


Network Moving Target Defense Technique Based on Self-Adaptive End-Point Hopping

Cheng Lei^{1,3}  · Hong-qi Zhang^{1,3} · Duo-he Ma² · Ying-jie Yang³

Received: 18 August 2016 / Accepted: 19 January 2017 / Published online: 9 February 2017
© King Fahd University of Petroleum & Minerals 2017

Abstract Moving target defense is a revolutionary technology changing the antagonistic pattern between attack and defense, with end-point information hopping one of the hotspots in this field. In order to counterpoise the defensive benefit of end-point information hopping and service quality of network system, a novel technique named self-adaptive end-point hopping technique based on adversary strategy awareness is proposed. To solve the blindness problem of hopping mechanism in the course of defense, hopping triggering based on adversary strategy awareness is applied to guide the choice of hopping mode by discriminating the scanning attack strategy, which enhances targeted defense. Furthermore, aimed at the low availability problem caused by limited network resource and high hopping overhead, satisfiability modulo theories are used to formally describe hopping constraints, so as to ensure low hopping overhead. Finally, both theoretical and experimental analyses are performed, demonstrating that the proposed technique can ensure low hopping overhead, while effectively discriminating and defending different types of scanning attacks.

Keywords Network moving target defense · Self-adaptive end-point hopping mechanism · Adversary strategy awareness · Software-defined network · Satisfiability modulo theories

✉ Cheng Lei
leicheng12150@126.com

- ¹ China National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450001, Henan Province, China
- ² State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing 100093, China
- ³ Henan Key Laboratory of Information Security, Zhengzhou 450001, Henan Province, China

1 Introduction

With the development of new types of attack techniques, such as zero-day exploit attack and advanced persistent threats, network security is facing serious challenges of “easy to attack and hard to defend” [1–3]. On the one hand, attackers have time advantage since they have enough time to scan and collect information on targeted systems before implementing attacks. And with the advantage of asymmetric information, attackers can install customized backdoors to control and threaten network systems once vulnerabilities have been found. On the other hand, existing defense methods, such as firewalls and intrusion detection, are often proposed or improved with a lag, because attack recognition and vulnerability patches usually lag behind attackers’ exploitation of system vulnerability. Furthermore, the essence of existing defense methods, based on prior knowledge, cannot ascertain all kinds of network attacks to defend the system proactively. The reasons are as follows: firstly, it is hard to prove the security of the design process of network systems, which inevitably leads to security vulnerabilities. Secondly, the certainty and the static structure of existing network information systems provide attackers with sufficient time to scan and discover vulnerabilities. Therefore, with network attacks having the tendency toward automation, intelligence and combination between hardware and software, it is increasingly difficult for the traditional passive defense architectures to effectively resist the unknown system hardware and software vulnerabilities, and to prevent potential types of backdoor attacks and the increasingly complex and intelligent network intrusion penetration, thus exacerbating the asymmetry between the offensive and the defensive in the network.

In order to improve the effectiveness of defensive mode, network moving target defense (NMTD) comes into being

[4,5] to provide a dynamic, non-deterministic and non-sustained runtime environment. NMTD breaks the dependency of the attack chain on the determinacy and consistency of network operating environment by multi-level dynamical changes. As one of the hotspots of NMTD, end-point hopping technique has received widespread attention [6,7]. However, these techniques do not unleash the full potentials of NMTD hopping, which restrict their applicability to naive network threat such as APT and zero-day attacks.

A review of the existing literature shows that there are two major problems in existing end-point hopping research:

1. The benefits from hopping defense decrease due to the inadequate dynamic of network hopping caused by self-learning insufficiency in reconnaissance attack strategy, leading to the blindness of hopping mechanism selection.
2. Due to the limited network resources and high overhead, the availability of hopping mechanism is poor.

To address the above problems, network moving target defense based on self-adaptive end-point hopping technique: (SEHT) is proposed. The key contributions of this paper can be shown in the following aspects:

1. In order to cope with the lack of existing hopping mechanisms able to self-adaptive to scanning attacks, a hopping triggering based on adversary strategy awareness is designed, using hypothesis tests to analyze scanning attack strategy, and guides the choice of hopping strategy, which enhances the defensive benefit.
2. Aimed at limited network resources and high hopping overhead, end-point hopping based on satisfiability modulo theories (SMT) [8] is proposed to formally describe the constraints of hopping, so as to ensure the low hopping overhead, which increases the availability of hopping mechanism.

The remainder of this paper is organized as follows: in Sect. 1, background knowledge and related work of end-point hopping are given. Self-adaptive end-point hopping algorithm is designed in Sect. 2, which consists of hopping triggering based on adversary strategy awareness and end-point hopping based on satisfiability modulo theory. In Sect. 3, the architecture of SEHT is constructed, and the communication protocol and hopping update policy are given. Sections 4 and 5 compare SEHT with existing hopping mechanism from the scanning attack resistance capability and hopping overhead. Finally, our work is concluded.

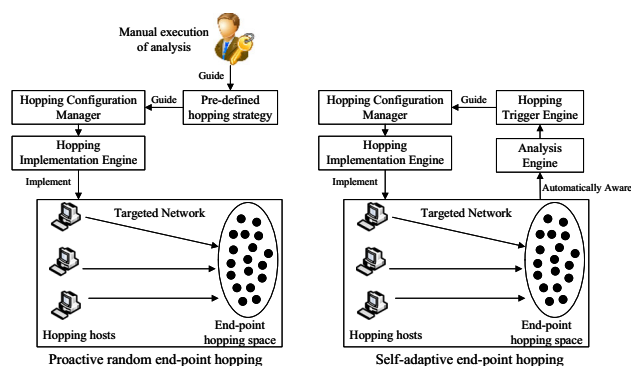


Fig. 1 Architecture of network hopping

2 Background Knowledge and Related Work

2.1 NMTD and End-Point Hopping

Network moving target defense [1] is an active defense novel to reverse the asymmetric situation between attack and defense. It keeps moving the vulnerabilities of a protected system through dynamic shifting, which can be controlled and managed by the administrator. In this way, the network resource vulnerabilities exposed to attackers appears chaotic and changes over time. Therefore, the cost and the complexity of attackers to launch a successful attack will be greatly increased, and the security of the protected system will be enhanced effectively. NMTD is rather an active defense principle than specific approaches. The active ability of NMTD is independent of the state of the environment it resides on. It keeps changing one or more attributes automatically and can be applied to different system attributes, such as IP address, service port number, protocol and running platform. End-point hopping is one of the key techniques in NMTD research.

End-point hopping, as shown in Fig. 1, tricks, evades and prevents scanning attacks by dynamically changing network status and configuration, such as IP address and port, thus, increasing the usage difficulty of vulnerabilities and backdoors, and ensuring the security of targeted systems. Existing end-point hopping mechanisms are proactive mode, which mainly adopt random hopping strategy [17–20]. As shown in the part with solid lines in Fig. 1, hopping configuration manager is used to configure end-point hopping on the basis of security objectives. Then, hopping implementation engine is used to implement end-point hopping. However, since random hopping lacks offensive and defensive situational awareness, the effectiveness and availability of end-point hopping are limited. Reactive mode is designed [9] so as to enhance the self-adaptive capability of NMTD. As is shown in Fig. 1, it is equipped with analysis engine and hopping triggering engine based on random hopping. Analysis engine is used to perceive and analyze network system security status,

triggering different hopping strategies in hopping triggering engine and consequently generating end-point hopping constraints.

2.2 Related Work

Existing end-point hopping technique research can be classified into end-point hopping based on traditional network architecture and end-point hopping based on newly invented network architecture, with details as follows.

In traditional network architecture, Atighetchi et al. [10] proposed a hopping mechanism using false IP and port information to confuse scanning attack during net-flow exchange. Lee et al. [11] proposed a random port hopping mechanism, which calculates next hopping end-point information to evade scanning attack by using pseudorandom function or shared secret key. MT6D [12] uses large IPv6 address space property to implement end-point information hopping so as to increase the unpredictability. Although the above hopping mechanisms have their own advantages, existing mechanisms lack the capability of self-adaptive to different reconnaissance strategies, leading to blindness in the process of network defense. Meanwhile, regarding the hopping synchronization in traditional network architecture, Lee and Thing [11] adopted a strict time synchronization mechanism with high-degree coupling among communication parties to ensure hopping synchronization, but the method is vulnerable to network delay interference. Hari and Dohi [13] introduced a discrete Markov chain based on RPH so as to improve the success rate among communication parties. Kai et al. [14] proposed a novel synchronization method by additionally opening the corresponding end-point information of the previous and the subsequent hopping period. HOPERAA algorithm was designed in [15], eliminating the influence of linear clock drift on hopping synchronization.

