OpenFlow Random Host Mutation: Transparent Moving Target Defense using Software Defined Networking

Jafar Haadi Jafarian, Ehab Al-Shaer, Qi Duan
Department of Software and Information Systems
University of North Carolina at Charlotte
Charlotte, NC, USA
{jjafaria, ealshaer, qduan}@uncc.edu

ABSTRACT

Static configurations serve great advantage for adversaries in discovering network targets and launching attacks. Identifying active IP addresses in a target domain is a precursory step for many attacks. Frequently changing hosts' IP addresses is a novel proactive moving target defense (MTD) that hides network assets from external/internal scanners. In this paper, we use OpenFlow to develop a MTD architecture that transparently mutates IP addresses with high unpredictability and rate, while maintaining configuration integrity and minimizing operation overhead. The presented technique is called OpenFlow Random Host Mutation (OF-RHM) in which the OpenFlow controller frequently assigns each host a random *virtual* IP that is translated to/from the *real* IP of the host. The real IP remains untouched, so IP mutation is completely transparent to end-hosts. Named hosts are reachable via the virtual IP addresses acquired via DNS, but real IP addresses can be only reached by authorized entities. Our implementation and evaluation show that OF-RHM can effectively defend against stealthy scanning, worm propagation, and other scanning-based attack.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations

Keywords

IP mutation, software defined networking (SDN), Moving target defense (MTD), Security

1. INTRODUCTION

Static assignment of IP addresses gives adversaries significant advantage to remotely scan networks and identify their targets accurately and quickly. Scanning tools and worms usually send probes to random IP addresses in the network in order to discover their targets. When a target responds, it can then be identified and attacked. Otherwise the probed addresses will be considered unused. Despite firewall deployment, most enterprise networks have many public and private hosts accessible from outside. The IP address assignment scheme can become more dynamic by using approaches based on DHCP or NAT, but they are insufficient to provide proactive countermeasure because the IP mutation is infrequent and traceable.

In this paper, we introduce a moving target technique called *OpenFlow Random Host Mutation* (OF-RHM) which mutates IP addresses of end-hosts randomly and frequently so that the attackers' premises about the static IP assignment of network fails. OF-RHM has two main objectives. Firstly, the IP mutation must be transparent to the end-host. To provide transparency, OF-RHM keeps the actual or real IP addresses of hosts (called *rIP*) unchanged, but associates each host with random short-lived virtual IP addresses (called *vIP*) at regular intervals which are translated to rIPs right before the host. Secondly, the IP mutation must be performed with high unpredictability and speed to maximize the distortion of attackers' knowledge about the network and increase deterrence of attack planning. To optimize IP mutation with respect to unpredictability and speed, the mutant vIPs are selected randomly from the entire unused address space in the network. The unused address ranges must be assigned to hosts such that it satisfies several constraints including mutation unpredictability and minimum required mutation rate of all hosts. We formulate this problem as a constraint satisfaction problem and solve it using Satisfiability Modulo Theories [1] (SMT) solvers.

Implementation of this technique requires two major components: (1) subnet gateways to perform rIP-vIP translation, and (2) a central management authority which coordinates mutation across network. In a traditional network these components must be incorporated in the network architecture. This incorporation could be disruptive and costly. Furthermore, it poses serious network management challenges such as real-time global reconfiguration, and synchronization of several network devices in a decentralized environment.

Software-defined networking (SDN) provides flexible infrastructure for developing and managing random host mutation efficiently and with minimal operational overhead. In SDN, the network controller (*e.g.*, NOX [2]) monitors and controls the entire network from a *central* vantage point via an interface, such as OpenFlow [3] and defines the forwarding and address translation behavior of switches distributed in the network accurately and synchronously.

In OF-RHM, the controller preforms the following tasks:

HotSDN 12, August 13, 2012, Helsinki, Finland.

(1) coordinates mutation across OpenFlow switches based on host mutation requirements and available unused address space, (2) determines optimal set of new vIPs for hosts using SMT [1], (3) manages active connections by installing flows in OF-switches along with required address translations actions, and (4) handles DNS updates. Each OpenFlow switch (OF-switch) performs the vIP-rIP translations as specified by the controller.

