Random Host Mutation for Moving Target Defense

Ehab Al-Shaer, Qi Duan, and Jafar Haadi Jafarian

Department of Software and Information Systems
University of North Carolina at Charlotte
Charlotte, NC, USA
{ealshaer,qduan,jjafaria}@uncc.edu

Abstract. Exploiting static configuration of networks and hosts has always been a great advantage for design and launching of decisive attacks. Network reconnaissance of IP addresses and ports is prerequisite to many host and network attacks. At the same time, knowing IP addresses is required for service reachability in IP networks, which makes complete concealment of IP address for servers infeasible. In addition, changing IP addresses too frequently may cause serious ramifications including service interruptions, routing inflation, delays and security violations. In this paper, we present a novel approach that turns end-hosts into untraceable moving targets by transparently mutating their IP addresses in an intelligent and unpredictable fashion and without sacrificing network integrity, manageability or performance. The presented technique is called Random Host Mutation (RHM). In RHM, moving target hosts are assigned virtual IP addresses that change randomly and synchronously in a distributed fashion over time. In order to prevent disruption of active connections, the IP address mutation is managed by network appliances and totally transparent to end-host. RHM employs multi-level optimized mutation techniques that maximize uncertainty in adversary scanning by effectively using the whole available address range, while at the same time minimizing the size of routing tables, and reconfiguration updates. RHM can be transparently deployed on existing networks on end-hosts or network elements. Our analysis, implementation and evaluation show that RHM can effectively defend against stealthy scanning, many types of worm propagation and attacks that require reconnaissance for successful launching. We also show the performance bounds for moving target defense in a practical network setup.

1 Introduction

In the current Internet architecture, network configuration parameters such as IP addresses are mostly static and easily discoverable. Although this simplifies reachability and manageability, it gives adversaries significant advantage to remotely scan networks and identify their targets accurately and quickly using off-the-shelf scanning tools [1,2]. Despite firewall deployment, most enterprise networks have many public and private hosts accessible from outside. Using the

A.D. Keromytis and R. Di Pietro (Eds.): SecureComm 2012, LNICST 106, pp. 310–327, 2013.

existing dynamic IP assignment techniques like *DHCP* does not protect from scanning, and using *NAT* makes it difficult to reach legitimate hosts remotely. In addition, these techniques are insufficient to provide proactive countermeasure because the IP mutation is infrequent and traceable.

In this paper we propose a novel proactive moving target defense, called *Random Host Mutation (RHM)*, that challenges the principal assumptions of scanning adversaries in cyber warfare: "if you can scan it (*i.e.*, a response received), you can find it. Otherwise, it is an unused address". We propose a mutable network architecture that mutates IP addresses of designated moving target (MT) hosts randomly and frequently so that the attackers' premises about the network fail. The goal of these mutations is to make the hosts untraceable via network reconnaissance attacks. However, developing an efficient and practical scheme that can be deployed on general networks requires careful consideration of tough challenges: (1) IP mutation must be transparent to the end-host to prevent disruption of active connections, (2) the integrity of end-to-end Internet reachability should be maintained, (3) IP mutations should be fast and unpredictable to deceive scanners by optimally using the whole available address range, (4) IP mutations should avoid service interruptions, routing inflation, delays and security violations, (5) RHM should be seamlessly deployed in any existing networks without requiring any changes in the end-host or network infrastructure.

RHM addresses each of these challenges and develops an optimized moving target defense architecture that maximizes the uncertainty on the adversary discovery, while satisfying the configuration management constraints. To keep the IP addresses of end-hosts unchanged, RHM creates routable short-lived virtual IP addresses (vIP) that will be changed randomly, consistently and synchronously in the network to allow unpredictable, yet safe mutation. The vIP addresses will be used for routing and are automatically translated into the real IPs (rIP) and vice versa at the network edges (subnet) close to the destination. Using rIP and vIP addresses allows for separating network administration and mutation management, making mutation transparent to administrators and end-host configuration. Under RHM architecture, a MT host is reachable by a name that is then resolved to a vIP address. However, scanners do not often query DNS for scanning networks because (1) it increases detection probability [3], and (2) not all hosts names are necessarily known by scanners or DNS. Although users commonly use named servers to reach their destinations, RHM allows only authorized users (e.g., administrators) to reach MT hosts based on policy-based access control predefined by RHM managers for each MT host.

To optimize IP mutation, the mutant vIPs are selected randomly from the entire unused address space in the network in order to increase unpredictability while satisfying various mutation speed requirements of different MT hosts, routing table size bound, routing convergence, and network operation integrity. We formulate this problem as a constraint satisfaction problem and solve it using Satisfiability Modulo Theories [4] (SMT) solvers. To allow for the maximum use of unused address space for mutation while considering routing convergence, RHM employs two-phase mutation: (1) *low frequency mutation (LFM)*

that solves the constraint satisfaction problem to select an optimal assignment of MT hosts to random mutation range of vIPs, and (2) *high frequency mutation (HFM)* that uses a cryptographic random function to select from a designated range a specific mutant vIP randomly, yet synchronously across RHM components in the network. In both mutations, active sessions are maintained.

The RHM architecture was implemented and tested in our university campus, and comprehensive evaluation were conducted to study the effectiveness and limitations of RHM. Our theoretical analysis, simulation and experimentation results show that RHM can protect up to $40 - 90\%$ of the network host from reconnaissance attacks lunched by scanning tools or vicious random scanning worms.

Previous works [5,6,7,8] propose techniques to allow for changing or hiding IP address using consecutive DHCP updates [5], encrypting headers [6], translation [7], or rerouting to another server [8]. These solutions are very limited as they do not support wide range of IP mutation. Also, they are not readily deployable solutions as they do not address the challenges discussed before.

