

An Introduction to Network Address Shuffling

Guilin Cai, Baosheng Wang, Xiaofeng Wang, Yulei Yuan, Sudan Li

College of Computer, National University of Defense Technology, Changsha, China

cc_cai@163.com, wangbaosheng@126.com, xf_wang@nudt.edu.cn, 377yuanyulei@163.com, nudtisd@163.com

Abstract— Moving Target Defense (MTD) has been proposed as a new revolutionary technology to alter the asymmetric situation between attacks and defences. Network address shuffling is an important branch of MTD technology. However, there is no systematic introduction to network address shuffling. In this paper, we present a brief introduction to the research achievements of network address shuffling according to two shuffling patterns which are identified and defined by us. We then summarize and analyze the supporting techniques and related features for each network address shuffling technique mentioned in this paper. What's more, the key issues to implement an effective network address shuffling mechanism are discussed, with the expectation of invigorating subsequent research.

Keywords— Moving Target Defense, Network Address Shuffling, introduction, analysis, discussion

I. INTRODUCTION

With the rapid growth of information technologies, the Internet has become a key national infrastructure. What's more, the Internet is profoundly changing our work habits and daily life, which influences the normal running of society. Therefore, all countries pay close attention to the security of the Internet. However, cyber-attacks (such as IP prefix hijacking [1], botnet [2], DDoS attack [3]) can be seen everywhere and at any time, and major security incidents have been frequently reported in recent years (such as the PRISM [4], the Heartbleed Bug [5], eBay data leakage). Such security disasters are repeatedly showing that the security of the Internet is always facing severe challenges.

One of the major reasons of the severe security situation is that the network configurations are typically deterministic, static, and homogeneous nowadays [6], [7]. These features reduce the difficulties for cyber attackers scanning the network to identify specific targets and gather essential information, which gives the attackers the advantages of building up, launching and spreading attacks. First, the attackers have the advantage of time, since they can perform vulnerability analysis and penetration testing for specific target repeatedly before they achieve the final goal. Second, the attackers have asymmetric advantage in terms of acquiring the information needed for initiating and launching an attack and the attackers can attack as long as there is a usable vulnerability; while the defenders have to secure all potential vulnerabilities and prevent all the attacking means that can be utilized by the attackers. Third, the attackers have the advantage of cost to expand the attack, since the homogeneity in network configurations enables the attackers to carry out

large-scale attack easily and at low cost once a small scale attack succeeds. Therefore, in the struggle between cyber network attack and defense, the attackers typically have asymmetric advantages and the defenders are always disadvantaged by being passive.

To alter the asymmetric situation between attacks and defenses, Moving Target Defense (MTD) is proposed as one of the “game-changing” themes in cyber security [6], [8]. Its vision is described as follows: create, evaluate, and deploy mechanisms and strategies which are diverse, continually shift and change over time to increase complexity and costs for attackers, limit the exposure of vulnerabilities and opportunities for attack, and increase system resiliency [6]. What's more, one way to achieve the goal is automatically changing one or more system attributes in a manner continually [8]. IP address is an important system attribute. It is the foundation of Internet communication, and is also the premise of the attacker to exploit and analyze vulnerabilities, and finally launch an attack. Port number is also an important attribute of communication and a condition for some certain types of attacks. For this reason, changing IP address and/or port number is an effective way to increase the work effort for attacking. That is the origin of network address shuffling.

Network address shuffling [9] is a dynamic reconnaissance defense that periodically permutes the relationships between addresses and devices. For the Internet, addresses are a combination of IP and transport layer information (protocol and port numbers), and either or both types of which can be used for shuffling. There have been many mechanisms in this category, and they have some common properties to ensure the normal running and defense effectiveness. In this work, we identify and discuss them, and the main contributions are as follows:

- 1) Identification and definition of the two patterns for shuffling.
- 2) A brief review and analysis to the main mechanisms in network address shuffling.
- 3) Discussions on the key issues for creating an effective mechanism in network address shuffling.

II. REVIEW TO NETWORK ADDRESS SHUFFLING

Network address shuffling technique aims to change the IP address (and port number) of target periodically or erratically. From the existing research on network address shuffling, we find that there are two patterns called hopping and mutation for the changing.