With the development of software-defined network (SDN) architecture [16], the feature of logic control plane being separated from data transfer plane of SDN has brought a new solution to effective collaborative management in distributed routing. What's more, end-point hopping based on SDN can change hopping period and hopping rules dynamically, which achieves the manageability of end-point hopping. NASR [17] prevents connection requests that do not fall within the service period by using address transition of packet header and the update of flow table based on DHCP update. SDNA [18] confuses scanning attackers by virtual hopping, which deploys a hypervisor node in each subnet to ensure hopping consistency. OF-RHM [19] proposes virtual end-point mapping mechanism based on Openflow [16], converting real IP to virtual IP so as to implement end-point hopping. However, since OF-RHM only implements space hopping, attackers can improve the

success rate of scanning attacks by changing scanning frequency. To address this problem, Jafarian et al. [20] proposed ST-RHM hopping mechanism, which can resist cooperative scanning attack effectively by using temporal-spatial mixed hopping based on SDN. However, the double hopping mechanism inevitably leads to overhead increase and service losses.

In the rest of this paper, we will give the detail of self-adaptive end-point hopping technique on the basis of related work mentioned above.

3 Self-Adaptive End-Point Hopping Algorithm

3.1 Overview of Self-Adaptive End-Point Hopping Algorithm

Self-adaptive end-point hopping algorithm consists of hopping triggering based on adversary strategy awareness and end-point hopping based on SMT. Hopping triggering based on adversary strategy awareness discriminates scanning strategy by using hypothesis tests, which guides the choice of hopping strategy. On that basis, end-point hopping based on SMT formally describes the constraints of hopping so as to ensure the low overhead and availability of hopping. As illustrated in Fig. 1, when self-adaptive end-point hopping algorithm is adopted in network. The hopping analysis engine will perceive and analyze network system security status. When scanning attacks are obtained, the hopping triggering engine will trigger different hopping strategies and consequently generating end-point hopping constraints. After generating the end-point information satisfying constraints, the hopping configuration manager will assign end-point information to the corresponding subnet according to security strategy. The hopping implementation engine is used to configure end-point hopping automatically.

The notions used in SEHT are listed in Table 1. Assume the network nodes needed protection are represented by $\{h^1, h^2, \dots, h^l\}$, which are distributed in subnets $\{s^1, s^2, \dots, s^k\}$, $k \leq l$. Hopping end-point information (hEI) consists of IP address and port, which can be expressed as $\langle \text{IP}, \text{Port} \rangle$. The availability hEI set are those all hEIs (hEI_A) except the actual end-point information (EI_i) and the set of hEI not meeting SMT constraints ($\neg(\text{hEI})$), which can be expressed as $\{\text{hEI} | \text{hEI}_A \wedge \neg(\text{EI}_1 \vee \dots \vee \text{EI}_l) \wedge \neg(\text{hEI})\}$. Maximum entropy entails that hEI of each node must be chosen from the *largest* available hEI space. However, it is impossible to choose an hEI from all unused ranges simultaneously because a range can only be assigned and routed to one physical subnet at any given time. Therefore, SEHT adopts multilayer hopping to assign available hEI space.

Table 1 The notions used in SEHT

Character	Description
<i>Variable</i>	
hEI	End-point information consisting of IP and Port
T_{EHP}	The hopping period of end-point
$T_{BHR}, T_{LTHR}, T_{HTHR}$	The hopping period of different hEI range
$N_{BHR}^{s_i}, N_{LTHR}^{n_i}, N_{HTHR}^i$	The number of hEI in different range in one hopping period
m_B, m_L, m_H	The number of hEI space of different range in one hopping period
N_{fail}	The number of failed requested packet
$P_i^{Src}(\pi), P_i^{Dst}(\pi)$	The probability distribution of the source/destination address of failed requests in one hopping period
w_i^{EI}	Weighted value of the i th hEI
$b_T^v(k), b_i^j, C_{j_1, j_2}, B_j^k$	Boolean variable
<i>Constant</i>	
n_{HTHR}	The minimum number of hEI required in HTHR
$T_{EHP}^{lb}, T_{EHP}^{ub}$	Setting minimum/maximum hopping period
$\delta_1, \delta_2, \delta_3$	Setting threshold value
L_{max}	The maximum length of forwarding path cannot exceed
σ	Tuning parameter
α	Smoothing coefficient
Φ	The lower bound of the number of end-point information in each hopping space
C_v^{max}	The maximum net-flow table size of switch v

$$N_{hEI} = \sum_{s_i=1}^{m_B} N_{BHR}^{s_i}, N_{BHR}^{s_i} = \sum_{n_i=1}^{m_L} N_{LTHR}^{n_i},$$

$$N_{LTHR}^{n_i} = \sum_{i=1}^{m_H} N_{HTHR}^i, N_{HTHR}^i \geq n_{HTHR} \quad (1)$$

$$T_{EHP}^i = \frac{T_{LTHR}}{V^i} \quad (2)$$

As shown in Eq. (1), $N_{hEI} = \sum_{s_i=1}^{m_B} N_{BHR}^{s_i}$ means the total number of available hEI is divided into m_B number of base hopping range (BHR) according to the number of subnet (s_i) and its scale in base hopping period (T_{BHR}). Then, each BHR ($N_{BHR}^{s_i}$) is divided into m_L number of low-frequency temporal hopping range (LTHR) in each T_{LTHR} according to the number of nodes in subnet and its resource value, which

is shown in the middle of Eq. (1). What's more, each LTHR ($N_{LTHR}^{n_i}$) is then divided into m_H number of high-frequency temporal hopping range (HTHR). And each HTHR (N_{HTHR}^i) contains at least number of n_{HTHR} hEIs, which is a setting threshold so as to ensure the hopping space of each node in on hopping period.

On the other hand, the relationship between T_{EHP} and node importance (V^i) is shown in Eq. (2). The hopping period of node i (T_{EHP}^i) decreases with the increase in node importance $V^i \in [0, 10]$. Besides, $T_{HTHR} = T_{EHP}$ and $T_{BHR} = c \cdot T_{LTHR}$, where c is constant ($c \in Z^+$).

3.2 Hopping Triggering Based on Adversary Strategy Awareness

In order to improve the self-adaptive of end-point hopping, SEHT adopts hypothesis tests based on Sibson entropy to discriminate scanning attack strategy since network scanning, as a precondition technique and the initial phase of attacks, plays an important role in network attacks [3]. Therefore, SEHT discriminates scanning strategy by analyzing behavior characteristic of different scanning strategies, thus achieving self-adaptive end-point hopping.

Network scanning is a kind of network reconnaissance technique by means of sending probe packets to selected end-point space range [21]. With different scanning techniques constantly springing up, network scanning attack improves its efficiency by selecting targeted scanning strategy based on the network structural characteristics and the knowledge gained. It can be described by two attributes: the scanning width and the scanning frequency. Accordingly, scanning attack strategy can be classified into three types: blind scanning, half-blind scanning and follow-up scanning:

1. Blind scanning strategy: It is used when an attacker has to scan the entire active end point. Due to the certainty and the static characteristic of existing network information system, attackers adopt blind scanning strategy to improve its efficiency by evenly scanning without repetition [22].
2. Half-blind scanning strategy: It is used when an attacker knows the node distribution of the selected range of end-point information to scan. Half-blind scanning strategy is adopted to achieve higher success rate by unevenly scanning with repetition [23].
3. Follow-up scanning strategy: It is directed at network systems implementing NMTD mechanisms. When knowing the node distribution and the use of hopping mechanism, attackers try to obtain the hopping pattern of end points by spatial compression and scanning frequency change. Then, follow-up scanning strategy is adopted so as to follow the hopping of specific end point by uneven scanning with changeable frequency [24].