We implemented the OF-RHM approach on OpenFlow managed by a NOX controller. To facilitate the development and analysis of our approach, we used Mininet [4] to generate fairly large networks of OpenFlow switches and hosts. Our theoretical analysis and implementation results show that OF-RHM can reduce the accuracy of information gathering via scanning up to 99%. Moreover, up to 90% of the network hosts are saved from vicious scanning worms. One limitation of OF-RHM is that a named host can still be reached via DNS. However, most existing scanners use IP address to collect information in order to avoid too many queries to DNS and thus detection.

The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 defines and formulates the IP mutation problem. In Section 4 we describe the architecture and protocol details of OF-RHM. Section 5 describes the implementation of OF-RHM using OpenFlow and its evaluation against several attack models.

2. RELATED WORK

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 [5] 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. DyNAT [6] provides a transparent approach 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 [7] 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).

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

3. PROBLEM DEFINITION AND FORMULATION

In OF-RHM, each host is associated with an unused address range of the network based on its specific requirement. At each mutation, a vIP is chosen from this range and associated with the host. The vIP of each host is mutated after each *mutation interval*.

The main objective of OF-RHM is to maximize both mutation unpredictability and mutation rate. The proposed technique must address both IPv4 and IPv6 address schemes. Scarcity of IP addresses in IPv4 networks makes the unused address space small and highly fragmented. Therefore, major challenge of OF-RHM is to guarantee that, even with a limited and fragmented unused address space, each host would mutate with its required rate such that no IP address is reused (assigned to any host more than once) for a reasonably long time.

These objectives can be achieved by choosing each vIP from largest possible unused address space such that the same vIP is not assigned more than once to any host in many consecutive vIP mutations. This problem can be divided into two sub-problems: (1) allocating unused ranges to hosts, and (2) mutation within allocated ranges.

3.1 Range Allocation

Suppose we have a set of n hosts $\{h_1, \ldots, h_n\}$. Minimum required mutation rate (R_i) for each host h_i is provided as input. In general, sensitive hosts are supposed to have higher mutation rates. Each host belongs to a subnet in the set $\{s_1, ..., s_z\}$, where subnet is a group of hosts that are physically connected through an OF-switch.

For mutation, we need to determine the unused address ranges in the network address space. Given used address ranges $A_1, \ldots, A_u$, we determine the contiguous blocks of unused address ranges of the network by simply masking the full network address space A as follows using Boolean operations:

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

If a range is larger than a maximum size, it is divided into smaller ranges.

Sharing ranges among hosts allows us to increase mutation unpredictability and rate, because the host can mutate in a larger range. However, routing limitation does not allow us to share all unused ranges between all hosts, because each range can only be routed to one subnet. Based on these considerations, the OF-RHM problem is:
Given unused ranges $r_1, ..., r_m$ and subnets $s_1, ..., s_z$, what is the appropriate assignment scheme such that the following objectives are achieved:

- *Objective I*: the ranges assigned to the subnet must include enough IP addresses to satisfy the minimum required mutation rate of all hosts in that subnet during an interval, T, such that no IP addresses is assigned twice in one interval.

- *Objective II*: unpredictability and mutation rates must be maximized by firstly allocating all unused address ranges, and secondly assigning ranges proportionate to the mutation requirement of each subnet.

The problem of assigning ranges to subnets is NP-hard because a subnet may be assigned with multiple ranges due to different mutation requirements and unequal range sizes. This is a generalization of the NP-hard knapsack problem (knapsack problem with multiple bags) [8]. Since the problem is NP-hard, we formulate this problem using the following SMT (Satisfiability Modulo Theories [1]) formulas. Boolean *variable* $b_{jk} \in [0,1]$ denotes whether range r_j is assigned to subnet s_k. Boolean *value* c_{ik} shows if host h_i belongs to subnet s_k. The unpredictability interval, T, denotes the interval during which a vIP must not be assigned to any host more than once.