The rest of the paper is organized as follows. Section 2 describes the related works. Section 3 presents the formulation and algorithms for RHM. In Section 4 the RHM architecture and protocol are described; Section 5 describes the required re-configurations. Section 6 presents implementation, analysis and evaluation, and Section 7 concludes the paper.

2 Related Works

A few research proposals on dynamically changing IP addresses for proactive cyber defense have been presented in the literature. The APOD (Applications That Participate in Their Own Defense) scheme [8] uses *hopping tunnels* based on address and port randomization to disguise the identity of end parties from sniffers. However, this approach is not transparent as it requires cooperation of both client and server hosts during the IP mutation process.

The DyNAT provides a transparent approach [9] for IP hopping by translating the IP addresses before packets enter the core or public network in order to hide the IP address from man-in-the-middle sniffing attacks. Although this technique will make network discovery infeasible for sniffers, it does not work for scanners who rely on probe responses for discovering the end-hosts.

A network address space randomization scheme called NASR [5] was proposed to offer an IP hopping approach that can defend against hitlist worms. NASR is a LAN-level network address randomization scheme based on DHCP update. NASR is not transparent to the end-hosts because DHCP changes are applied to the end-host itself which results in disruption of active connections during address transition. Moreover, it requires changes to the end-host operating system which makes its deployment very costly. Also, NASR provides very limited unpredictability and mutation speed because its IP mutation is limited on the LAN address space and will require DHCP and host to be reconfigured for this purpose (the maximum IP mutation speed is once every 15 minutes).

A technique called OF-RHM (OpenFlow Random Host Mutation) was proposed in [10]. OF-RHM offers an IP mutation technique for software-defined networks. Although the technique is transparent to end-hosts and provides high mutation rate, it is not deployable on traditional networks.

Yegneswaran *et al.* [11] and Cai *et al.* [12] present techniques for defending honeynets from systematic mappings that aim at differentiating live IPs from monitored ones and blacklisting monitored IPs for efficient target selection. RHM completely wipes out systematic mapping attacks, because generated blacklists are only valid for a relatively short interval.

In summary, none of the previous techniques provides a deployable transparent mechanism for IP mutation that can defend against external and internal scanning attacks without changing the configuration of the end-hosts. RHM implement an efficient IP mutation in term of unpredictability, mutation speed and configuration management. Unlike the previous techniques, RHM uses the entire address space to increase unpredictability and updates configurations at real-time while preserving network operation integrity.

3 Host Mutation Optimization

Maximizing mutation unpredictability and mutation speed are primary objectives of RHM. To achieve the first goal, RHM uses the maximum portion of unused address space for mutation. However, achieving the second goal is limited by the routing convergence time and table size bounds. Thus increasing the mutation speed implies bounding the mutation space to local ranges. To satisfy these conflicting objectives, RHM uses two levels of random mutation granularity: *Low Frequency mutation (LFM)* and *High Frequency mutation (HFM)*. LFM is used for selecting a random network address, denoted as *virtual address range (VAR)* for the MT hosts, and HFM is used to select a random vIP within VAR assigned during LFM. Combining the two levels of mutation, enhance not only the mutation unpredictability and speed but also the network manageability.

The duration of an LFM or HFM is called an LFM or HFM interval, respectively. An LFM interval contains multiple HFM intervals, and in every HFM interval the MT host will be associated with a unique vIP from the designated VAR of that particular host. Since LFM is more expensive than HFM, LFM interval is fixed, while the HFM interval is customized based on the required mutation speed of each MT host. To maintain connectivity with MT hosts, MT hosts engaged with active sessions will retain their vIPs in addition to new ones during mutation. Therefore, a MT host might be associated with more than one vIP simultaneously.

In the following, we describe the main phases of RHM algorithm: (1) generation of unused *VARs*, (2) LFM for optimal assignment of VARs to MT hosts, (3) VAR segmentation, and (4) HFM for random and synchronized vIP selection within the allocated VARs for each MT host. The configuration management and session tracking for active connections will be discussed in subsequent sections.

Table 1. Description of parameters

b_{ij}	denotes whether range r_j is assigned to host h_i ($b_{ij} \in \{0,1\}$)
$\{h_1, \ldots, h_n\}$	set of MT hosts
$\{r_1, \ldots, r_m\}$	set of VARs
B_{kj}	denotes whether range r_j is assigned to at least one of the host in subnet S_k ($B_{kj} \in \{0,1\}$)
E_i	The expected value of vIP repeat probability in LFM interval for host h_i
F_i^p	set of address ranges used by those hosts similar to host h_i in the last p LFM intervals
N_i	The number of vIP mutations in an LFM interval for host h_i
S^l	set of address ranges uses by any host in the previous l LFM intervals
$\{S_1, \ldots, S_z\}$	set of subnets
T_d	maximum routing update propagation delay
$R_i = 1/T_{HFM_i}$	mutation speed of host h_i
T_{LFM}	length of an LFM interval
U	upper bound for routing table size
V_i	minimum required address space for h_i

3.1 VAR Generation

The first step of each LFM interval is to generate unused address blocks (VARs) in the network address space. The unused address space is defined as the address space that includes rIPs and vIPs that are currently in-use for active sessions. Given used address ranges $A_1, \ldots, A_u$ of the network and vIPs, $q_1, \ldots, q_k$ vIP addresses used in currently active sessions, we can generate contiguous blocks of unused address space by simply masking the full network address space A as follows:

$$\{r_1, r_2, .., r_m\} \leftarrow A \wedge \neg(A_1 \vee \ldots \vee A_u \vee q_1 \vee \ldots \vee q_k) \quad (1)$$

We implemented this by encoding A, $A_1, \ldots, A_u$, and $q_1, \ldots, q_k$ as Boolean expressions using Binary Decision Diagram (BDD) [13] operations.

