SAND2015-10117C

Operational Cost of Deploying Moving Target Defense

Defensive Work Factors

Brian Van Leeuwen, William Stout, and Vincent Urias
Sandia National Laboratories
Albuquerque, New Mexico, USA
{bpvanle, wmstout, veuria}@sandia.gov

***Abstract*—Moving Target Defense (MTD) is the concept of controlling change across multiple information system dimensions with the objective of increasing uncertainty and complexity for attackers. Increased uncertainty and complexity will increase the costs of malicious probing and attack efforts and thus prevent or limit network intrusion. As MTD increases complexity of the system for the attacker, the MTD also increases complexity in the desired operation of the system. This introduced complexity results in more difficult network troubleshooting and can cause network degradation or longer network outages.**

In this research paper the authors describe the defensive work factor concept. Defensive work factors considers in detail the specific impact that the MTD approach has on computing resources and network resources. Measuring impacts on system performance along with identifying how network services (e.g., DHCP, DNS, in-place security mechanisms) are affected by the MTD approach are presented. Also included is a case study of an MTD deployment and the defensive work factor costs. An actual experiment is constructed and metrics are described for the use case.

Keywords—moving target defense; metrics; defensive work factors

I. INTRODUCTION

Moving Target Defense (MTD) recently has received much attention in technical publications. The publications describe various approaches to implementing MTD. The publications describe MTD approaches that periodically change some attribute of the information system. The attribute that is changed, in most cases, is one that an adversary attempts to gain knowledge of through some reconnaissance and may use its knowledge of the attribute to exploit the system. The fundamental mechanism an MTD uses to secure the system is to change the system attribute such that the adversary never gains the knowledge and cannot execute an exploit using the knowledge prior to the attribute changing value. Thus, the MTD keeps the adversary from gaining the knowledge of attributes necessary to exploit the system

In most cases changing the value of an attribute of the system makes exploiting the system more challenging. However, it should be noted that in some cases an MTD can make a system more vulnerable to an attack. An example is an MTD that rotates through a number of servers, each running a similar application on a different operating system (OS). In this case, the adversary needs to identify a vulnerability in only one OS and has potentially multiple OSs from which to select.

A fundamental limitation of proposed MTD approaches is the lack of quantifiable metrics that describe MTD approach. Metrics are necessary to quantify the security improvement obtained by deploying the MTD on a system that may be subject to a particular threat. Additionally, the cost to the information system defender should also be quantified. In general, deployment of any security solution has a cost and benefit that typically is used by a system defender to make decisions on what security solution provides the greatest impact given the available resources. In this paper the study of a novel approach for determining the cost to resources and performance of an MTD approach is presented. The cost of deploying an MTD approach is considered the defensive work factor.

A. Our Contribution

We propose to employ the extensive advances made in system instrumentation as applied to Application Performance Monitoring (APM) and Network Performance Monitoring (NPM) to measure MTD defensive work factor. We briefly identify the aspects of APM and NPM instrumentation applicable to quantifying MTD metrics. We also introduce the classes of MTD approaches our proposed metrics are applicable. Additionally, the authors have developed an MTD technique that uses software defined networking (SDN) to perform a sophisticated IP address randomization [1]. Using the IP Randomization technique we create numerous experiments with extensive instrumentation to collect data that quantitatively defines the defensive work factor. Furthermore, we describe impacts that some MTD approaches have on the information system's traditional security approaches such as intrusion detection systems (IDS) and intrusion protection systems (IPS). In these cases, the MTD approach may vary an aspect of the information system that another defensive measure may monitor. As an example, consider a network-based intrusion detection system (NIDS) performing security analysis on packets passing between subnets that may include an MTD measure that is based on an IP address randomization. For a stateful NIDS the IP randomization approach would render it ineffective if multiple packets intended for a specific destination appear to the NIDS to be addressed to multiple destinations. Thus, the impacts that MTD has on a security measure's ability to perform its security function must also be considered in MTD defensive work factor analysis.

To our knowledge, this is the first MTD defensive work factor study on describing the resource and performance cost of deploying MTD approaches.

B. Related Work

Identifying or creating metrics to quantitatively describe the efficacy of an MTD approach is an important problem and is a subset of the well-known hard problem of cyber security metrics. The difficulties with creating meaningful security metrics and limitations of measuring trust in an absolute sense are described in [2]. Metrics that quantitatively describe the efficacy of MTD from a defensive position continues to receive attention and is an important science. Various approaches to measure defensive effectiveness of an MTD can be found in numerous publications [3]. Metrics are used, in many cases, to inform decisions on applying security measures to achieve some desired security posture. The decision process of determining the security measure to deploy is based on various factors including two primary factors: effectiveness of security measure and cost of the security measure. Cost of security measures can have numerous factors, such as monetary cost, cost to performance, operation, and maintenance of system. Thus, the need for defensive work factor metrics when system defenders are assessing MTD security implementation options.

Application Performance Monitoring (APM) and Network Performance Monitoring (NPM) are disciplines within information system management and monitoring. In its narrowest sense it focuses on the monitoring and availability of a software application; however, in its more accepted broader sense it focuses on the monitoring, operation, and availability of the distributed application and supporting components such as network, databases, middleware, and services. APM and NPM can be effective in describing end-user experience, as well as enable troubleshooting system performance issues. Because of its system wide monitoring, APM and NPM can be used to measure some impacts of MTD on system operation.

The rest of the paper is organized as follows. In Section 2, a description of MTD approaches that our research focuses on is provided. In Section 3, impacts an MTD may impose on a system are presented. Section 4 and 5 describe methods to measure performance impacts component of defender work factors. An example case study of an MTD approach, experiment, data collection and metrics generation is presented in Section 6. Conclusions and further studies are presented in Section 7.