Mutation Rate Constraint: *Objective I* denotes that the total number of mutated vIPs of all hosts in subnet s_k during T must be less than the aggregate size of all ranges assigned to s_k (Eq. 2). The required vIPs of a host h_i during one repetition cycle is $T * R_i$.

$$\forall k, \left(\sum_{1 \le i \le n} c_{ik} R_i \right) * T \le \sum_{1 \le j \le m} b_{jk} |r_j| \qquad (2)$$

Range Allocation Constraint: Each range must be assigned to exactly one subnet (Eq. 3), because *objective II* entails that each range must be allocated to *at least* one subnet, while routing constrains us to assign each range to *at most* one subnet.

$$\forall j, \sum_{1 \le k \le z} b_{jk} = 1 \qquad (3)$$

Range Distribution (Unpredictability) Constraint: Based on *objective II*, ranges must be assigned to subnets proportionate to their total required mutation rate. To this aim, we define a variable P_k as the total required mutation of subnet s_k during T on total size of ranges allocated to it.

$$\forall k, P_k = \frac{T * \sum_{1 \le i \le n} c_{ik} R_i}{\sum_{1 \le j \le m} b_{jk} |r_j|}$$

Also, we define P_a as total required mutation of all hosts on total size of unused address ranges:

$$P_a = \frac{T * \sum_{1 \le i \le n} R_i}{\sum_{1 \le j \le m} |r_j|}$$

It is easy to see that the unpredictability constraint can be denoted as:

$$\forall k, P_k \simeq P_a \qquad (4)$$

The constraint means that the total size of assigned address spaces for a subnet should be proportional to the requirements of the subnet. To solve the constraint with SMT, we rewrite it as:

$$\forall k, |P_k - P_a| < \delta \qquad (5)$$

where δ is a constant value.

We use Yices [9] SMT solver to find values for b_{jk} variables according to constraints of Eq. 2, 3 and 5. Initially we set $T = \frac{\sum_{1 \le j \le m} |r_j|}{\sum_{1 \le i \le n} R_i}$ and δ to a default value and assert the constraints. If no solution is found, we decrement T and relax δ to a larger value, and assert the constraints again. We repeat these steps until a solution is found.

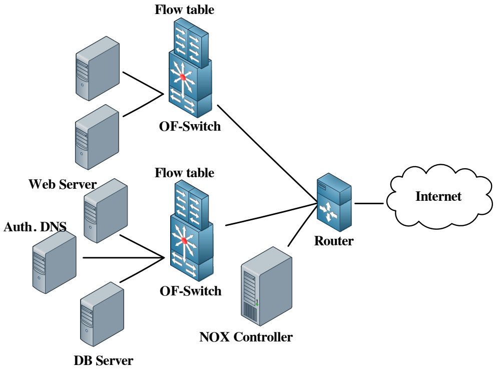

Figure 1: The architecture of OF-RHM network

The optimal mutation rate of h_i is called R'_i and determined based on the total size of allocated ranges to its subnet, s_k:

$$\forall i, R'_i = \frac{\sum_{1 \le j \le m} b_{jk} |r_j|}{T} \qquad (6)$$

3.2 vIP Mutation

Each host must be associated with a new vIP after each mutation interval (that is, $1/R'_i$). Assume some ranges are assigned to a subnet. The problem is to assign vIPs to all hosts in the subnet, such that:

- No collision occurs in vIP assignment; *i.e.*, a vIP must not be assigned to two or more hosts simultaneously.
- A vIP must not be assigned to a specific host more than once in a T interval. This is ensured by keeping track of mutations during last T seconds.

The new vIP can be chosen in two ways:

- *Blind mutation*: the vIPs are chosen from the designated address space with uniform probability.
- *Weighted mutation*: a weight is associated with each vIP based on a certain criteria. The vIPs are chosen randomly from the address space such that the selection probability is directly related to the weights of vIPs.

In Section 5 we describe our weighting function.