In addition, LFM will require sufficient unused address space to allow for swapping VARs during mutation. This means that the unused address space should be at least twice the total mutation space required by all MT hosts (formally, $2 \sum_{1 \le i \le n} V_i \le \sum_{1 \le j \le m} |r_j|$, where V_i is the minimum required address space for MT h_i).

3.2 LFM Formulation

The core problem of LFM is to assign VARs to MT hosts at each interval such that (1) mutation unpredictability can be maximized, (2) mutation speed, and (3) routing table size constraints are satisfied. Suppose we currently have a set of MT hosts $\{h_1, \ldots, h_n\}$, VARs $\{r_1, \ldots, r_m\}$, mutation rate (R_i) for each host h_i, the expected value of vIP repeating probability (E_i) for each h_i, maximum routing convergence time T_d, and the upper bound for the routing table size

(U). Each host belongs to a subnet in the set $\{S_1, ..., S_k\}$, where subnet is a group of hosts that are physically connected through a switch. We can then formulate LFM constraint optimization problem using the following SMT-based (Satisfiability Modulo Theories [4]) formulas:

The following is the description of these constraints. Table 1 describes the important parameters of our formalization.

VAR Allocation Constraint: Eq. 2 is to guarantee that at least one VAR must be assigned to each MT host.

Unpredictability Constraint: Eq. 3, 4, and Eq. 6 are used to maximize unpredictability during LFM and HFM, respectively. Eq. 3 is to guarantee that VARs used in the past l intervals (S^l) will not be repeated for *any* host during the coming LFM interval. Similarly, Eq. 4 is to avoid using the same VAR that has been used by another host with similar characteristics in last p intervals (F_i^p represents the list of VARs used by hosts similar to h_i). This is important to countermeasure fingerprinting attacks by preventing scanners from utilizing vulnerability information discovered for another host. A longer interval assures that, similar hosts share vIPs less frequently. Users can increase l and p (usually $p > l$) to achieve the desired level of unpredictability.

$$\sum_{1 \le j \le m} b_{ij} \ge 1 \tag{2}$$

$$b_{ij} = 0, \text{if } r_j \in S^l \tag{3}$$

$$b_{ij} = 0, \text{if } r_j \in F_i^p \tag{4}$$

$$\sum_{1 \le i \le n} b_{ij}\ V_i \le |r_j| \tag{5}$$

$$V_i \ge \frac{(N_i - 1)}{2E_i} \tag{6}$$

$$b_{ij} \le B_{kj}, \forall h_i \in S_k \tag{7}$$

$$\sum_{1 \le k \le z} \sum_{1 \le j \le m} B_{kj} \le U \tag{8}$$

$$b_{ij}, B_{kj} \in \{0, 1\}, 1 \le i \le n, 1 \le j \le m, 1 \le k \le z$$

Mutation Speed Constraint: RHM allows each MT host to specify the target mutation rate (mutation per second) it requires based on its security requirements. During each LFM interval the size of allocated VARs should be sufficient

for the mutation rate of each and all moving hosts. Each MT host h_i has a mutation rate R_i, based on which the HFM interval of the host, T_{HFM_i}, is calculated: $T_{HFM_i} = 1/R_i$. Also, during an LFM interval all the vIPs of a host are selected from the same VAR. T_{LFM} is greater than the maximum routing convergence time: $T_{LFM} > T_d + \delta$, where T_d is the routing convergence time and δ is the LFM planning time.

Thus total number of vIPs selected by RHM during an LFM interval will be $N_i = \lceil T_{LFM}/T_{HFM_i} \rceil$. We can then calculate the probability of repeating a vIP for a MT host after selecting j^{th} vIP as $P_i = (j-1)/V_i$, where V_i is the minimum required size of the VAR associated with a host. Therefore, we can then calculate the *expected value* of the repeating probability in HFM interval as follows:

$$E(P_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{j-1}{V_i} = \frac{N_i - 1}{2V_i}$$

where $1 \leq i \leq n$. Therefore:

$$E_i \geq \frac{N_i - 1}{2V_i}$$

The constraint in Eq. 6 is to guarantee that V_i has minimum addresses required to ensure that the expected value of P_i will not exceed the expected threshold (E_i) associated with this host. Since a VAR can be assigned to more than one MT host, Eq. 5 is used to ensure that VAR size ($|r_j|$) is large enough to accommodate MT hosts sharing the same VAR, r_j.

Routing Table Size Constraint: We should minimize the routing table size incurred by the VAR assignments. To this aim, one should assign those hosts that are in the same subnet with VARs that have the same prefixes. We define B_{kj} as a Boolean parameter (that is, $B_{kj} \in \{0, 1\}$) to indicate if range r_j is assigned to at least one host in subnet S_k. Eq. 7 denotes that if a range r_j is assigned to a host h_i ($b_{ij} = 1$) in subnet S_k, then the routing entry for r_j must be added to the total routing entries of the subnet.

Eq. 8 constraint is used to bound the number of distinct VARs assigned to different subnets, S_k. This consequently implies assigning minimum number of VARs to moving hosts that are in the same physical subnet to minimize the routing table size (supernetting or route aggregation).

3.3 VAR Segmentation

RHM allows more than one MT host to share the same VAR in order to optimize the use of VAR spaces and allow for maximizing the possibility of supernetting for MT hosts in the same subnet. To avoid address collision within a VAR, participating MT hosts will be eventually allocated non-overlapping ranges within the shared VAR. Since a VAR r_j may be assigned to multiple MT hosts,

$\sum_{1 \le i \le n} b_{ij} = p$ means that r_j is allocated to p MT hosts. So r_j must be divided into p separate sub-VARs proportional to the V_i requirement for each MT host. The assignment of sub-VARs is randomized to minimize the possibility that the same host uses the same sub-VAR in two consecutive LFM intervals.