In terms of the behavior characteristics of different network scanning strategies, SEHT adopts Sibson entropy [25] to obtain the distribution of failed requested packets so as to discriminate scanning strategy. Only failed request packets are chosen because successful requests contain both normal packets of legitimate users and the successful probe packets of attackers, but there is only one valid hEI for each end point in every hopping period [20,29]. Therefore, the collected failed request packets can be used as samples to characterize malicious scanning strategy effectively. Sibson entropy calculates the difference between two given probability distributions based on information theory. It has high accuracy and good stability in different anomalous awareness application scenarios [26].

Suppose the total number of failed request packets in the t th hopping period is N_{fail} . The number of failed request packets in the i th divided hEI range is denoted as N_{fail}^i . Equation (3) is used to calculate the probability distribution of the source and the destination address of failed requests in one hopping period, denoted as $P_i^{\text{Src}}(\pi)$ and $P_i^{\text{Dst}}(\pi)$, respectively, with $j \in \{\text{Src}, \text{Dst}\}$, $\pi \in \{\text{hEI}\}$. Then, follow-up scanning strategy is discriminated after analyzing source address probability distribution of probe packets in adjacent T_{LTHR} . Besides, blind scanning strategy is then discriminated after analyzing destination address probability distribution of probe packets in each T_{EHP}

Equation (4) indicates the Sibson entropy of the source address probability distribution of the failed request packets in the two consecutive T_{LTHR} of the i th end point, in which $D_i(p, q) = \sum_{\pi \in \Pi_i} p(\pi) \cdot \log \frac{p(\pi)}{q(\pi)}$, and $\overline{P^{\text{Src}}} = \frac{1}{2}[P_{t-1}^{\text{Src}}(\pi) + P_t^{\text{Src}}(\pi)]$. In order to prevent the interference of network jitter, Sibson entropy is calculated in two consecutive T_{LTHR} instead of in two consecutive T_{EHP} of the i th end point. Based on Eq. (4), whether the scanning is follow-up strategy or not can be discriminated by comparing the Sibson entropy with the setting threshold.

Chauvenet criterion, shown in Eq. (5), is used to eliminate the abnormal high-frequency temporal hopping space. If blind scanning strategy is used, attackers are to scan the entire end-point space. The average number of scanned times of every end point is $N_{\text{fail}}/m_B m_L$ in the ideal condition. However, because attackers might not always complete the scan of the whole end-point space within one T_{EHP} , the Sibson entropy directly calculated based on the distribution of failed probe packets of destination address and that of $N_{\text{fail}}/m_B m_L$ in one T_{EHP} will be larger. Therefore, the destination address probability distribution of the failed probe packets in the t th T_{EHP} and its modified Sibson entropy are calculated by using Eq. (6), where $D(p, q) = \sum_{\pi \in \Pi} p(\pi) \cdot \log \frac{p(\pi)}{q(\pi)}$, and $\overline{P_t^{\text{Dst}}} = \frac{1}{2}(P_t^{\text{Dst}}(\pi) + \frac{n_{\text{fail}}}{m'_B m'_L})$. By comparing with the setting threshold, whether blind scanning strategy is adopted or not can be determined. If not adopted, attackers will use half-

blind reconnaissance strategy.

$$P_i^j(\pi) = \pi_k \cdot \left(\sum_{k=1}^{N_{\text{fail}}} \pi_k \right)^{-1} \tag{3}$$

$$D_s(P_{t-1}^{\text{Src}}(\pi), P_t^{\text{Src}}(\pi)) = \frac{1}{2} \left\{ D_i \left[P_{t-1}^{\text{Src}}(\pi), \overline{P^{\text{Src}}} \right] + D_i \left[P_t^{\text{Src}}(\pi), \overline{P^{\text{Src}}} \right] \right\} \tag{4}$$

$$\frac{N_{\text{fail}}^i - N_{\text{fail}}/m_B m_L}{(m_B m_L)^2 / 12} < -\xi \tag{5}$$

$$D_s \left(P_t^{\text{Dst}}(\pi), \frac{N_{\text{fail}}}{m'_B m'_L} \right) = \frac{1}{2} \left\{ D \left[P_t^{\text{Dst}}(\pi), \overline{P_t^{\text{Dst}}} \right] + D \left[\frac{N_{\text{fail}}}{m'_B m'_L}, \overline{P_t^{\text{Dst}}} \right] \right\} \tag{6}$$

In order to improve the unpredictability of end-point hopping, SEHT selects different hopping strategies according to the discrimination of scanning attack strategy. Consequently, hEI space is generated. The details are as follows.

1. If there is $\sqrt{D_s(P_{t-1}^{\text{Src}}(\pi), P_t^{\text{Src}}(\pi))} \leq \delta_1$, follow-up scanning strategy is implemented by attackers. Since only spatial hopping strategy cannot reach Nash equilibrium between end-point hopping and scanning attack [14], SEHT selects spatial-temporal mixed hopping strategy, which introduces hopping period stretch policy based on weighted random hopping. The so-called hopping period stretching means that the hopping period of hEI is stretched according to both network environment and reconnaissance attack frequency change. When SEHT detects malicious follow-up scanning strategy, it will reduce T_{EHP} according to scanning attack frequency, which improves network security. When SEHT does not detect follow-up scanning attack in two consecutive T_{EHP} , it will increase T_{EHP} according to network environment, which improves network communication performance. The magnitude of T_{EHP} to decrease is shown in Eq. (7). It means the $t+1$ th hopping period T_{EHP}^{t+1} is determined by the t th hopping period and scanning attack frequency. T_{EHP}^{lb} is the setting minimum hopping period, which is to prevent communication interruption caused by short hopping period. α is smoothing coefficient, which is set $\alpha = 0.75$ [27].

$$T_{\text{EHP}}^{t+1} = \max[\alpha T_{\text{EHP}}^t d/n_{\text{fail}}^t + (1 - \alpha)T_{\text{EHP}}^t, T_{\text{EHP}}^{lb}] \tag{7}$$

The magnitude of T_{EHP} to increase is shown in Eq. (8). It means the $t+1$ th hopping period T_{EHP}^{t+1} is determined by network delay. T_{EHP}^{ub} is the setting maximum

hopping period, which is used to prevent communication security decrease caused by long hopping period. SEHT employs discrete time hidden Markov models [28] to calculate network delay. It consists of quintuple: $\lambda \triangleq \{N, M, \boldsymbol{\varphi}, \mathbf{P}, \mathbf{B}\}$, in which $\boldsymbol{\varphi}$ is the initial distribution vector of network status; \mathbf{P} is Markov chain transition matrix of network status; and \mathbf{B} is observation matrix from network status to network delay. If network status space is $Q_s = \{1, 2, \dots, N\}$, the quantization space of network delay t_d is $O_s = \{1, 2, \dots, M\}$. The process is as follows: firstly, the incomplete mathematical expectation maximization algorithm is used to calculate the maximum likelihood estimation of $\lambda \triangleq \{N, M, \boldsymbol{\varphi}, \mathbf{P}, \mathbf{B}\}$, which is $\lambda^* = \arg \max P(o|\lambda)$. Then, Viterbi algorithm is employed to calculate the optimal state sequence $q_s^{kt'} = \{q_1', q_2', \dots, q_{kt'}\}$. Finally, $q_s^{kt'}$ and transition probability matrix \mathbf{P} are used to calculate the $t+1^{\text{th}}$ network delay $t'_d = o'_{t+1}$.

$$T_{\text{EHP}}^{t+1} = \begin{cases} T_{\text{EHP}}^t + t'_d & T_{\text{EHP}}^t + t'_d \leq T_{\text{EHP}}^{ub} \\ T_{\text{EHP}}^{ub} & \text{else} \end{cases} \quad (8)$$

- If there is $\sqrt{D_S(P_t^{\text{Dst}}(\pi), \frac{N_{\text{fail}}}{M'})} \leq \delta_2$, blind scanning strategy is implemented by attackers. SEHT selects weighted random hopping strategy. SEHT uses Eq. (9) to calculate weighted value w_i^{EI} of hEI in T_{EHP} . Then, SEHT randomly selects high weighted value from end-point space as hEI in the next hopping period. High weighted value is chosen because malicious scanning with blind strategy adopts non-repetitive uniform scanning. The Sibson entropy of scanned end-point information in hEI is $\sqrt{D_S} = 0$. The higher the weighted value is, the higher the possibility of the end point being scanned is. Therefore, choosing scanned end point can evade malicious scanning effectively. Assume n_{HTHR} number of hEI can be selected by end-point h^i , indicated as $\{\text{hEI}_1, \dots, \text{hEI}_p\}$. Equation (10) is used to select hEI in the next hopping period, where hash function H_f and secret key K_s are shared parameters.