4. OF-RHM: ARCHITECTURE AND PROTOCOL

In this section, we discuss the architecture of OF-RHM, and its communication protocols.

4.1 Architecture

We implemented OF-RHM in a large Mininet network controlled by a NOX controller. Figure 1 shows the architecture of the OF-RHM network. NOX controller acts as the central authority managing IP mutation, flow installation in switches, and DNS responses. For scalability, this architecture is extended to include several controllers, each managing a segment of the network. Each controller has full autonomy in management of its designated network segment, because no information need to be shared among controllers. Range assignment is performed only once in the initialization stage, and the routers and controllers are updated with routing and access control policies. After initialization, no information is exchanged among controllers. This allows OF-RHM to be scalable to any network size.

The general algorithm of NOX controller is presented in Algorithm 1. OF-switches are configured to encapsulate unmatched packets (that have no matching flows in flow tables) and send them to the controller. The controller determines the type of connection (*i.e.*, via rIP or vIP) and installs necessary flows in all OF-switches in the path. Each connection must be associated with a unique flow, because the rIP-vIP translation changes for each connection. This property guarantees the end-to-end reachability of hosts, because the rIP-vIP translation for a specific connection remains unchanged regardless of subsequent mutations.

Algorithm 1 NOX controller algorithm

```
determine unused ranges.
determine range-to-subnet assignments
for all packets $p$ from OF-Switches do
    if $p$ is a Type-A DNS response for host $h_i$ then
        set DNS addr to current $vIP(h_i)$, $TTL \simeq 0$
    else if $p$ is a TCP-SYN or UDP from $h_i$ to $h_j$ then
        if $p.src$ is internal then
            install in flow in src OF-switch with
                action $srcIP(p) := vIP(h_i)$
            install out flow in src OF-switch with
                action $dstIP(p) := rIP(h_i)$
        end if
        if $p.dst$ is rIP then
            if $h_i$ access to $h_j$ is authorized then
                install in and out flows in dest OF-switch
            end if
        else[$p.dst$ is vIP]
            install in flow in dest OF-switch with
                action $dstIP(p) := rIP(h_j)$
            install out flow in dest OF-switch with
                action $srcIP(p) := vIP(h_j)$
        end if
    end if
    for all mutation of each host $h_i$ do
        set $vIP(h_i)$ to a new vIP
    end for
end for
```

4.2 Protocol

There are two ways to communicate with hosts: using host name or host rIP. These two scenarios are depicted in Figures 2 and 3, respectively. As represented in Figure 2, when a DNS query is sent to resolve the name of a host, the DNS response is updated by the NOX controller to replace the rIP of the server with its active vIP (steps 1-3). The NOX controller also sets the TTL value in the DNS response to a small value. The source host can then initiate the connection using the vIP of the destination. The OF-switch encapsulates and sends the initial packet to the controller (step 4), because there exists no matching flow for it. The NOX controller installs relevant flows in OF-switches in the route (step 5). These flows are associated with relevant required *Set-Field* actions which determine the translation of source/destination rIP addresses to/from vIP addresses. Relevant flows are installed in destination OF-switch as well. Any other switches in the path are only configured (*i.e.*, flows with no actions) to route this traffic based on vIP. Future packets will be matched and forwarded by OF-switches (without being sent to controller) according to the installed flows in the flow table. The vIP-rIP translation actions will be applied to packets by OF-switches.

Figure 3 shows how authorized users (*e.g.*, administrators) can reach hosts using rIPs. In this case, the source host initiates a connection with the destination using its rIP. Similar to DNS scenario, the OF-switch will fail to match the new packet with any flow and sends it to NOX controller (step 1). The NOX controller authorizes the access request (step 2). If granted, the controller installs appropriate flows in OF-switches in the route (step 3) with appropriate vIP-rIP translation actions according to Figure 3. As represented in Algorithm 1, if the source host is internal, NOX controller installs two inbound and outbound flows in source OF-switch, same to the DNS scenario. Two flows are also inserted in the flow table of destination OF-switch. However, destination flows require no translation, because the destination host is being reached via its rIP. The rest of the packets will be translated and forwarded according to these flows. Access control policies for authorizing rIP connections are defined based on the criticality of the host.