3.4 HFM Formulation

Each host h_i has a specific HFM interval T_{HFM_i}, which is determined based on the security requirements of the MT host. To achieve synchronization in HFM, every MTG of the network will use a pre-established hash function H and a shared key K to compute virtual addresses for all moving hosts in its subnet. The shared key is distributed by the MTC. Suppose there are p available vIPs $\{a_1, a_2, \ldots, a_p\}$ for host h_i in the current LFM interval, then the MTG can compute the vIP of HFM interval I_j of MT host h_i as:

$$A(I_j, h_i) = a_{(H(K,j) \bmod p)+1} \tag{9}$$

Here j is the index of the current HFM interval which can be calculated from the mutation speed of the MT host. The *mod* operation guarantees that the computed address index fall into the valid range between 1 and p. The randomness (or unpredictability) of the vIP mutations in VAR is guaranteed by the randomness of the hash function. However, Eq. 6 guarantees that for a host h_j, even in case of uniform vIP selection, the repeat probability never surpasses E_i.

This synchronization of MTGs is not precise time synchronization. Instead it is a loose synchronization that is realized via sharing of K, mutation index j and the designated VARs of MT hosts. The sharing allows each MTG to compute the active vIP addresses of every MT host in the network. In the case when a MTG crashes, it can still get the shared key and mutation index from the MTC to resume the IP mutations of the MT hosts within its subnet.

4 RHM Architecture and Protocol

4.1 Architecture

The main architecture of RHM network is depicted in Figure 1. The tasks of assigning a VARs to MT hosts (Sections 3.1, 3.3 and 3.2) are performed by a MTC. At each LFM interval, MTC selects new VARs for each MT hosts such that it satisfies constraints in Section 3. Then, the new designated VARs are announced to MTGs, which are boxes deployed at the boundary of subnets (between subnet switch and the core).

Each MTG is responsible for management of MT hosts in one subnet. MTG has various functions. Firstly, it selects a vIP from the current VAR of a MT host using a cryptographic function and a secret random key to guarantee unpredictability and intractability (Section 3.4). Secondly, it translates source rIP to vIP for outbound, and destination vIP to rIP for inbound packets. MTG stores the mapping between rIP and vIP in a translation table and performs address

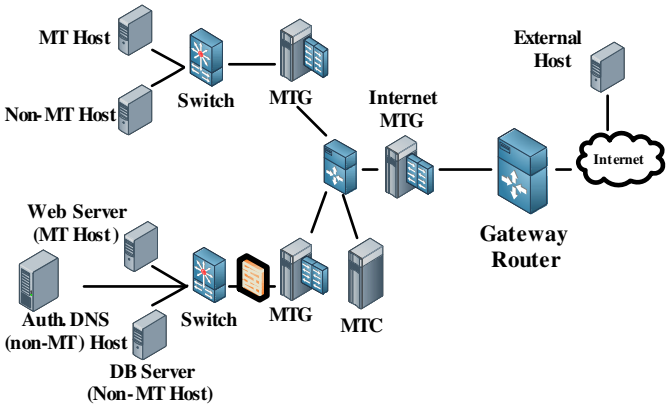

Fig. 1. The Architecture of moving target network

translation for incoming and outgoing packets. Active connections using old vIP will be maintained by storing the information of forwarding flows in the connection table. MTG will forward packets from old connections until the session is terminated (*e.g.*, FIN for TCP) or expired (*i.e.*, long inactive time for both TCP and UDP). Thirdly, it advertises routing updates of assigned VARs (for MT hosts in its subnet) by using the appropriate Interior Gateway Protocol (Section 5). Finally, it is responsible for changing DNS responses of local authoritative DNS servers (Section 4).

In addition to VAR selection, MTC is responsible for management of MTGs, key distribution for HFM vIP selection, and authorization of rIP-based flows (Section 4).

4.2 Protocol

There are two ways to communicate with MT hosts: using host name or host rIP. These two scenarios are depicted in Figures 2 and 3, respectively. These figures show a scenario where a MT host communicates with another MT host. Other scenarios (*e.g.*, non-MT host communicating with a MT host) are special cases of this scenario.

Figure 2 shows that when a DNS query is sent to resolve the name of an MT host, the DNS response is intercepted by the MTG and the rIP of the MT server is replaced with its current vIP (steps 1-3). Moreover, the MTG also sets the TTL value in the DNS response according to the HFM interval. As a result, clients will receive the vIP mapping to MT host name and initiate their connections accordingly (steps 3-4).

Figure 3 shows how authorized users (*e.g.*, administrators) can reach MT hosts using rIPs. In this case, MTG will request and authorize access for this source from MTC (steps 1-4). If access is granted, the MTG of the source will translate the rIP of the destination to the corresponding vIP and update its translation table