In the hopping pattern, the synchronization of communication is strict in time, i.e., the two sides are fully aware of the hopping pattern information (including the hopping sequence and hopping timelessness) of both, or one side (e.g. the clients) is fully aware of the hopping pattern information of the other side (e.g. the server). The synchronization is usually achieved by time synchronization scheme, or exchanging the hopping pattern information, or pre-setting the same function and initial value.

In the mutation pattern, the synchronization of communication is not strict in time, i.e., one side of communication (e.g. the clients) do not need to know the shuffling information of the other side of communication (e.g. the server). The synchronization is usually achieved by routing update and DNS request/respond, or the other supporting third-party mechanism.

Next we will take a brief introduction to network address shuffling techniques according to the two patterns.

A. Hopping

Early in 2001, DYNAT [10] is proposed to defend any adversary sniffing the network. Before the packet was routed, the host identity information that can uniquely describe a connection between two hosts, host portion of destination's IP address, and TCP/UDP port number, were translated by a cryptographic algorithm. Then the datagram was sent to the public network. When the stand-alone DYNAT gateway receives the datagram, it reverses the translation firstly to acquire the true host identity information, and then passes it to the receiver. In DYNAT, the encryption algorithm of the sender and the receiver is configured with an initial secret seed value. The secret seed value is periodically changed by using a time-based mechanism, and the secret-switching mechanisms between the sender and receiver are synchronized by wall-clock time. DYNAT can effectively increase the difficulty of collecting essential information to map the network and build up attacks, and the packets can be routed normally without increasing additional routing overhead. However, it is not transparent to the clients, for the DYNAT shim which is used to do the translation of host identity must deploy on the client-side.

APOD [11] attempts to integrate port and address hopping with a set of underlying network-based mechanisms (for example, intrusion detection, firewalls, and VPNs), to increase an application's resilience against cyber attacks. The implementation of the hopping in client-side is performed by a hopping delegate, which intercepts IIOP RPC calls to the real server and replaces all (realaddress: realport) header information with (fakeaddress: fakeport). A NAT gateway that locates on the server's LAN or directly on the server host is needed to reverse the replacement. There are two kinds of synchronization in the mechanism: the initial value of the random number generators, and the time of switchover using a newly generated (fakeaddress: fakeport) pair. What's more, to ensure the effective of the approach, the synchronization must be strict in time. The APOD toolkit can adaptively use some individual network-based defenses, and the complexity

changes with the selection of the network-level defenses. However, it is not transparent to the clients.

Network Address Hopping [12] is proposed to enhance the security of data by transferring them across multiple data connections named channels. Channel is defined by either the destination IP address or the <source end, destination end> tuple and each end is expressed by either the IP address, or IP address and port number, or an application specific identifier. The hopping pattern, in which include the hopping sequence (i.e., the order and the amount of time each channel is occupied), is defined by each endpoint respectively and sent to the other endpoint, thus the two sides are fully aware of the hopping pattern information. The mechanism changes the communication mode of two endpoints, and can confuse and disturb the attackers effectively due to the hopping. It is fully transparent to the end-user applications but not to the end-user. Furthermore, it needs to equip one side or both sides with multiple addresses.

Service Hopping [13] and End Hopping [14] are proposed by L. Shi et al.. Actually, the two concepts are with the same meaning, which are treated as the dynamic tactics that change the end information of service port, network address, service timeslot, cryptographic algorithm, or even protocol pseudo-randomly during end-to-end transmission. Their main work focused on designing a port and address hopping mechanism and a novel timestamp-based synchronization scheme to evade the DoS/DDoS attack. When a hopping service (actually, the hopping service is a service of which the (address, port) pair is changed over time) is to be launched, the hopping algorithm shared between the server and trusted hosts would generate candidate (address, port) pairs with the server's current timestamp, and the timestamp is transferred to the trusted clients in either public channel or covert channel for synchronization. In other words, the clients are fully aware of the hopping pattern information of the server. The mechanism can effectively decrease the success rate of DoS/DDoS attack. However, it is not transparent to the clients, and the complexity and cost of deploying the mechanism is adaptive with the shuffling parameters (in addition to the IP address and port number, there are protocol, timeslot and cryptographic algorithm).