5. IMPLEMENTATION AND EVALUATION

We used Mininet to create a network of OpenFlow switches (Open vSwitch kernel switches) that were connected according to Figure 1. Each subnet had a separate network address, and the routing was handled by NOX controller. The network addresses were chosen as subnetworks of a class B network. To simulate external hosts, we designated one of these subnets as external subnet (with an external network address). We used methods presented in [10] to ensure lack of misconfigurations as a result of repeated mutations.

For weighted mutation, the NOX controller counts the number of times each vIP has been scanned in a sliding time window. Specifically, each time an unused IP is scanned the controller increments its weight by one. In selecting a new vIP for a host, the NOX controller randomly chooses a vIP with a selection probability related to the weights. This method is effective, because an already scanned IP is less likely to be scanned again by an external scanner or worm. While this approach is easily implemented on OpenFlow, its implementation on traditional network is very challenging.

Although comprehensive evaluation of random host mutation is not the focus of this paper we show the viability and basic effectiveness of OF-RHM against external scanners and worms. The detailed and rigorous evaluation of random host IP mutation on external scanners, random scanning worms, and denial of service attacks is presented in [11].

5.1 Random External Scanners

Scanning is usually the precursory step for attacks. The attackers usually use scanning tools such as Nmap to discover active hosts in the target network and use it as a hitlist. The active targets can also be discovered by tracing audit logs, caches, sniffing, etc. OF-RHM can prevent hitlist-based attacks (*e.g.*, hitlist worms) effectively, since the IP addresses in the hitlist will be soon out-of-date. To show the effectiveness of OF-RHM against hitlist attacks, we run 100 *Nmap* scan on our Mininet class B network which consists of 2^{10} hosts. By comparing all scan reports with the ground truth (achieved via an initial scan) we found not

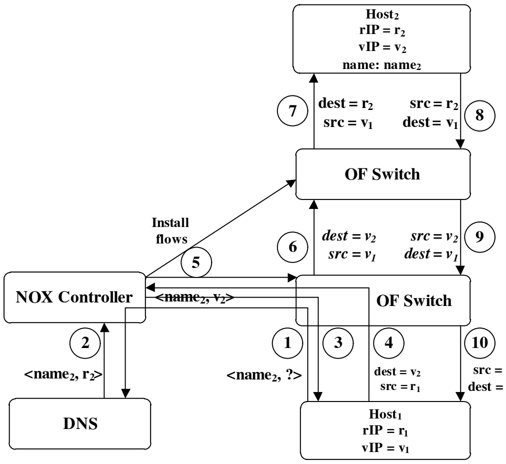

Figure 2: Communication via name

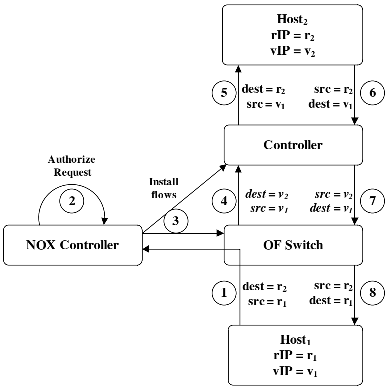

Figure 3: Communication via rIP address

more than 1% of initial vIP addresses are discovered in any scan, as shown in Figure 4. The attacker can use reverse-DNS lookup to discover the name of scanned targets and use it in future attacks. However, rDNS is mainly used for spamming and we disable it for all services except for mail servers.

5.2 Worms

The scanning strategy adopted by the worm for target discovery determines its effectiveness in terms of propagation. The effectiveness of a scanning strategy is determined by decreasing the probability of multiple scanning of a specific IP. The most effective scanning strategy is known as cooperative scanning (*e.g.*, divide-and-conquer worms [12]) in which all infected hosts cooperate with each other such that no IP address would be scanned more than once [12]. Random scanning worms usually use sophisticated random number generators to reduce the possibility of multiple scans of the same IP [12].