$$w_i^{\text{EI}} = \begin{cases} 0 & l \leq \frac{l}{M} - \frac{M^2}{6} \\ 1 - \frac{1}{\delta_2} \min(\sqrt{D_S}, \delta_2) & \text{else} \end{cases} \quad (9)$$

$$\begin{aligned} \text{hEI}_\alpha &\in \text{hEI}_{n_{\text{HTHR}}}, \alpha \\ &= H_f(\text{SrcIP}, \text{SrcID}, K_s) \bmod n_{\text{HTHR}} + 1 \end{aligned} \quad (10)$$

- Otherwise, when $\sqrt{D_S(P_t^{\text{Dst}}(\pi), \frac{N_{\text{fail}}}{M'})} > \delta_2$ and $\sqrt{D_S(P_{t-1}^{\text{Src}}(\pi), P_t^{\text{Src}}(\pi))} > \delta_1$ establish, half-blind scanning strategy is implemented by attackers. SEHT selects reversed hopping strategy based on weighted value. Since half-blind scanning strategy is to scan spe-

cific range of end-point information repeatedly [29], SEHT calculates w_i^{EI} of hEI and selects end-point information with lower weighted value from hEI as the end-point information in the next T_{EHP} , which is shown in Eqs. (9) and (10).

Furthermore, if attackers use mixed scanning strategies based on the self-learning of scanning strategies, SEHT implements corresponding hopping strategy according to the priority of follow-up scanning, half-blind scanning and blind scanning for efficient defense.

3.3 End-Point Hopping Based on SMT

In order to achieve the manageability and low overhead in the process of hopping implementation, SMT solver is used to obtain the required hEI set, which meets the security and performance constraints in end-point hopping. Although SMT solving is still NP problem, the existing SMT solver, such as Z3 [8], can reach millions of orders of magnitude and thus can be used effectively to obtain required hEI set. Since end-point information hopping implementation needs HS and HC in collaboration, the hopping constraints can be divided into routing constraints, end-point constraints and forwarding path constraints. Define Boolean variable $b_T^v(k)$ indicates whether hopping switch v forwards the k th net flow in T_{EHP} or not. If hopping switch v forwards the k th net flow in T_{EHP} , $b_T^v(k) = 1$. Otherwise, $b_T^v(k) = 0$. The details of SEHT constraints are shown as follows:

- Capacity constraint: This constraint is used to select hopping routers that can carry the maximum net-flow table size so as to prevent packet loss caused by data overflow. The details are shown in Eqs. (11)–(13). One important characteristic of the resource usage on hopping switches is that the marginal costs of resource usage inflate with the increase in the workloads of the resources [30]. A heavily loaded hopping switch will spend time and energy on matching a forwarding rule for an incoming network packet compared with a lightly loaded one because more rules need to be considered in such a heavily loaded switch. Equation (11) indicates the exponential function of marginal cost, where $\sigma = 2n$ is a tuning parameter [31]. $1 - \frac{C_v(k)}{C_v}$ indicates the utilization ratio of the forwarding table of v when the forwarding table of the k th net flow is added. Equation (12) indicates that the accumulated cost of added net-flow table should be under the maximum net-flow table size C_v^{max} that hopping routers can carry. Therefore, the selected routers have the capability of carrying data fluctuation due to network load balance and network jitter. Equation (13) reduces route overhead by using route aggregation and adjacent allocation principles in routing update, which prevents the explosion of flow table size. $B_{j_1}^k \wedge B_{j_2}^k \wedge C_{j_1, j_2}$ means

the assigned end-point information j_1 and j_2 are in consecutive in the same subnet in continuous T_{EHP} , in which $B_j^k = \vee_{h^i \in s^k} b_j^i$ represents there is at least one end-point node h^i in subnet s^k assigned to hopping space j . Besides, Φ is the lower bound of the number of end-point information in each hopping space.

$$c_v(k) = C_v \left(\sigma^{1 - \frac{C_v(k)}{C_v}} - 1 \right) \tag{11}$$

$$\forall h R_i, \text{ if } C_v^{\max} - \sum_{i=1}^k b_T^v(i) \cdot c_v(i) \geq 0, b_T^v(i) = 1 \tag{12}$$

$$\sum_k \sum_{j_i} \sum_{j_i \neq j_2} B_{j_1}^k \wedge B_{j_2}^k \wedge C_{j_i, j_2} \geq \Phi \tag{13}$$

2. Hopping space selection constraint: this constraint ensures the unpredictability of SEHT by limiting repetition rate in hEI selection. Equation (14) ensures that every end-point node can be assigned hEI. Equation (15) sets repetition rate threshold δ_3 so as to ensure the repetition of selected hEI not exceeding the threshold. Furthermore, Eq. (16) requires that the assigned hEI in the last hopping period won't be assigned in the following hopping period. This constraint ensures every node can be assigned required hEI and improves the unpredictability of hopping.

$$\sum_{1 < j \leq M} b_i^j \geq 1 \tag{14}$$

$$\sum b_i^j \geq \frac{N_{LTHR}^i - 1}{2\delta_3 n_{HTHR}} \tag{15}$$

$$\forall hEI \in \{hEI\}_{T_{EHP}}, b_i^j = 0 \tag{16}$$

3. Reachability constraint: This constraint means all net flows in forwarding routers are reachable to destination end-point nodes. Equation (17) represents that the in degree and out degree of each router in the forwarding path are equal. Equation (18) means each router in the forwarding path is physically adjacent to its last hopping router and next hopping router, in which $\chi(hR_i)$ is routing set eliminating source and destination routers in the forwarding path. However, forwarding net flows from one router to its next physical adjacent router is not enough to guarantee the reachability of net flow. Equation (19) requires that the distance from the next hopping router to destination router is no larger than the distance from the current hopping router to destination router, in which d_k^{i-Dst} represents the distance from router i to destination router.

$$\text{If } b_T^k = 1, k \in [1, n], \sum_{i \in I} b_T^v(i) = \sum_{o \in O} b_T^v(o) \tag{17}$$

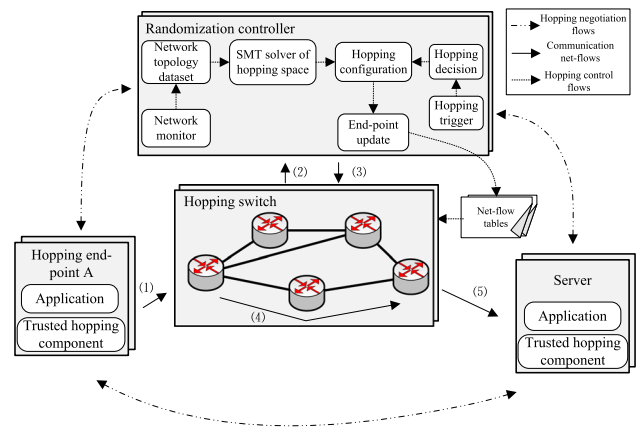


Fig. 2 Architecture of SEHT

$$\text{If } b_i^k = 1, \forall h R_j \in \chi(h R_i), \sum b_j^k = 2 \tag{18}$$

$$\text{If } \forall h R_j \in \{h R | \text{next-hop of } h R_i\}, d_k^{j-Dst} \leq d_k^{i-Dst} \tag{19}$$

4. Forwarding path delay constraint: This constraint prevents service performance decrease due to the excessive transmission delay. Since net-flow transmission delay is positively correlated with the number of routing nodes [32], Eq. (20) indicates that the maximum length of forwarding path cannot exceed the threshold L_{\max} .

$$\sum b_i^k \leq L_{\max}, i \in \{Src, h R_1, \dots, Dst\} \tag{20}$$

4 Architecture and Protocols

4.1 Architecture of SEHT

As shown in Fig. 2, SEHT uses hopping switch (HS), randomization controller (RC) and the trusted hopping components (THC) of end-point nodes to implement network hopping collaboratively. RC divides {hEI} into BHR according to the number of subnet and its scale. Then, HS divides BHR into LTHR according to the number of end points and their importance. After that, THC selects hEI according to hopping strategy by using shared parameters with HS.