MT6D [15] aims to prevent attackers from targeting specific address through dynamically rotating the source and destination network- and transport-layer addresses for both communicating hosts. At each time increment, MT6D computes the next IIDs (Interface IDentifier in an IPv6 address, which can identify a particular node) for both sender and receiver of each communicating pair, by using a hash function and three parameters: the current IID, a shared symmetric key, and the system timestamp. In this way, the two sides of the communication can compute its own and the other's next IID, i.e., they are fully aware of the hopping pattern information of both. MT6D is transparent to the user, and it makes no side effect on the normal communication because the obscuration of address can be made in the middle of ongoing sessions without breaking the connection or requiring a new handshake [16]. It can increase the work

efforts to attack because the address space of IPv6 is large and the IIDs of a session keep changing. However, MT6D devices must maintain multiple IPv6 address for each node at any given time, which may increase storage resource and cause additional cost for routing update.

MORPHINATOR [17] aims to build a prototype network capable of morphing over time to confuse and thwart potential attackers on military networks. We know that the prototype focuses on the address hopping and port hopping techniques to constantly change the characteristics of the networks it resides on. However, there is no more detail about it.

B. Mutation

NASR [18] is proposed to against hitlist worms. A basic form of NASR can be achieved by using three timers to configure a DHCP server to expire DHCP leases at various intervals for controlling host addresses. In this approach, the two communicating hosts are not aware of the other's next address until sending a DNS request and getting a reply. The implementation is transparent to user. However, it should be deployed in the network segment that already performs dynamic address allocation, and adjust the address shuffling frequency to achieve its goal.

SDNA [19] is an architecture that constructively combines hypervisor technology, Common Access Card-based authentication together in a complementary way. In this architecture, the SDNA Entity within each node can rewrite the address of packets entering and exiting the Operating System (OS) to prevent each Guest from knowing the identity of other nodes within the enclave. When a DNS response comes to the guest, the SDNA Entity would replace the real IP with Token IP which is generated by the SDNA Entity. When the guest initiates a connection to a Token IP, the SDNA Entity would rewrite the packets by replacing the Token IP with the real IP. In other words, one side of a communication does not know the other's real address, and the Token IP is obtained from the other's SDNA Entity when it requests a DNS resolution. The SDNA is transparent to the OS and compatible with the existing infrastructure. However, the traffic between the communication endpoints must flow through one or more intermediate nodes to be rewritten for concealing the endpoints' identities, and it requires multiple key exchanges and authentications in the paths' establishment process, thus the complexity and cost of implementation is high.

OF-RHM [20], RHM [21], and Spatio-temporal address mutation [22] are proposed by E. Al-Shaer et al.. OF-RHM [20] is a shuffling mechanism that should be deployed in the SDN network. In OF-RHM, each host is associated with an unused address range (i.e., the set of virtual IPs) that is assigned by the OpenFlow controller based on its specific requirement using SMT (Satisfiability Modulo Theories). A new virtual IP is chosen from the range and assigned to the host after each mutation interval, and the new vIP is selected based on uniform probability, or a selection probability directly related to the weight which is associated with each vIP based on a certain criteria. The OpenFlow switch (OF-switch) can perform the real IP-virtual IP (rIP-vIP) translation, and thus make that the packets transmission in the network

between the source and destination gateway are with vIP which increases the work effort of identifying a particular host. However, the OF-RHM can only be used in SDN network, thus the scalability is low.

To address this question, RHM [21] is proposed. Its design principles and implementation are similar to the OF-RHM. The main differences from OF-RHM are the vIP allocation mechanism and the components for distribution. RHM uses a two-phase mutation approach which consists of LFM (Low Frequency mutation) and HFM (High Frequency mutation) to assign vIP. A LFM interval contains multiple HFM intervals. In each LFM interval, a random network address range denoted as VAR (virtual address range) is selected for each MT (Moving Target) host using SMT(Satisfiability Modulo Theories). Then in each HFM interval, a new random vIP within the VAR assigned during last LFM is chosen for the MT host, and the selection of the new vIP is based on a hash function. In addition, the virtual address allocation is performed by the MTC (Moving Target Controller), and the translation between the real IP and virtual IP is performed by the MTG (Moving Target Gateway).

