and performance, which ensures the timed mutation after a security-related event and reduces interruptions of legitimate operations. As shown in Figure 7, the average response time of the system is significantly influenced by the frequent mutation of NMTD. From the view of the ratio of NMTD mutation time on system response time, the network mutation does not play the major parts when the shifting period is above 166.67ms.

![Figure 6](image-url)  
**Figure 6.** The throughput and queue utilization rate of Network-based MTD system.

![Figure 7](image-url)  
**Figure 7.** The average response time of the system, the ratio of NMTD mutation, and the ratio of NMTD interrupt on Network-based MTD system.
Table 2. Average response time for NMTD-TISQPN model.

| Strategies | Category | Average response time (ms) |
|------------|----------|----------------------------|
|            | FE       | BE            | NMTD-m   | Sum          |
| Tau = 5    | 2.853    | 0.156         | 6.658    | 9.667        |
| Tau = 10   | 5.272    | 0.160         | 3.341    | 8.773        |
| Tau = 15   | 21.855   | 0.162         | 2.230    | 24.247       |
| Tau = 20   | 49.534   | 0.163         | 1.668    | 51.365       |
| Tau = 25   | 52.235   | 0.163         | 1.331    | 53.729       |
| Tau = 30   | 53.212   | 0.164         | 1.109    | 54.485       |
| Tau = 35   | 53.699   | 0.164         | 0.955    | 54.818       |
| Tau = 40   | 54.153   | 0.164         | 0.835    | 55.152       |
| Tau = 45   | 54.481   | 0.164         | 0.743    | 55.388       |
| Tau = 50   | 54.589   | 0.164         | 0.665    | 55.418       |
| None NMTD  | 54.519   | 0.164         | 0        | 54.755       |
| Event-based MTD | 54.591 | 0.164 | 2.225 | 56.980 |

Moreover, the interruption rate of network reconfiguration is decreased when the mutation period is increased. From the point of average response time shown in Table 2, both FEs, BEs, and NMTD affect requests of clients. When the period of NMTD transformation is short, frequent mutations hinder FEs and BEs to process requests and make the average response time increase. For a prolonged period of mutation, the bottleneck of FEs dominates the reduction of performance. Moreover, the influence of bottleneck for FE processing is more significant in event-based strategies of NMTD, which imparts that by applying suitable strategies the performance cost of NMTD protected typical information system is still attributed to the processing capacity of FEs and BEs.

5. Conclusions
The evaluation of performance cost for NMTD protected system is an important part when implementing NMTD to typical information systems. In this paper, a QPN based performance cost model for NMTD is introduced to cover both qualitative and quantitative analysis. Compared with Queueing Network and Colored Petri Net, the QPN model can describe the NMTD as well as analyze the performance of the system more accurately. By dividing the execution of NMTD into translation and transformation, both transparent transmissions and mutations of the network are included in the NMTD-TISQPN model. Simulated by QPME, a powerful QPN editor and simulator, results and analyses are collecting and demonstrating adequately and explicitly. From simulation results, the performance of a typical information system is affected by diverse strategies of NMTD. Frequent mutations of NMTD reduce throughput and queue utilization.

Moreover, event-based strategies bring fewer performance consumptions compared to time-based strategies. However, from the point of response time, the bottleneck in server processing is the central obstruction to improve performance. Moreover, the network shifting method is an affordable defense technique against adversaries effectively. In future works, our approach can be extended to multi-layered MT technologies and more general performance cost can be assessed, which will guide the implement and optimization of MTD.

Acknowledgment
Thanks to Tomasz Rak from the Rzeszow University of Technology in Poland, who gives supports in QPN modeling and QPME simulations.
References

[1] Carvalho M and Ford R 2014 IEEE Security Privacy 12 73–76 ISSN 1540-7993
[2] Jafarian J H, Al-Shaer E and Duan Q 2015 IEEE Transactions on Information Forensics and Security 10 2562–2577 ISSN 1556-6013
[3] Luo Y B, Wang B S, Wang X F, Hu X F and Cai G L 2015 2015 International Conference on Information and Communications Technologies (ICT 2015) pp 1–6
[4] Shi L, Jia C and Lu S 2007 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007) pp 119–122
[5] Carroll T E, Crouse M, Fulp E W and Berenhaut K S 2014 2014 IEEE International Conference on Communications (ICC) pp 701–706 ISSN 1550-3607
[6] Xiong X, Xu W and Zhao G 2018 Proceedings of the 8th International Conference on Communication and Network Security ICCNS 2018 (New York, NY, USA: ACM) pp 7–11 ISBN 978-1-4503-6567-3
[7] Okhravi H, Hobson T, Bigelow D and Streilein W 2014 IEEE Security Privacy 12 16–26 ISSN 1540-7993
[8] Connell W, Menascé D A and Albanese M 2017 Proceedings of the 2017 Workshop on Moving Target Defense MTD ’17 (New York, NY, USA: ACM) pp 53–63 ISBN 978-1-4503-5176-8
[9] Leeuwen B V, Stout W M S and Urias V 2015 MILCOM 2015 - 2015 IEEE Military Communications Conference pp 966–971
[10] Kounev S, Spinner S and Meier P 2012 Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering ICPE ’12 (New York, NY, USA: ACM) pp 9–18 ISBN 978-1-4503-1202-8
[11] Spinner S, Kounev S and Meier P 2012 Application and Theory of Petri Nets ed Haddad S and Pomellos L (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 388–397 ISBN 978-3-642-31131-4
[12] Bause F 1993 Proceedings of 5th International Workshop on Petri Nets and Performance Models pp 14–23
[13] Rak T 2014 2014 Federated Conference on Computer Science and Information Systems pp 769–774