Figure 5 shows the effectiveness of OF-RHM with blind mutation, and OF-RHM with weighted mutation on propagation of cooperative and random scanning worm in a class B network with 2^{10} hosts. The scanning worm is assumed to have a low IP repetition probability. The mutation rate of each host is 0.2 (one mutation each 5 seconds). Our results show that OF-RHM is significantly effective against scanning worms and it can save up to 90% of hosts against the most sophisticated worms.

5.3 Overhead

Address Space: The number of vIPs assigned to host h_i must be at least $T * R_i$ to satisfy its mutation rate constraint. So, the total size of required mutation space is $(\sum_{1\leq i\leq n} R_i)T$. Figure 6 shows the address space required for various mutation intervals $(1/R_i)$, and various number of hosts. For simplicity, we assume that mutation rate of all hosts are equal.

Flow Table Size: In traditional networks, the length of flow table in OF-switches is an order of $O(n)$, because flows are specified and matched only by destination address (regardless of source address, ports, or other matching fields). In OF-RHM, two flows must be specified for each TCP/UDP session between each two hosts. According to Little's law, if a network has n hosts where each host on average establishes λ sessions per second, and each session on average takes w seconds to terminate, the mean length of flow table is λwn. Figure 7 shows flow table length for various mutation establishment rates (λ) and session durations (w).

6. CONCLUSION

In this paper, we described OF-RHM as a moving target defense approach using software-defined networking. The basic goal of OF-RHM is to thwart scanning via random and unpredictable mutation of host IP addresses. We formalized the problem of unpredictable and rapid mutation as determining a valid assignment of unused address ranges under multi-constraint satisfaction. We also described the architecture and communication protocols of OF-RHM on OpenFlow. Our implementation represented the feasibility and effectiveness of this technique against scanning. We showed that OF-RHM can invalidate the information gathering of external scanners up to 99%. It can also save up to 90% of network hosts from even zero-day unknown worms. We also scrutinized the overhead of OF-RHM in terms of number of required IP addresses for achievement of certain mutation rates and unpredictability as well as the average length of flow table.

For future, we plan to study OF-RHM effect against other attack models such as distributed denial of service (DDoS) and application-layer attacks.

7. REFERENCES

[1] N. Bjørner and L. de Moura. $z3^{10}$: Applications, enablers, challenges and directions. In *CFV '09*, 2009.

[2] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. Nox: towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, 38(3):105–110, July 2008.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38.

[4] B. Lantz, B. Heller, and N. McKeown. A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the Ninth ACM*