In order to further improve the security, Spatio-temporal address mutation is proposed [22]. In this approach, each host is associated with a unique set of IP address, which called ephemeral IP addresses (eIPs, which is similar to the vIP in OF-RHM and RHM) to reach other hosts, the selection of new eIP is based on the source (requestor) identity as well as time, and a time-to-live (TTL) value which is determined based on the identity of both end-points is set to indicate the expiration time of the rIP-eIP mapping. Hence a host must use different ephemeral IP (eIP, which is similar to the vIP in OF-RHM and RHM) to communicate with another host at various time intervals. In all the three mechanisms, the new vIP/eIP is provided to users by DNS responses.

Decoy-based MTD is used to reduce the rate that the real node is identified and attacked [23]. In the mechanism, a lot of virtual decoy nodes are introduced in the network, and each of decoys is assigned a valid IP address and equipped with simplified common protocols to be looked like a valid system. Thereafter, the addresses of all the real and the decoy nodes are randomly refreshed and reassigned over time. In this mechanism, the two communicating hosts are not aware of the other's next address until sending a DNS request and getting a reply.

MOTAG [24] is proposed to help online application server against network flooding attacks. The mechanism requires a group of proxy nodes to be deployed around the server node, and allocates an active working proxy for each certification user to forward data traffic between the user and the server. The mechanism can protect the online server strongly since the address of the server is privacy for any user, even each client is only aware of its own working proxy's IP address. When a working proxy is under attack, it would startup a process called client-to-proxy shuffling, i.e. the attacked proxy is replaced by a new proxy, and the associated users are migrated to the alternative proxy too. In other words, the client is passive to accept the address shuffling of its working

proxy, and knows nothing about the address of server. MOTAG can gradually concentrate the attacker to a particular proxy to minimize the impact of an attack. However, it must work together with the attack detection mechanism since there is no shuffling if there is no attack.

The SDN shuffle approach [25] uses synthetic addressing information to replace the real addressing information for defending reconnaissance. Each time when a client requests a DNS resolution of the server, the DNS server would notice SDN controller. Then the SDN controller generates a synthetic IP address and MAC address for the server, and sends them to DNS server to reply the client. Also the SDN controller orders the server to install NAT rules that translate the synthetic IP and MAC address into the real addresses. In this way, each client receives synthetic addresses for the server, which the SDN controller can rotate for each new DNS resolution. This allows the SDN controller to provide a different, moving IP address for each client and the short TTL for the client ensures that the client will re-issue DNS requests for new connections, allowing the server to again change the addresses, and thus effectively defend reconnaissance. However, it needs to deploy in SDN network and modify the operation system of DNS server to cooperate with the SDN controller.

III. A BRIEF COMPARISON

We have briefly presented the major network address shuffling techniques in previous section. To help readers to form a preliminary understanding of existing network address shuffling techniques, we take a brief comparison and summarization on them in this section.

First, we use Table 1 to describe the supporting techniques which are made to ensure each technique to run correctly and strengthen the defense effect for each network address shuffling technique.

Then, Table 2 presents the comparison between the network address shuffling techniques from transparency, complexity, scalability, and cost perspectives, respectively. The four comparison metrics are illustrated as follows:

1) Transparency: for which we focus on the transparency to user, not to application or OS. Transparency means that a real thing seems not to be existing, in other words, users have no perception on it.

2) Complexity: The Complexity indicates the degree of simplicity for deploying a network address shuffling technique and ensuring the normal running of the technique, which has certain correlation with the supporting techniques with each shuffling technique.

3) Cost: The Cost includes the deployment cost (include the cost required by supporting mechanisms installed on clients and the modification cost of current network devices) and the transmission cost (the cost to ensure the normal operation of the shuffling, such as additional routing cost and storage cost).

4) Scalability: Scalability includes two meanings, one is whether the mechanism can be deployed on various network, such as current IPv4 network, IPv6 network, or other type network. And the other is whether the mechanism can work

together with other security mechanisms, such as IDS and Firewall, to enhance the performance against attacks.