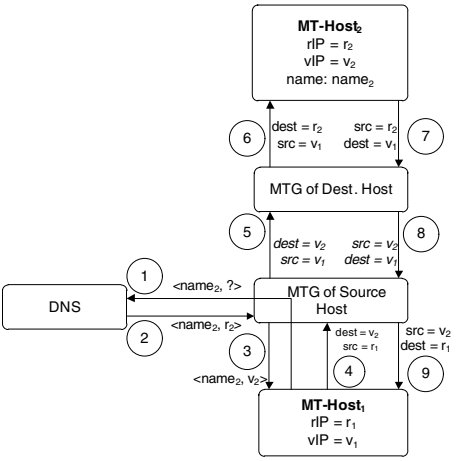

Fig. 2. hosts communicating with moving hosts through name

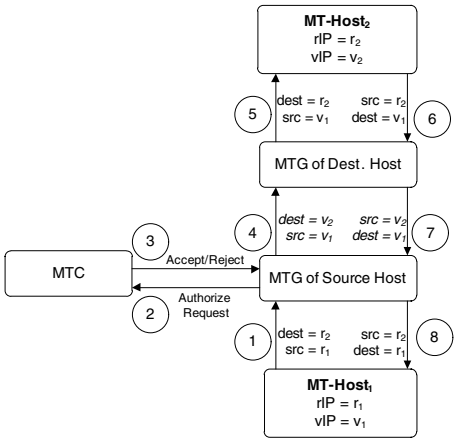

Fig. 3. hosts communicating with moving hosts through IP address

accordingly. It is important to note that this authorization is performed once per every session that includes rIP as destination. MTC access control policy can be managed by administrators based on the criticality of the MT host. In both scenarios (access by name or rIP) the source rIP is always translated to vIP.

As a result, RHM protocol restricts routing to vIP destinations in order to (1) ensure MT host mutation in the network, and (2) filter out traffic destined to rIPs and inactive vIPs that can be generated randomly by internal scanners at source MTG.

5 Reconfiguration Planning

RHM deployment does not require any change to current Internet infrastructure. In this section, we describe required reconfigurations that must be considered for deployment of RHM in current setting of Internet infrastructure.

5.1 Session Tracking

In order to prevent the disruption of active sessions, the MTG stores the rIP-vIP mapping of each flow in its translation table, and does not delete them until the termination of the flow. Active sessions continue using their vIPs without any disruption, and MTG handles their packets based on the translation table. The vIP is evicted from the available unused address space, and thus will not be assigned to any other MT host. However, the MT host will be assigned a new VAR that will be used for the next HFM. Therefore, an MTG might keep multiple vIP entries for the same MT host in its translation table in order to handle old and new active sessions.

5.2 DNS

Ideally, the TTL values of MT host DNS records should be set to be not more than the T_{HFM_i}. However, setting a small TTL values (order of seconds) might generate high volume of DNS traffic to the authoritative DNS. On the other hand, higher TTL values will increase DNS cashing but result in decreasing of the mutation speed in HFM. Therefore, this trade-offs between mutation speed and volume of external DNS traffic can be adjusted by administrators to satisfy particular network requirements.

5.3 Access Control Devices

Figure 1 shows possible locations of firewall devices in the network. For firewalls located behind MTG, no changes are required because only rIP is seen at this end. Firewalls that are in front of MTGs need to be reconfigured to be consistent with recent vIP changes. However, from a practical perspective, firewall polices that are in front of MTGs usually use domain/subnet ranges instead of specific IP addresses.

Thus a simple approach is to use default-accept in these firewalls for only the unused address space leaving the responsibility of filtering out the actually unused address space to MTGs. Since allocated VARs are strictly from the unused address space, this will not overlap with any of the existing rules in the firewall. This simply implies delegating the filtering out of the traffic destined to the unused addresses to MTGs, which eliminates spurious traffic by discarding any traffic not destined to an active vIP [14].

We assume that IDS/IPS devices are deployed behind the MTGs which is practically sound for most networks. Moreover, MTG bypasses hosts that use IPSec traffic because they are inherently protected from scanning attacks by IPSec gateways.

5.4 Routing

For implementation of RHM, no change is required on gateway and other external routers, because they simply route the traffic to/from our network. Routing updates must be advertised for internal routers. To address the routing convergence time, which is relatively small, MTC pre-computes VAR assignments one LFM interval before using them for mutation. Therefore, routing updates can be propagated in a timely and conflict-free manner.

For non-MT hosts using real IP address no routing update is required. MTC delegates routing update responsibility to MTGs, because they act as the gateway between routers and subnets. MTC informs MTGs of the next set of designated VARs for their MT hosts. MTGs generate initiates and broadcast advertisement messages as new VARs being assigned. If authentication is required, MTG will be given the credentials to authenticate itself to routers in the network. MTG triggers updates both periodically (required for most interior gateway protocols), and upon receiving new VAR assignments from MTC.

5.5 Implementation and Deployment

To study and demonstrate the feasibility of RHM, we implemented a proof-of-concept for RHM in a designated class C subnet in our university campus network. The network is further divided into 3 subnets each containing up to 3 MT hosts. The MTG and MTC components are implemented on Linux-based (Ubuntu) boxes and given privilege to interact with RIP-2 based routers and local firewalls. One RHM subnet includes an *Apache* Web Server, an *Apache* FTP server, and an OpenSSH server that reside on different MT hosts. To update routing information, MTC pre-computes and distributes VAR assignments to all MTGs for every LFM interval. MTG boxes implement RIP-2 protocols to communicate and advertise VARs to routing devices.

Our implementation proved that the RHM approach is feasible. We run several network activities during mutation: downloading files from FTP server and SSH server, video streaming from the HTTP server, and web browsing. Availability of these surfaces was not affected and long-lived connections functioned soundly and accurately, even after numerous LFM intervals. The routing propagation convergence was fast and the delay was negligible (less than 60 seconds). This shows that RHM is deployable and manageable on real networks. However, the implementation may not measure the scalability of the approach, since scalability evaluations require thousands of network elements. For this purpose, to show the effectiveness and scalability of RHM approach we performed analytical studies and simulation experiments, and we provide this result in Section 6.

6 Analysis and Evaluation

In this section, we evaluate RHM effectiveness against attacks and the overhead it incurs on the network. We use analytical modeling, experimentation and simulation to evaluate RHM.

6.1 Effectiveness

We evaluate the RHM effectiveness against scanning external and internal scanners.