RC mainly consists of hopping triggering, hopping decision engine and SMT solver of hopping space module. The function of hopping triggering module is to analyze scanning strategy based on hypothesis tests, according to the illegal connection packets reported by HS. Hopping decision engine is to select different hopping strategies according to scanning strategies, while SMT solver is to obtain the required end-point information set according to hopping constraints and global view of SDN. Then, RC updates LTHR to HS.

THC of end-point nodes is used to implement virtual mapping from EI to hEI. THC in SEHT is based on a universal virtual-network kernel driver TAP [33]. In order to be transparent to users' applications, network hopping needs to operate Ethernet frames using TAP under Linux. The seamless hopping method is used as follows:

1. Connection interception: THC creates an initial virtual mapping and replaces EI provided by the application with the hEI. As a result, the transport protocol stacks on both the client and the server perceive a connection identified by EI. Since the identity of the transport end point is detached from the network end point, the movement of the physical node is transparent to the transport layer or higher layer protocols.
2. Connection translation: While the actual traffic is routed through the network using hEI, to allow network packets flowing through internal connections, THC intercepts packets in the network layer and translates EI in the packet headers to or from hEI for outgoing packets and incoming packets, respectively.
3. Connection transformation: It coordinates the moving of an end point associated with active connections to another place. The process involves first suspending an active connection at one location and later resuming it at another. To suspend a connection, THC on hopping end point saves the current state. When a connection is resumed, THC updates end-point information mapping and notifies the other end point the new hEI.

The functions of HS are as follows: It is used for detecting, filtering and collecting network topology changes and illegal connection requests, which are reported to RC at regular intervals. Besides, HS forwards net-flow packets according to net-flow tables. What's more, if the packets cannot be matched, or they are packets of ARP, ICMP, DNS and DHCP protocols, HS will forward them to RC. Since the flow tables need to update because of end-point and routing hopping during network communications, it is necessary to prevent the inconsistency of flow table update and packet loss.

In order to ensure the hopping efficiency of SEHT and the stability of network sessions, end points will store two hEI the first time. One is considered as the active hopping end-point information. The other will be utilized at the next hopping period, which is pre-calculated so as to notice other communicating THCs to be prepared to hopping when T_{EHP} is expired. At the same times, since there are still ongoing sessions in the network during end-point hopping, Change Time To Live (CTTL) is set so that expired hEI is retained to receive packets of existing sessions. When CTTL is expired, the previous hEI cannot be used anymore.

4.2 Communication Protocol

The communication protocol of SEHT under SDN architecture is shown in Fig. 2. The details are as follows:

1. Client A sends session request packet $K_{Ec}(ID_A, req, K_s)$. It uses the private key of client K_{Ec} to make signature of client identification ID_A , request information req and shared key K_s .
2. The corresponding HS of client A receives the request packet and verifies the signature of client by using client publish key K_{Dc} . After the identification of client is successfully verified, MS transforms the address in packets ($IP(A), IP(B) \rightarrow hIP(A), hIP(B)$) and sends requests to RC for the corresponding hEI of destination server.
3. RC virtually maps EI to hEI according to query corresponding HS of server and updates the flow table of forwarding path, respectively. RC sends $K_{EHC}(ID_{HC}, mEI', hR)$ by using its private key K_{EHC} to make signature of RC identification ID_{HC} . It transforms the address in packets ($IP(A), IP(B) \rightarrow hIP(A), hIP(B)$). Then, the hopping information of EI and flow tables are updated.
4. HR forwards the net flows according to the updated flow table and sends the request $K_{Ec}(ID_A, req, K_s)$ to the corresponding HS of server.
5. The corresponding HS of server transforms address in packets ($hIP(A), hIP(B) \rightarrow hIP(A), IP(B)$) after receiving the request and sends it to the server providing services.

Since the flow tables need to update because of end-point and routing hopping during network communications, it is necessary to prevent the inconsistency of flow table update and packet loss. To solve this problem, SEHT adopts "delete in sequential order, and add in reverse" update policy. The so-called delete in sequential order net-flow table update method means that net-flow tables are deleted in the order from source HS to destination HS, while "add in reverse" net-flow table update method means that net-flow tables are added in the order from destination HS to source HS. What's more, Theorem 1 below proves the correctness of net-flow table update policy.

4.3 Implementation of SEHT

As is shown in Fig. 3, we implemented a proof-of-concept SEHT in a designated class C subnet. The network is divided into 3 subnets, each containing several network nodes, such as hosts, servers and databases.

If Client A is a benign user, it will send request packets to its corresponding HS by using THC when Client A wants to get access to IIS server. HS of Client A will randomly

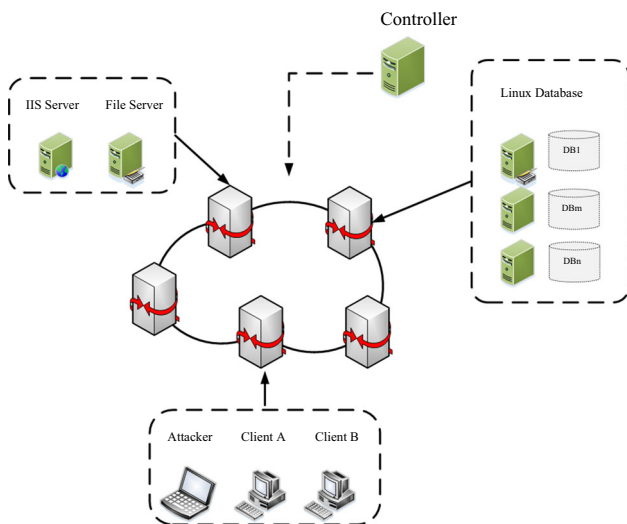


Fig. 3 Implementation of SEHT

select hEI from HTHR and transform EI to hEI. The hEI of Client A is used to establish connection with IIS server by using SEHT communication protocol. After that, when RC triggers to implement end-point hopping, the corresponding HS of Client A and IIS server will select hEI according to self-adaptive end-point hopping algorithm. Several network activities are running during end-point hopping, including files downloading and web browsing. Connections, especially long-lived connections, function soundly and are not affected by end-point hopping. Therefore, the implementation proved that SEHT is feasible in SDN network.

On the other hand, when attacker wants to access IIS server, it does not have private key to send request packets to its corresponding HS. HS will not assign an available hEI to attacker. Besides, when attacker scans the hEI of Client A in the subnet, SEHT changes hEI at random intervals according to attacker scanning strategies. Hence, the scanning success rate will decrease dramatically, which is analyzed in Sect. 4.1. What’s more, even though attacker intercepts the hEI of Client A or IIS server, it does not know the hEI in the following hopping period. When the existing hEI is expired, packets using the expired hEI will be failed and obtained by SEHT. Consequently, the malicious action of attacker will be exposed. Therefore, SEHT increases the difficulty of attacker to establish connection with IIS server.

5 Theoretical Analysis

5.1 Security Analysis

Suppose there are n_l active end-point nodes in the network, with the end-point information space being m , scanning width of attacker being w , and the scanning frequency

being $1/T_{SCN}$. The number of the end-point information scanned by the attack is $n_s = w \cdot t/T_{SCN}$, $n_s \leq m$. The ratio of scanning frequency to hopping frequency is

$$r = T_{EHP}/T_{SCN}.$$

5.1.1 The Capability to Resist Blind Scanning Attack

Since the blind scanning strategy is used so as to enhance the scanning rate, the success rate of scanning x active end-point nodes by attackers in static network, which can be supposed as $T_{EMP} = \infty$, obeys hypergeometric distribution expressed as $P_b(x) = (C_{n_l}^x \cdot C_{m-n_l}^{n_s-x})/C_m^{n_s}$. Hence, the success rate of attackers in static network is $P_{hb}^{static}(x > 0) = 1 - aC_{\phi m - n_l}^{n_s/a}/\phi C_{\phi m}^{n_s/a}$. In OF-RHM [19], ST-RHM [20] and SEHT network, the probability of successfully scanning x active nodes during one hopping period obeys Bernoulli distribution. The success rate of attackers using blind scanning strategy is $P_b(x > 0) = 1 - [1 - rwn_l/(mn_l + mrw)]^{n_s}$. Particularly when $r = 1$, the scanning attack frequency is the same as the hopping frequency, and the probability that an attacker successfully launching blind scanning is $P_b(x > 0) = 1 - [1 - rwn_l/(mn_l + mrw)]^{n_s}$. Compared with static network, it can be concluded that OF-RHM, ST-RHM and SEHT can effectively resist blind scanning strategy, which is consistent with the conclusion in [34,35].