TABLE 1. THE BASIS OF THE NETWORK ADDRESS SHUFFLING TECHNIQUES

Mechanism	Supporting Techniques
DYNAT[10]	DYNAT shim+ DYNAT Gateway+ encryption algorithm+ time-based synchronization
APOD[11]	Hopping Delegate+ NAT Gateway+ random number generation+ time synchronization+ IDS+ Firewall+ VPN+ TCP Stack Probes+ Bandwidth Reservation Schemes+ Traffic shaping
Network Address Hopping[12]	Multiple endpoint address on one or both ends+ synchronization
Service Hopping [13]	Hopping algorithm+ timestamp synchronization+ multiple addresses and port numbers for each NIC
End Hopping[14]	Hopping algorithm+ timestamp synchronization+ multiple addresses and port numbers for each NIC
MT6D[15]	IPv6+ hash function+ IID rotation+ IID notification
MORPHINATOR [17]	Hopping mechanism
NASR[18]	DHCP Server modification
SDNA[19]	Hypervisor+ CAC-based Authentication+ IPv6+ metadata-based verification+ honey pot+ customized way for rewriting traffic
OF-RHM[20]	SDN network+ an unused addresses range for each host+ vIP selection mechanism
RHM[21]	an unused addresses range for each MT host+ vIP selection mechanism+ Gateway modification
Spatio-temporal address mutation[22]	eIPs for each MT host+ eIP selection mechanism+ host-IP binding determination+ TTL setting for each host-IP binding+ controller+ Gateway modification
Decoy-based MTD[23]	hypervisor+ virtualization + randomization mechanism+ routing update
MOTAG [24]	Attack detection+ a set of working Proxies+ client-to-proxy shuffling+ filter ring+ Authentication with Pow protection
the SDN shuffle approach[25]	SDN network + DNS server modification

IV. FURTHER DISCUSSION

Now we take a discussion on some key issues for implementing an effective and practical network address shuffling technique.

A. The unpredictability of shuffling

Unpredictability is a key characteristic of the MTD technology, which can make it hard for the attacker to predict the precise information of the target in next period, and thus increase the work effort of attack, decrease the probability of successful attacks and increase the target's resiliency. The usual approach to provide unpredictability is combining diversity with randomization, in which diversity provides a large configuration space and randomization maximizes the use of configuration space [26]. In the network address shuffling technology, there is usually a large configuration

TABLE 2. COMPARISON OF DIFFERENT NETWORK ADDRESS SHUFFLING TECHNIQUES

Mechanism	Transparency	Complexity	Cost	Scalability
DYNAT[10]	No	low	low	high
APOD[11]	No	adaptive	adaptive	high
Network Address Hopping[12]	No	low	low	medium
Service Hopping[13]	No	adaptive	adaptive	high
End Hopping[14]	No	adaptive	adaptive	high
MT6D[15]	No	medium	medium	medium
MORPHINATOR[17]	unknown	unknown	unknown	unknown
NASR[18]	Yes	low	medium	high
SDNA[19]	No	high	high	low
OF-RHM[20]	Yes	low	medium	low
RHM[21]	Yes	medium	high	high
Spatio-temporal address mutation[22]	Yes	high	high	high
Decoy-based MTD[23]	Yes	low	medium	high
MOTAG [24]	Yes	high	medium	low
the SDN shuffle approach[25]	Yes	low	medium	low

space (a given address range) to be used, thus the critical problem is the way of randomization to achieve unpredictability.

B. The determination of the shuffling frequency

The combat between the attacks and defenses likes an arms race, in which each side hopes to be one step ahead than the other to achieve their own goals. For the defender, the speediness is influenced by the shuffling frequency. If the shuffling frequency is too low, while the attacker is fast enough, there is most likely a successful attack, i.e., it can't effectively reduce the success rate of attacks. On the other hand, if the shuffling frequency is too high, although it can provide a high degree of security, it would reduce the system performance and availability of the services. Therefore, determining the optimal shuffling frequency is critical.