External Scanners. The prolonged interval between target discovery and attack allows RHM to mutate the vIP of the scanned host before the actual launching of the attack. RHM can prevent information gathering by external scanners, which may be used for various purposes including hitlist attacks effectively, since the IP addresses in the hitlist will be soon out-of-date. Due to high mutation speed, and unpredictability of vIP assignments, our solution will be the *optimal* solution for defense against hitlist worms. To show the effectiveness of RHM against hitlist attacks, we run 100 different *Nmap* scanning over 90 minutes for a class B RHM network of up to $10 - 20\%$ MT hosts. Then, after comparing all the hundred scanning reports with the ground truth we found not more than 3% of actual IP addresses has been discovered, as shown in Figure 4.

Internal Scanners. Internal network scanning is performed via sequence of probes sent to random IPs, usually by random scanning worms. We can further classify random scanning worms into two categories: the first category is non-repeat random scanning worms, which never repeat addresses that have been scanned before. This can be achieved by some periodic pseudo-random generators or more sophisticated cooperative scanning approaches such as divide-and-conquer, or sequential scanning [15]. The second category is repeatable random scanning worms, which may choose a repeated address during random scan. In this section, we study the effectiveness of RHM on random scanning worms using the following two metrics: (1) *The mutation success probability*: the probability that a host is not hit by a scanner; and (2) slow-down rate of worm propagation: the total infection time with and without RHM.

Mutation Success Probability: Suppose there are N addresses in the available address space of the MT host and the MT host will use a random address from the address space in any HFM interval. Assume a non-repeat uniform scanner that is scanning an RHM network. We define speed ratio k as scanning rate of scanners on mutation speed of MT host. It can be shown that for N scans and $k = 1$, the scanner will miss the target with probability

$$P_{miss} = \left(1 - \frac{1}{N}\right)^N \approx e^{-1} = 0.37 \tag{10}$$

Given k and N, for a non-repeat random scanner, the scanner will miss the target with the following *mutation success* probability:

$$M = \prod_{j=0}^{\lfloor \frac{N-k}{k} \rfloor} \left(\frac{j \cdot k}{N} + \frac{N - j \cdot k}{N} \prod_{i=0}^{k-1} \left(1 - \frac{1}{N - j \cdot k - i}\right)\right)$$

Figure 5 shows the theoretical and simulated mutation success probability of the moving hosts with $N = 30000$, 60000 and 120000 and different k values for the non-repeatable scanners. The scanner makes a total of 30000 scans, which means the scan ratios are 1, 0.5, 0.25 respectively in the three cases. In the simulation, every data point is the average of 10 runs, and there are 10% MTs in the network. The mutation success probability is the percentage of the MTs that are not infected at the end of the simulation. We can see that the simulated result is roughly consistent with the theoretical analysis. We can also see that the mutation success probability is stable when k is less than some threshold. If the scanner can scan the whole mutation address space, the mutation success probability can reach a maximum value about 0.4. If the scanner cannot scan the whole space, the mutation success probability can be much higher than 0.4. When the scanner can only scan one quarter of the address space, the mutation success probability can reach about 0.8.

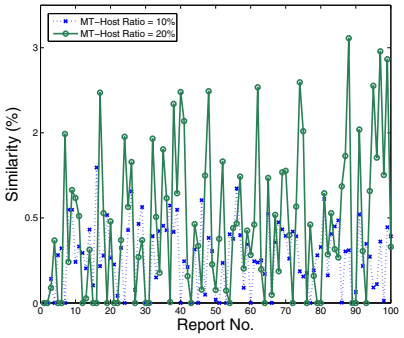

Fig. 4. Ratio of common IP addresses between Nmap reports

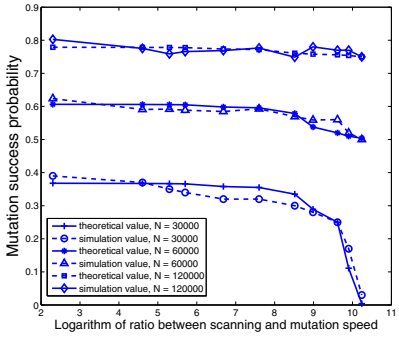

Fig. 5. Mutation success probability of a MT host for non-repeatable scanners

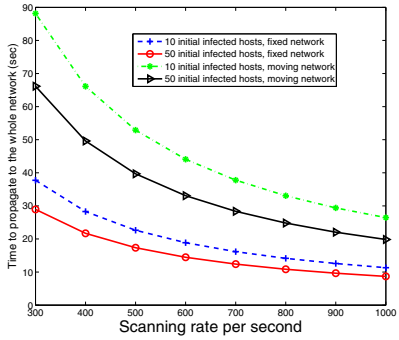

Fig. 6. Worm propagation speed comparison for fixed and moving networks

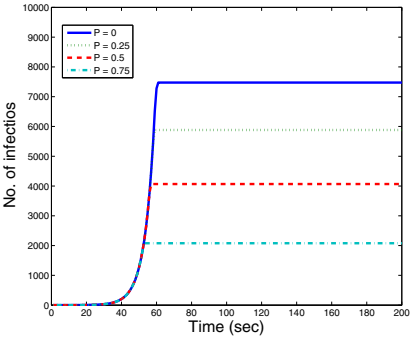

Fig. 7. Worm propagation in RHM network with various safe-area migration probabilities

For a repeatable uniform scanner, if a host uses a fixed or moving address and the scanner uses a random address from a set that has N possible addresses for every scan, then the scanner may hit the host with probability $1/N$ for every scan. In other words RHM has no effect on the repeatable scanners. For routing worms [16], the worm scanning space is determined based solely on BGP routing data, and the RHM effect is similar to uniform scanners.