5.1.2 The Capability to Resist Follow-up Scanning Attack

When attackers use follow-up scanning strategy, there will be $r \geq 1$ in active scanning. Suppose attackers can repeat scanning b times in one T_{EMP} . The success rate of attackers in OF-RHM is $P_{fu}(x > 0) = 1 - [1 - bn_l'/(n_l' + \phi mb)]^{n_s}$, which is consistent with the analysis in [14]. The success rate of attackers in ST-RHM is $P_{fu}(x > 0) = 1 - [1 - (bn_l' - n_\gamma)/(n_l' + \phi mb)]^{n_s}$. Since SEHT deploys hopping period stretch policy, the hopping rate will lead to $r \leq 1$ after the follow-up scanning strategy is learnt by SEHT. As a result, the success rate of attackers in SEHT is $P_{fu}(x > 0) = 1 - [1 - (rn_l' - n_\gamma)/(n_l' + \phi m)]^{n_s}$. Analysis shows that compared with ST-RHM, SEHT can effectively defend the follow-up scanning by combining spatial hopping with hopping period stretch policy.

5.1.3 The Capability to Resist Half-Blind Scanning Attack

Since half-blind scanning strategy is used to actively scan specific range of end-point information which is physically adjacent to scanning source, it can be assumed that attackers can repeat scanning a times, and the scanning range is ϕm , $\phi \in (0, 1)$, where there are n_l' active end-point nodes. Since OF-RHM adopts random hopping, the success

rate of attackers using half-blind scanning strategy in OF-RHM is $P_{hb}(x > 0) = 1 - a[1 - wrn'_l / (\phi mn'_l + \phi mwr)]^{n_s}$. As for ST-RHM, it uses deceiving hopping. It can be assumed that there are n_γ hEI invalid at the end of each hopping period. The success rate of attackers using half-blind scanning strategy in ST-RHM is $P_{hb}(x > 0) = 1 - a[1 - (wrn'_l - \phi mn_\gamma) / (\phi mn'_l + \phi mwr)]^{n_s}$. Since SEHT deploys random hopping based on weighted value, σ hEI will be selected for the next hopping period in each T_{EHP} . The success rate of attackers using half-blind hopping strategy in SEHT is $P_{hb}(x > 0) = 1 - a[1 - \sigma wrn'_l / (\phi mn'_l + \phi mwr)]^{n_s}$.

5.2 Hopping Overhead

5.2.1 Consistency of net-flow table update

Since the flow tables need to update because of end-point and routing hopping during network communications, SEHT adopts “delete in sequential order, and add in reverse” update policy. Theorem 1 below proves the correctness of net-flow table update policy.

Theorem 1 “Delete in sequential order, and add in reverse” update policy can guarantee the consistency of net-flow tables during the process of net-flow table update.

Prove: Suppose “Delete in sequential order, and add in reverse” update policy cannot guarantee the consistency of net-flow tables during the process of net-flow table update. It indicates the presence of some packets being unable to transmitted to its destinations during the process of net-flow table update. It can be classified into four categories:

1. $hRt \notin hR_{new} \wedge hRt \notin hR_{old}$: It indicates that the forwarding routers are neither used in this hopping period nor used in the next hopping period. Therefore, this kind of forwarding routers does not receive any net-flow packets.
2. $hRt \in hR_{new} \wedge hRt \notin hR_{old}$: It indicates that the forwarding routers are just used in the next hopping period. Therefore, this kind of forwarding routers does not receive any net-flow packets in these hopping period sessions. They will only forward packets according to the updated net-flow tables in the next hopping period.
3. $hRt \in hR_{new} \wedge hRt \in hR_{old}$: It indicates that the forwarding routers are both used in this hopping period and the next hopping period. Therefore, this kind of forwarding routers will forward net-flow packets according to the corresponding net-flow table entries.
4. $hRt \notin hR_{new} \wedge hRt \in hR_{old}$: It indicates that the forwarding routers are only used in this hopping period. Therefore, this kind of forwarding routers only receives

Table 2 End-point hopping overhead

Hopping mechanism	Computational complex	Average transmission delay	Net-flow table size
Static network	$O(1)$	$t \times L_s$	n_l
OF-RHM	$O(\phi n_h)$	$t \times L_s$	$1 + n_m/n_s$
ST-RHM	$O((\gamma n_h)^2)$	$t \times L_s$	$1 + n_m m_H$
SEHT	$O((\gamma n_h)^2)$	$t \times L_s$	$1 + m_H n_m/n_a$

net flows in these hopping period sessions. What’s more, the latest time that the current forwarding routers may receive a packet after new forwarding routers are activated will less than the round-trip time between the source and destination. Before this time, the current forwarding routers will forward the packets soundly. Afterward, the routers will not receive any net-flow packets.

Consequently, the net-flow packets are accessible during the process of net-flow table update, which contradicts the assumption. “Delete in sequential order, and add in reverse” update policy proposed in SEHT can guarantee the consistency of ongoing net flows in the process of end-point hopping.

5.2.2 End-Point Hopping Overhead

The overhead of static networks, ST-RHM and SEHT hopping is shown in Table 2. It mainly consists of hopping computational complex, average transmission delay and flow table size.

Assume the number of nodes in a subnet is n_t , hEI space is n_m , and EI that can be aggregated is n_a . The size of net-flow table size in static network is n_t . Because in each hopping period, hEI is selected from all hEI set available, the size of net-flow table is $1 + n_m m_H$. While with capacity constraints, the size of net-flow table is $1 + m_H n_m/n_a$. Compared with ST-RHM, SEHT can effectively reduce the size of net-flow table.

6 Experiments and Analysis

In order to verify the feasibility and effectiveness of SEHT, we use *Mininet* [36] to build simulation network topology and adopt *Erdos-Renyi* model for random network topology generation. We choose *OpenVSwitRC (OVS)* supporting Openflow protocol [37] as HS and *OpenDaylight* [38] as RC. SEHT is deployed on *OpenDaylight* and *OVS*. Besides, Z3 SMT solver is used to solve the constraints. The configuration of source and destination nodes is shown in Table 3, where Linux CentOS 6.5 is used in web server and FTP server.

Table 3 Configuration of experimental network

Nodes	OS	V
Web server	CentOS6.5	5
FTP server	CentOS6.5	3
Clients	Windows XP	2

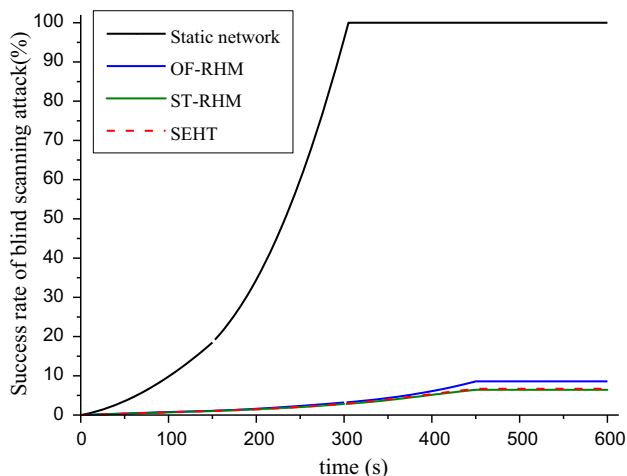


Fig. 4 Success rate of blind scanning attack strategy

Windows XP is used in client. Besides, hEI is composed of Class B IP address pool and 2^{16} size port pool. What's more, $\sigma = 5$, $\delta_1 = 0.05$, $\delta_2 = 0.075$, $\delta_3 = 0.05$, $\gamma = 0.4$, $\lambda = 0.02$, $L_{max} = 32$, $T_{LTHR} = 50$ s, and $\xi = 2.0$.