C. The security of the rules of shuffling

The shuffling has to be controlled and managed by the administrator to provide active defense while ensuring the continuity of mission and the functionality of system, hence the shuffling is actually in certain regularity. In the mutation pattern, the address of next period is often obtained from the DNS resolution, thus the security of the DNS resolution information would influence the security of the mechanism. In the hopping pattern, clients should get the hopping information from the server, or they would exchange the hopping information, thus the security of the hopping information transmission would affect the security provided by the mechanism greatly.

D. Practicability

Being practical is the basis for a technique to be widely applied. There are two problems that need to be addressed for practicability. The first one is the transparency to the user, which means that the deployment of the network address shuffling mechanism must not change or influence the normal user operation greatly. Also, the performance loss induced by the deployment must be acceptable to the user. The second one is the acceptable deployment cost and complexity. If the deployment cost and complexity for deploying a network address shuffling mechanism is too high to accept, the user will refuse to deploy it.

V. CONCLUSIONS

In this work, we identified and described two patterns of the network address shuffling techniques, and then categorized the main existed network address shuffling techniques according to the two patterns. What's more, under each category, we gave a detailed description on each mechanism. Thereafter, we analyzed and summarized them. Finally, we discussed some key issues on implementing an effective mechanism in network address shuffling. With this work, we hope to stimulate more follow-up research in this field.

ACKNOWLEDGMENT

This work was supported by the National Key Basic Research Program of China (973 Program). [2012CB315906, Reconfigurable Network Emulation Testbed for Basic Network Communication]

REFERENCES

- [1] Y. Liu, W. Peng and J. Su, "A study of IP prefix hijacking in cloud computing networks", *SECURITY AND COMMUNICATION NETWORKS*, vol.7, no.11, pp.2201-2210. 2014.
- [2] T. Wang, H. Wang, B. Liu, B. Ding, J. Zhang and P. Shi, "Further Analyzing the Sybil Attack in Mitigating Peer-to-Peer Botnets", *KSII Transactions on Internet and Information Systems (TIIIS)*, vol.6, no.10, pp.2731-2749. 2012.
- [3] F. Wang, H. Wang, X. Wang and J. Su, "A new multistage approach to detect subtle DDoS attacks", *Math Comput Model*, vol.55, no.1, pp.198-213. 2012.
- [4] *Prism*. [https://en.wikipedia.org/wiki/PRISM_\(surveillance_program\)](https://en.wikipedia.org/wiki/PRISM_(surveillance_program)), 2013
- [5] *The Heartbleed Bug*. <http://heartbleed.com/>, 2014
- [6] Cybersecurity Game-Change Research & Development Recommendations. Available: http://www.nitrd.gov/pubs/CSIA_IWG_%20Cybersecurity_%20GameChange_RD_%20Recommendations_20100513.pdf. 2010.
- [7] Trustworthy CyberSpace: Strategic plan for the federal cybersecurity research and development program. Available: http://www.whitehouse.gov/sites/default/files/microsites/ostp/fed_cybersecurity_rd_strategic_plan_2011.pdf. 2011.
- [8] National Cyber Leap Year Summit 2009 Co-Chairs' Report. https://www.nitrd.gov/nitrdgroups/index.php?title=Category:National_Cyber_Leap_Year_Summit_2009. 2009.
- [9] T. E. Carroll, M. Crouse, E. W. Fulp and K. S. Berenhaut, "Analysis of network address shuffling as a moving target defense", in *Proc of 2014 IEEE International Conference on Communications (ICC)*, 2014, pp.701-706.
- [10] D. Kewley, R. Fink, J. Lowry and M. Dean, "Dynamic approaches to thwart adversary intelligence gathering", in *Proc of DARPA Information Survivability Conference and Exposition II*, 2001, pp.176-185.