Slow-Down Rate of Worm Propagation: The ideal case of non-repeatable scanning can be achieved via cooperative scanning, such as divide-and-conquer scanning [15]. Based on our analysis of non-repeat scanner in equation 10, a cooperative will miss about e^{-1} (more than one third) portion of the vulnerable hosts (this can be considered to be equivalent to that the whole network only contains $e^{-1}V$ vulnerable hosts). Also, the propagation speed will also decrease because the total number of hittable vulnerable hosts decreases.

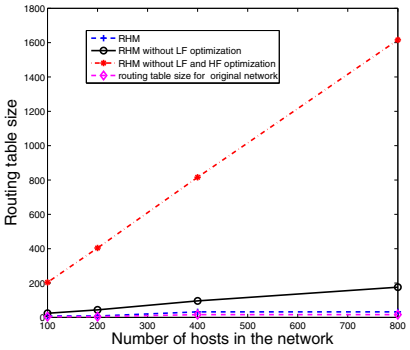

Fig. 8. Routing table size for different RHM settings

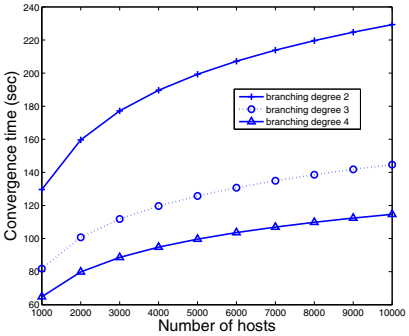

Fig. 9. Routing convergence for different network sizes

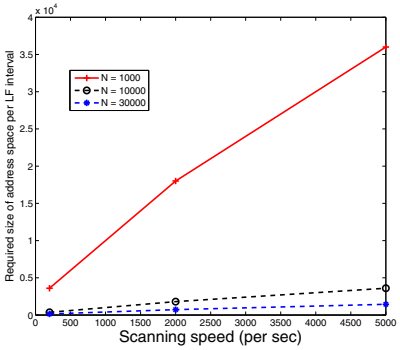

Fig. 10. Address space requirement

U	T (sec)
50	0.28
45	0.29
40	0.9
35	1.52
30	UNSAT

Fig. 11. Case 1

U	T (sec)
160	87.94
150	109.25
140	FAIL
130	UNSAT

Fig. 12. Case 2

Based on [15], in a moving target network, the propagation model for an ideal non-repeat cooperative scan worm is

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \begin{cases} \frac{(1-e^{-1})\eta}{N} V I(t) & I(t) < V \\ 0 & I(t) = V \end{cases} \tag{11}$$

The solution of the equation is:

$$I(t) = I(0)e^{a(1-e^{-1})t} \quad \text{for } I(t) < (1-e^{-1})V \tag{12}$$

Here $a = \frac{\eta V}{N}$ and $I(0)$ is the number of infected hosts in the beginning. We calculated the worm propagation speed based on the above analysis. Fig. 6 shows the time (in seconds) for the worm to propagate a class A network (2^{24} total addresses) with 10000 vulnerable hosts.

We can see that with RHM, worm propagation takes about two times more than usual time. This means that RHM can slow down the worm propagation

significantly. We also integrated IDS feedback in mutation decision, in order to move MT hosts to scanned safe area with a probability P. Figure 7 shows that the mutation success probability will be improved to 40 to 80% with IDS feedback.

6.2 Overhead Evaluation

In this section, we evaluate (1) the required address space overhead with varying scanning rate, (2) the computational complexity of the constraint satisfaction solution, and (3) the routing and firewall overheads.

Address Space Overhead. The required address space necessarily depends on mutation speed. Similarly, the HF and LF mutation speed is dependent on targeted attack model.

To maximize the defense benefit of RHM, enough addresses for an LFM interval should be provisioned. For example, based on our analysis for non-repeat uniform scan worm in Section 6.1, if we want to keep the mutation success probability to be over 0.3, then the size of the address space assigned to a single MT host for an LFM interval should be at least

$$(\text{scanning speed}/P_{0.3}) \cdot T_{LFM} \tag{13}$$

Here $P_{0.3}$ is the mutation speed that can achieve mutation success probability 0.3 for this scanning speed. Figure 10 shows the required size of address space with network size $N = 1000$, 10000 and 30000.

SMT Formalization. We also tested the feasibility of the constraint satisfaction algorithm for LFM VAR assignment. We use the Z3 SMT solver [4] for our evaluation. The running time of the SMT instance is very sensitive to the selection of the upper bound U of the routing table size in Eq. 8. Figure 11 shows the running time of the SMT formalization for a network with 100 moving hosts, 40 empty address ranges, while the demand of every host is a random number between 1 and 5, the size of the empty ranges is a random number between 10 and 20. In the table, "UNSAT" means the SMT solver reported that the instance is unsatisfiable. In this case one must relax some of the parameters (such as decreasing the routing table size upper bound or increasing the size of empty address ranges) to get a feasible solution.

Figure 12 shows the running time of the SMT formalization for a network with 300 moving hosts, 120 empty address ranges. In the table, "FAIL" means the SMT solver failed to solve the instance. This means that the upper bound used in the constraint is beyond the solving ability of the SMT solver. In this case one also needs to relax some of the parameters to get a feasible solution.