6.1 The Capability to Resist Scanning Attack

6.1.1 Capability to resist blind scanning attack

The success rate of scanning attack using blind scanning strategy is shown in Fig. 4. On the one hand, the success rate of attackers reaches 100% after spending 334 s in a static network because of using uniform active scanning. On the other hand, the success rate does not exceed 10% in OF-RHM, ST-RHM and SEHT since they deploy end-point information hopping. Furthermore, end-point hopping based on weighted value has better effectiveness in resisting scanning attack using blind reconnaissance strategy.

6.1.2 Capability to resist follow-up scanning attack

The success rate of scanning attack using follow-up strategy is mainly shown in Fig. 5. Due to the follow-up scanning strategy used in network with NMTD, the success rate is the same as that of scanning attack using half-blind strategy in a static network, which is consistent with the results of the analysis in Sect. 3.1. The success rate of attackers in OF-RHM exceeds 89% because spatial compression can be used

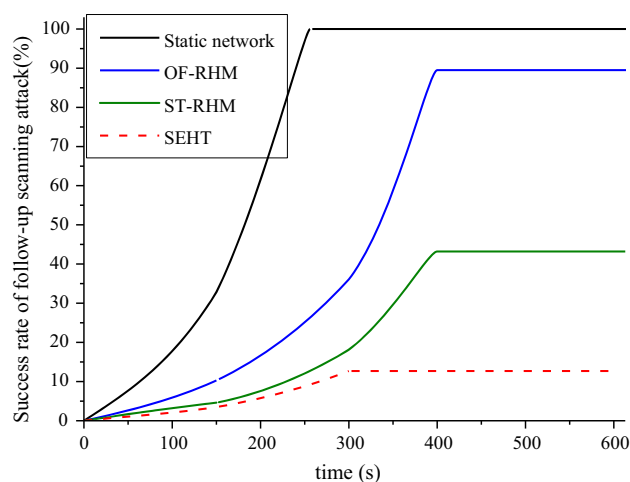


Fig. 5 Success rate of follow-up scanning attack strategy

in scanning attack using follow-up scanning strategy [14]. Despite the temporal hopping introduced by ST-RHM, it cannot adjust hopping period according to scanning frequency. As a result, it is difficult to resist scanning attacks effectively using follow-up scanning strategy. However, more than 87% of scanning attacks can be resisted by SEHT because it introduces hopping period stretch policy.

6.1.3 Capability to resist half-blind scanning attack

The success rate of scanning attack using half-blind scanning strategy is shown in Fig. 6. Because half-blind scanning is to scan specific end-point information range repetitively and unevenly, the success rate of attackers reaches 100% after spending 256 s in a static network. Besides, the success rate of attackers is enhanced in OF-RHM because only random hopping is deployed by OF-RHM. As for SEHT and ST-RHM, they can resist more than 90% of active scanning attacks using half-blind scanning strategy because hopping based on weighted value and deceiving hopping are used, respectively.

6.1.4 Capability to resist mixed scanning attack

In practical environments [22], attackers often filter EI through blind scanning. On this basis, half-blind or follow-up scanning is used in specific EI range. The success rate of mixed scanning attack is shown in Fig. 7. Since in static network, the success rate of attackers increases dramatically when the strategy changes from blind scanning attack to half-blind scanning attack. Since SEHT introduces hopping period stretch policy after discriminating follow-up scanning, it can effectively reduce approximately 29% of scanning attacks compared with ST-RHM and approximately 75% of scanning attacks compared with OF-RHM.

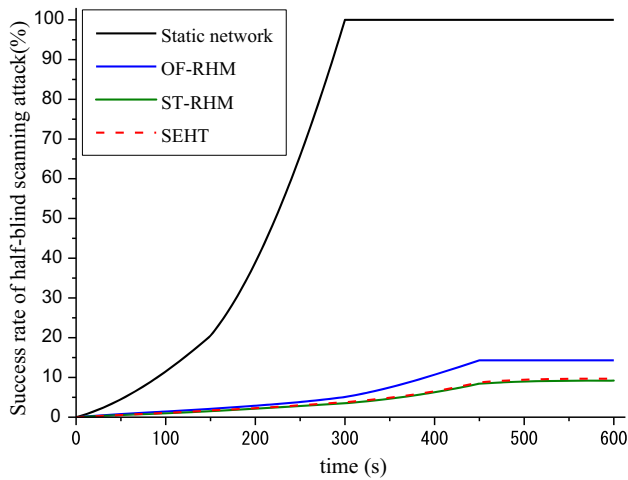


Fig. 6 Success rate of half-blind scanning attack strategy

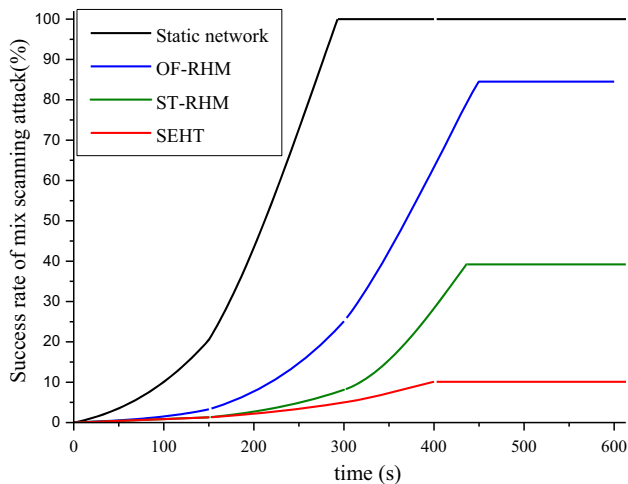


Fig. 7 Success rate of mixed scanning attack strategy

6.2 Performance Overhead Experiments

Based on the analysis of Sect. 3.2, this section mainly performs experiments on the SMT computational cost and net-flow table size. Z3 solver is used to solve the constraints in SEHT. The results are shown in Table 4, where UNSAT indicates that the SMT solver cannot find qualified solution, and FAIL indicates that the solver cannot solve the above problem under the input conditions. The above problems can be solved by weakening the constraints set. Analysis indicates that the net-flow table size change has a greater influence on SMT solving time.

Since the routing forwarding complexity in SEHT hopping is proportional to the size of net-flow table to be updated, the latter is tested in our experiment to analyze routing load caused by SEHT hopping. Figure 8 shows that SEHT routing capacity constraints can effectively reduce the size of net-flow table.

Table 4 SMT solving time

Node number	EI space	Upper bound of table size	Time (s)
100	200	100	4.87
100	300	100	5.68
100	400	100	6.77
300	400	100	UNSAT
300	400	120	FAIL
300	400	160	103.1
300	400	200	89.88

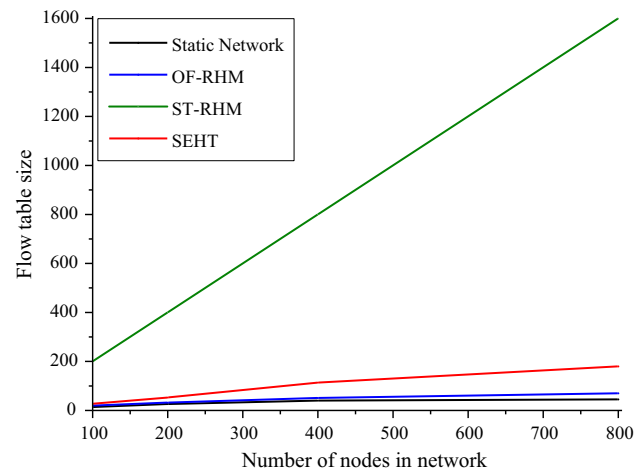


Fig. 8 Experiments of flow table size

6.3 Limitations

SEHT is a novel end-point hopping based on SDN architecture. It cannot be deployed for nodes where the IP address or Port is the primary identifier, such as an unnamed printer or server, or where the correct functionality of network nodes' services depends on the destination end-points' end-point information.

Besides, SEHT is designed based on SDN architecture, it cannot be used in traditional network architecture directly. Our future work will concentrate on how to adopt SEHT to traditional network architecture.