- [11] M. Atighetchi, P. Pal, F. Webber and C. Jones, "Adaptive use of network-centric mechanisms in cyber-defense", in *Proc of Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, 2003, pp.183-192.
- [12] M. Sifalakis, S. Schmid and D. Hutchison, "Network Address Hopping: Enhancing Network Security", Lancaster University Technical Report COMP-007-2004, 2004.
- [13] L. Shi, C. Jia and S. Lü, "DoS Evading Mechanism upon Service Hopping", in *Proc of IFIP International Conference on Network and Parallel Computing Workshops*, 2007, pp.119-122.
- [14] L. Shi, C. Jia and S. Lü, "Research on end hopping for active network confrontation". *Journal on Communications*, vol.29, no.2, pp.106-110. 2008.
- [15] M. Dunlop, S. Groat, W. Urbanski, R. Marchany and J. Tront, "MT6D: A Moving Target IPv6 Defense", in *Proc of MILCOM 2011*, 2011, pp.1321-1326.
- [16] M. Dunlop, S. Groat, W. Urbanski, R. Marchany and J. Tront, "The Blind Man's Bluff Approach to Security Using IPv6", *IEEE Security and Privacy*, vol.10, no.4, pp.35-43. 2012.
- [17] MORPHINATOR. <http://defense-update.com/tag/morphinator>, 2012
- [18] S. Antonatos, P. Akritidis, E. P. Markatos and K. G. Anagnostakis, "Defending Against Hitlist Worms Using Network Address Space Randomization", in *Proc of WORM '05*, 2005, pp.30-40.
- [19] J. Yackoski, P. Xie, H. Bullen, J. Li and K. Sun, "A Self-shielding Dynamic Network Architecture", in *Proc of MILCOM 2011*, 2011, pp.1381-1386.
- [20] J. H. Jafarian, E. Al-Shaer and Q. Duan, "Openflow Random Host Mutation: Transparent Moving Target Defense Using Software Defined Networking", in *Proc of HotSDN '12*, 2012, pp.127-132.
- [21] E. Al-Shaer, Q. Duan and J. Jafarian, "Random Host Mutation for Moving Target Defense", *Security and Privacy in Communication Networks*, pp.310-327, 2013.
- [22] J. H. Jafarian, E. Al-Shaer and Q. Duan, "Spatio-temporal Address Mutation for Proactive Cyber Agility Against Sophisticated Attackers", in *Proc of MTD '14*, 2014, pp.69-78.
- [23] Q. Jia, K. Sun and A. Stavrou, "MOTAG: Moving Target Defense against Internet Denial of Service Attacks", in *Proc of 22nd International Conference on Computer Communications and Networks (ICCCN)*, 2013, pp.1-9.
- [24] A. Clark, K. Sun and R. Poovendran, "Effectiveness of IP address randomization in decoy-based moving target defense", in *Proc of 52nd Annual Conference on Decision and Control (CDC)*, 2013, pp.678-685.
- [25] D. C. MacFarland and C. A. Shue, "The SDN Shuffle: Creating a Moving-Target Defense Using Host-based Software-Defined Networking", in *Proc of MTD '15*, 2015, pp.37-41.
- [26] R. Zhuang, S. A. DeLoach and X. Ou, "Towards a Theory of Moving Target Defense", in *Proc of MTD '14*, 2014, pp.31-40.



Xiaofeng Wang has been working as an assistant professor in the College of Computer at National University of Defense Technology in China. He received the B.S., M.S. and Ph.D. degrees in computer science from National University of Defense Technology in 2004, 2006 and 2009, respectively. His research interests include trustworthy networks and systems, applied cryptography, network security.



Yulei Yuan has been working as a research assistant in the College of Computer at National University of Defense Technology in China. He received the B.S., M.S. and Ph.D. degrees in geodetic science from Information Engineering University in 2006, 2009 and 2012, respectively. His research interests include satellite navigation, software engineering and computer network.



Sudan Li has been working as an associate professor in the College of Computer at National University of Defense Technology in China. He received B.S., M.S and Ph.D. degrees in control theory and engineering from National University of Defense Technology in 1995, 1998 and 2001 respectively. His research interests include network architecture and protocol, router software design.



Guilin Cai is a doctor student in the College of Computer at National University of Defense Technology in China. She received the B.S. and M.S. degrees in computer science from National University of Defense Technology in 2005 and 2010, respectively. Her main research interests include cyber security and proactive defense.



Baosheng Wang has been working as a professor and PhD supervisor in the College of Computer at National University of Defense Technology in China. He received the B.S., M.S. and Ph.D. degrees in computer science from National University of Defense Technology in 1992, 1995 and 2005, respectively. His research interests include router architecture, routing protocol and cyber security.