Routing and Firewall Updates Overhead. The overhead of routing update is proportional to the routing table size after every LF mutation. Suppose the total number of hosts in a subnet is H_t, and the number of moving hosts in the subnet is H_m. We also assume that in the LFM optimization algorithm we can arrange N_h moving hosts with the address ranges that have the same prefix (route summarization). Then we can see that the total number of routing entries

needed for the subnet in one LFM interval is $1 + (H_m/N_h)$. Here (H_m/N_h) are the routing entries required for the moving hosts, and we need to add 1 entry for all fixed hosts in the same subnet. If no LFM optimization is adopted, then the total number of routing entries needed for the subnet in one LFM interval will be close to $1 + H_m$, because RHM will assign different address blocks for different moving hosts, which requires a different entry in the routing table. If there is no LFM and HFM optimization and the motion is completely random, and there are on average N_{HFM} HFM intervals in a LFM interval, then the total number of routing entries needed for the subnet in one LFM interval will be close to $1 + H_m \cdot N_{HFM}$, because we need to use a different entry in the routing table for every host in every HFM interval.

We simulated RHM in networks with various sizes, ranging from 100 to 800 hosts. The networks contain up to 16 subnets and every subnet contains up to 50 hosts respectively. Fig. 8 shows the routing table size for every LFM interval for different kinds of RHM setting. We can see from the figure that LFM and HFM optimization can greatly reduce routing table size.

We also simulated the routing convergence time for different network sizes. We assumed each subnet includes 50 hosts, and the network uses *RIP* for routing advertisements. In RIP, each router broadcasts its routing table every 30 seconds. We assumed the routers form a full tree and simulated routing convergence time for branching degrees 2, 3, and 4. As represented in Figure 9, the convergence time for branching degree 2 and a network including 10000 hosts is less than 4 minutes.

For firewall updates, the analysis and results are similar. The firewall updates occurs for new VARs in each LFM interval, and the firewall updates are basically equal to routing updates. The only difference is that for firewalls we have to evict old entries, while in routers unadvertised destinations will be excluded after a certain timeout interval.

7 Conclusion and Future Work

Moving target is a game changing technique that puts the defender in a stronger position with proactive rather than reactive defense. In this paper we present a novel framework called Random Host Mutation (RHM). We formulated intelligent host randomization with constraint satisfaction problem to achieve high unpredictability and speed while satisfying routing and configuration constraints. We implemented RHM in existing network without requiring changes in end-hosts or network infrastructure. We performed rigorous theoretical analysis and experimentation to evaluate the effectiveness and overhead of RHM.

We evaluated RHM through implementation, experimentation and simulation. Our experimentation shows that RHM can defeat scanning tools by invalidating at least 97% of its discovery. We also show that RHM can defeat random scanning worms by decreasing the number of infected hosts by 40 to 80% and by slowing down the propagation speed by 50%.

Our implementation and simulation also shows that the routing update overhead is tens of times smaller than random mutation without optimization and the average packet translation and lookup overhead is less than one tenth of a millisecond.

In the future, we plan to study reliability and security issues with RHM operation. For example, we would like to study impact of failures and attacks on RHM devices. We also plan to investigate other related moving target techniques such as random route mutation and deceptive fingerprinting.

References

1. Lyon, G.F.: Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure, USA (2009)
2. Laudicina, A.P.: Nessus - a powerful, free remote security scanner. Sys. Admin. 11(5) (2002)
3. Whyte, D., Kranakis, E., van Oorschot, P.C.: DNS-based Detection of Scanning Worms in an Enterprise Network. In: Proceedings of The 12th Annual Network and Distributed System Security Symposium (February 2005)
4. Bjørner, N., de Moura, L.: $z3^{10}$: Applications, enablers, challenges and directions. In: CFV 2009 (2009)
5. Antonatos, S., Akritidis, P., Markatos, E.P., Anagnostakis, K.G.: Defending against hitlist worms using network address space randomization. Comput. Netw. 51(12), 3471–3490 (2007)
6. Kewley, D., Fink, R., Lowry, J., Dean, M.: Dynamic approaches to thwart adversary intelligence gathering. In: DARPA Information Survivability Conference and Exposition, vol. 1, p. 0176 (2001)
7. Michalski, J.T.: Network security mechanisms utilising network address translation. International Journal of Critical Infrastructures 2(1), 10–49 (2006)
8. Atighetchi, M., Pal, P., Webber, F., Jones, C.: Adaptive use of network-centric mechanisms in cyber-defense. In: ISORC 2003, p. 183. IEEE Computer Society (2003)
9. Kewley, D., Fink, R., Lowry, J., Dean, M.: Dynamic approaches to thwart adversary intelligence gathering. In: Proceedings of DARPA Information Survivability Conference Exposition II, DISCEX 2001, vol. 1, pp. 176–185 (2001)
10. Jafarian, J.H., Al-Shaer, E., Duan, Q.: Openflow random host mutation: Transparent moving target defense using software defined networking. In: Proceedings of HotSDN workshop at SIGCOMM 2012, Helsinki, Finland (2012)
11. Yegneswaran, V., Alfeld, C.: Camouflaging honeynets. In: Proceedings of IEEE Global Internet Symposium (2007)
12. Cai, J.-Y., Yegneswaran, V., Alfeld, C., Barford, P.: An Attacker-Defender Game for Honeynets. In: Ngo, H.Q. (ed.) COCOON 2009. LNCS, vol. 5609, pp. 7–16. Springer, Heidelberg (2009)
13. Chakravarty, S.: A characterization of binary decision diagrams. IEEE Trans. Comput. 42(2), 129–137 (1993)
14. Al-Shaer, E.S., Hamed, H.H.: Discovery of policy anomalies in distributed firewalls. In: Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, vol. 4, pp. 2605–2616 (March 2004)
15. Zou, C.C., Towsley, D., Gong, W.: On the performance of internet worm scanning strategies. Elsevier Journal of Performance Evaluation 63, 700–723 (2003)
16. Zou, C.C., Towsley, D.: Routing Worm: A Fast, Selective attack worm based on IP address information. In: Workshop on Principles of Advanced and Distributed Simulation (PADS 2005), pp. 199–206 (2005)