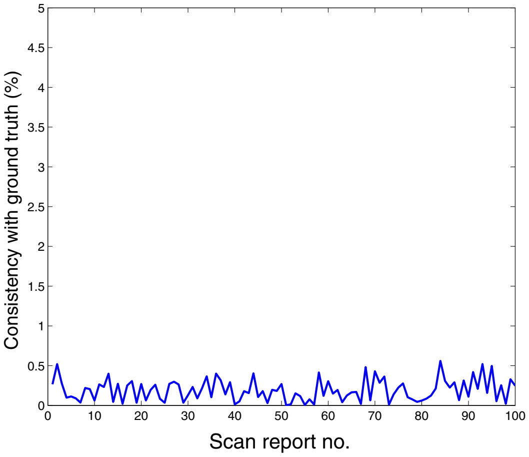

Figure 4: Consistency of consecutive Nmap scan reports with ground truth

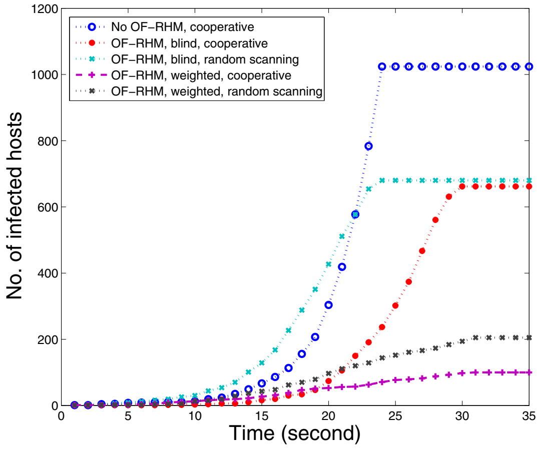

Figure 5: Worm propagation for various network setups

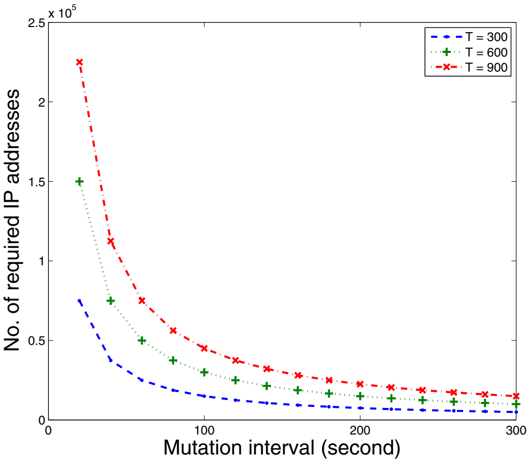

Figure 6: Required IP address size for various mutation intervals and number of hosts

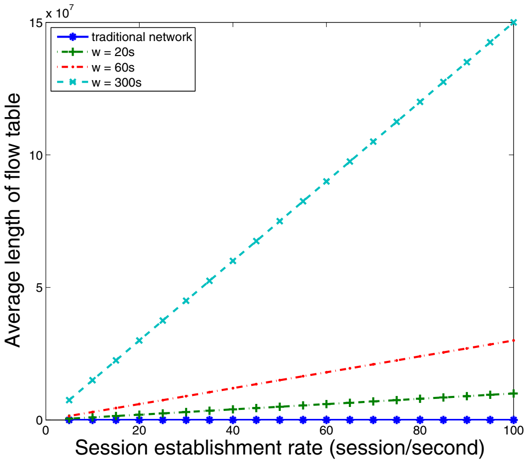

Figure 7: Flow table length for different session establishment rates and session durations

SIGCOMM Workshop on Hot Topics in Networks, Hotnets '10, pages 19:1–19:6, New York, NY, USA, 2010. ACM.

[5] M. Atighetchi, P. Pal, F. Webber, and C. Jones. Adaptive use of network-centric mechanisms in cyber-defense. In *ISORC '03*, page 183. IEEE Computer Society, 2003.

[6] D. Kewley, R. Fink, J. Lowry, and M. Dean. Dynamic approaches to thwart adversary intelligence gathering. In *DARPA Information Survivability Conference Exposition II, 2001. DISCEX '01. Proceedings*, volume 1, pages 176 –185 vol.1, 2001.

[7] S. Antonatos, P. Akritidis, E. P. Markatos, and K. G. Anagnostakis. Defending against hitlist worms using network address space randomization. *Comput. Netw.*, 51(12):3471–3490, 2007.

[8] Ellis Horowitz and Sartaj Sahni. Computing partitions with applications to the knapsack problem. *J. ACM*, 21(2):277–292, april 1974.

[9] SRI International. *Yices: An SMT Solver*, 2012. `http://yices.csl.sri.com/`.

[10] E. Al-Shaer, W. Marrero, A. El-Atawy, and K. ElBadawi. Network configuration in a box: towards end-to-end verification of network reachability and security. In *Network Protocols, 2009. ICNP 2009. 17th IEEE International Conference on*, pages 123 –132, 2009.

[11] Ehab Al-Shaer and Qi Duan. Random host IP mutation for moving target defense. Technical Report UNCC-CYBERDNA-0728, CyberDNA Lab, University of North Carolina at Charlotte, Charlotte, NC, July 2011.

[12] C. C. Zou, D. Towsley, and W. Gong. On the performance of internet worm scanning strategies. *Elsevier Journal of Performance Evaluation*, 63:700–723, 2003.