7 Conclusion

To counterpoise the defensive benefit of end-point information hopping and service quality of network system, a novel technique named self-adaptive end-point hopping technique based on adversary strategy awareness is proposed. Aimed at the blindness of hopping mechanism in the course of defense and the low availability caused by limited network resource and high hopping overhead, self-adaptive end-point hopping

algorithm is designed. It consists of end-point hopping triggering based on adversary strategy awareness and end-point hopping based on satisfiability modulo theory. In end-point hopping triggering based on adversary strategy awareness, hypothesis test is used to guide the choice of hopping strategy by discriminating the scanning attack strategy, which enhances the defensive benefit. In end-point hopping based on satisfiability modulo theory, satisfiability modulo theories is used to formally describe and solve the constraints of hopping, which decreases the defensive cost. Theoretical analysis compares the defensive benefit and cost of SEHT with those of static network, OF-RHM and ST-RHM in different kind of scanning attack strategy conditions. Simulation experiments show that compared with OF-RHM and ST-RHM, SEHT can disrupt approximately 90% of scanning attacks even in mixed scanning strategy. Besides, the flow table size of SEHT increases slightly with the increase in the number of nodes. Consequently, SEHT achieves the balance between security and performance.

Acknowledgements This work was supported by the National Basic Research Program of 973 Program of China (2011CB311801); the National High Technology Research and Development Program of China (863 Program) (2015AA016106); Zhengzhou Science and Technology Talents (131PLKRC644); and “Strategic Priority Research Program” of the Chinese Academy of Sciences, Grant No. XDA06010701.

References

- Jajodia, S.; Ghosh, A.K.; Swarup, V.; et al.: Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats. Springer, Berlin (2011)
- Networking F. IT Research, and D.(NITRD). Federal Cybersecurity Game-change and DThemes, [EB/OL]. <https://www.nitrd.gov/cybersecurity/page/federal-cybersecurity-1Themes>, [EB/OL]. <https://www.nitrd.gov/cybersecurity/page/federal-cybersecurity-1>
- Kewley, D.; Fink, R.; Lowry, J.; et al.: Dynamic approaches to thwart adversary intelligence gathering. In: Proceedings on DARPA Information Survivability Conference and Exposition II, 2001 DISCEX'01, vol. 1, pp. 176–185. IEEE (2001)
- Sun, K.; Jajodia, S.: Protecting enterprise networks through attack surface expansion. In: Proceedings of the 2014 Workshop on Cyber Security Analytics, Intelligence and Automation, pp. 29–32. ACM (2014)
- Evans, D.; Nguyen-Tuong, A.; Knight, J.: Effectiveness of moving target defenses. In: Moving Target Defense, pp. 29–48. Springer, New York (2011)
- Xu, J.; Guo, P.; Zhao, M.; et al.: Comparing different moving target defense techniques. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, Arizona, pp. 97–107 (2014)
- Al-Shaer, E.: Toward network configuration randomization for moving target defense. In: Moving Target Defense, pp. 153–159. Springer, New York (2011)
- Bjørner, N.; De Moura, L.: Z310: applications, enablers, challenges and directions. In: Sixth International Workshop on Constraints in Formal Verification (2009)
- Carvalho, M.; Eskridge, T.C.; Bunch, L.; et al.: Mtc2: a command and control framework for moving target defense and cyber resilience. In: 2013 6th International Symposium on Resilient Control Systems (ISRCS), pp. 175–180. IEEE (2013)
- Atighetchi, M.; Pal, P.; Webber, F.; et al.: Adaptive use of network-centric mechanisms in cyber-defense. In: 2003 Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, pp. 183–192. IEEE (2003)
- Lee, H.C.J.; Thing, V.L.L.: Port hopping for resilient networks. In: IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall, vol. 5, pp. 3291–3295. IEEE (2004)
- Dunlop, M.; Groat, S.; Urbanski, W.; et al.: Mt6d: a moving target ipv6 defense. In: Military Communications Conference, 2011-Milcom, pp. 1321–1326. IEEE (2011)
- Hari, K.; Dohi, T.: Dependability modeling and analysis of random port hopping. In: 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing (UIC/ATC), pp. 586–593. IEEE (2012)
- Kai, L.; Jia, C.; Shi, L.: Improvement of distributed timestamp synchronization. *J. Commun.* **33**(10), 110–116 (2012)
- Malathi, P.: Mitigating distributed denial of service attacks in multiparty applications in the presence of clock drifts. In: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1–6. IEEE (2013)
- Kirkpatrick, K.: Software-defined networking. *Commun. ACM* **56**(9), 16–19 (2013)
- Antonatos, S.; Akritidis, P.; Markatos, E.P.; et al.: Defending against hitlist worms using network address space randomization. *Comput. Netw.* **51**(12), 3471–3490 (2007)
- Yackoski, J.; Xie, P.; Bullen, H.; et al.: A self-shielding dynamic network architecture. In: Military Communications Conference, 2011-MILCOM, pp. 1381–1386. IEEE (2011)
- Jafarian, J.H.; Al-Shaer, E.; Duan, Q.: Openflow random host mutation: transparent moving target defense using software defined networking. In: Proceedings of the First Workshop on Hot topics in Software Defined Networks, pp. 127–132. ACM (2012)
- Jafarian, J.H.H.; Al-Shaer, E.; Duan, Q.: Spatio-temporal address mutation for proactive cyber agility against sophisticated attackers. In: Proceedings of the First ACM Workshop on Moving Target Defense, pp. 69–78. ACM (2014)
- Libo, M.; Xing, L.; Liang, Z.: On modeling and deploying an effective scan monitoring system. *J. Softw.* **20**(4), 845–857 (2009)
- Wang, Y.; Wen, S.; Xiang, Y.; et al.: Modeling the propagation of worms in networks: a survey. *IEEE Commun. Surv. Tutor.* **16**(2), 942–960 (2014)
- Badishi, G.; Herzberg, A.; Keidar, I.: Keeping denial-of-service attackers in the dark. *IEEE Trans. Depend. Secure Comput.* **4**(3), 191–204 (2007)
- Chunlei, Z.; Chunfu, J.; Chen, W.; et al.: Research on adaptive strategies for end-hopping system. *J. Commun.* **11A**, 7–57 (2011)
- Sibson, R.: Information radius. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **14**(2), 149–160 (1969)
- Yu, S.; Thapngam, T.; Liu, J.; et al.: Discriminating DDoS flows from flash crowds using information distance. In: Third International Conference on Network and System Security, 2009 NSS'09, pp. 351–356. IEEE (2009)
- Ding, Y.; Yan, E.; Frazho, A.; et al.: PageRank for ranking authors in cocitation networks. *J. Am. Soc. Inf. Sci. Technol.* **60**(11), 2229–2243 (2009)
- Cong, S.; Ge, Y.; Chen, Q.; et al.: DTHMM based delay modeling and prediction for networked control systems. *J. Syst. Eng. Electron.* **21**(6), 1014–1024 (2010)
- Collins, M.P.; Reiter, M.K.: Hit-list worm detection and bot identification in large networks using protocol graphs. In: Recent Advances in Intrusion Detection, pp. 276–295. Springer, Berlin (2007)



30. Kar, K.; Kodialam, M.; Lakshman, T.V.; Tassiulas, L.: Routing for network capacity maximization in energy-constrained ad hoc networks. In: Proceedings on INFOCOM (2003)
31. Huang, M.; Liang, W.; Xu, Z.; et al.: Dynamic routing for network throughput maximization in software-defined networks. In: IEEE INFOCOM the 35th Annual IEEE International Conference on Computer Communications, pp. 978–986. IEEE (2016)
32. Peng, B.; Kemp, A.H.; Boussakta, S.: QoS routing with bandwidth and hop-count consideration: a performance perspective. *J. Commun.* **1**(2), 1–11 (2006)
33. TUN/TAP: <http://en.wikipedia.org/wiki/TUN/TAP> (2000)
34. Lei, C.; Ma, D.; Zhang, H.; et al.: Moving target network defense effectiveness evaluation based on change-point detection. *Math Probl Eng* **2016** (2016)
35. Carroll, T.E.; Crouse, M.; Fulp, E.W.; et al.: Analysis of network address shuffling as a moving target defense. In: 2014 IEEE International Conference on Communications (ICC), pp. 701–706. IEEE (2014)
36. Lantz, B.; Heller, B.; McKeown, N.: A network in a laptop: rapid prototyping for software-defined networks. In: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (2010)
37. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J.: OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* **38**(2), 69–74 (2008)
38. Medved, J.; Varga, R.; Tkacik, A.; et al. Opendaylight: towards a model-driven sdn controller architecture. In: 2014 IEEE 15th International Symposium on, A World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6. IEEE (2014)