II. Types of Moving Target Defense Techniques

MTD techniques can be applied to most any part of an information system with the objective of inducing changes to the system structure, architecture, and/or parameters. Thus, segmenting MTD techniques into groups is a common approach. In this paper, we segment MTD techniques into two classes: network-based MTD techniques and host-based MTD techniques. In test cases, the MTD technique will fall into a single class or the MTD technique may be a combination of multiple varying system structure, architecture, or parameters and thus fit into both classes.

A. Network-based MTD Techniques

The network-based MTD technique class includes MTD techniques that dynamically vary network aspects of a distributed information system. The various aspects that provide network connectivity and enable system transactions across multiple computing platforms are candidates for MTD techniques. Example techniques included in this class are dynamic IP address and/or port randomization [4], routing path randomization [5], and proxy-location randomization [6]. The fundamental idea for these techniques is periodically changing the structure of the network an adversary must use to access resources or data in the protected information system.

Challenges with varying the structure or parameters of the underlying network of the information system include synchronizing legitimate access for components of the distributed information system while disrupting reconnaissance and attacks from an adversary. Additionally, network protocols must operate through a network-based MTD technique, such as a TCP connection that must be maintained and not be broken and reestablished. Finally, nodes under the MTD umbrella must be able to connect or communication with legacy or unsupported devices, either network elements or endpoints. This later concern may actually have a greater impact on network services, security and packet forwarding than affecting the adversary's attempts at reconnaissance or attack.

B. Host-based MTD Techniques

Host-based MTD techniques are those that vary some aspect of the computing platform itself. The technique may vary aspects of OS, application code, or properties of the platform [7]. More specifically host-based techniques often use one or more of the following mechanisms:

Dynamic Runtime Environment: The OS dynamically varies some aspect of the runtime environment presented to the application. The approach includes mechanisms that vary the location in memory of application program code, libraries, stack/heap, and functions. Also included are mechanisms that vary the interface presented to the application (e.g., system calls for I/O devices).

Dynamic Application Code and Application Data: Includes mechanisms that vary some aspect of application code during runtime (e.g., instruction ordering) or representation of application data (e.g., encoding).

Dynamic Platform: Includes mechanisms that result in variation of host platform properties such as OS type and/or version or CPU architecture.

As an example, a commonly deployed host-based MTD approach is Address Space Layout Randomization (ASLR) [8]. This dynamic runtime environment approach is a memory-location technique that can assist in defeating code-

injection attacks by randomizing the memory layout of application program code.

III. MTD Defensive Work Factor

The threat surface of an information system may contain any number of security holes that may be plugged if given a suitable mechanism to block, migrate or obscure the vulnerabilities. When considering the development, implementation and deployment of an MTD approach, often the focus is strictly adversary-based. With respect to threat surfaces, it is completely appropriate to consider an adversary's existence, tactics, techniques, and procedures (TTP)s and perspective when designing and deploying the MTD technique. As a result, much literature has attempted to capture the costs to adversary's work flow while combating a defense. However, what cost does the defender persevere to stand up and maintain this defense? We have identified four areas that should undergo examination when considering the deployment of an MTD.

What does implementation impose?
Most networked systems are heterogeneous in nature. Green fields are quite rare; enterprises are often evolved over the course of many technology iterations and refreshes such that Mustangs share the roads with Pintos. Thus, implementation must consider the operating expenses (OPEX) as well as the capital expenses (CAPEX). For the latter, legacy equipment may not be upgradeable. Hence, budgets must allow for the replacement or retrofit of equipment if it needs to meet the MTD technology requirements. If equipment cannot be replaced or retrofitted – then the MTD technology must be modified to include other techniques to envelop the equipment into the MTD system, else the defenders assume the risk of leaving them in-place as-is. Regarding OPEX, new technology requires new mindsets and new training. Those responsible for network defense must overcome learning curves to properly operate the technology. Furthermore, if the MTD technology is a separate system and not capable of integrating with defenses already in-place, then defense workflows may increase to maintain the product. If neither of these options is feasible, then the enterprise owners must contract with another party to address these issues – thereby increasing the budget as well.

How does the MTD system affect performance?
No function operates without the use of resources, and no network exists without communication. The MTD approaches we discuss here involve deployment on either the host, or in the network; both approaches require CPU cycles and memory to function and may result in the 'trickle-up' consequence to affect higher level applications and services. What overhead comes with host-based techniques that may require longer memory access times, or memory state switching? How do packets transmission times change when assigning new IPs or ports – and how do intermediate switches and routers respond with altered packet addresses? Furthermore, we have to consider systems that rely on some of the static aspects of systems. How are DNS and DHCP affected; must these services be drawn into the MTD system (and how)? Considerations must be given to defenses like IDS or behavior-based analytics; will constantly fluctuating network attributes be misconstrued as maliciousness?

What is the stability of the MTD System?
Some MTD approaches do well on single systems, such as host-based defenses. However, network approaches involve device and system components that must often work in concert with each other. Consider SDN-based approaches that revolve around an SDN-controller. How large can the network scale before the controller responses are too delayed? Is there an upper-bound to the number of connections the controller can facilitate? If boundaries are imposed, then the defender must find the appropriate amount of movement that balances scale and effectiveness – which may not be an easy task to determine if only an operation network exists to test on. If the MTD is comprised of several systems, consideration must be given to failure modes. If there aren't backup systems, should MTD enclaves fail opened or closed? If they fail open, will they lack the ability to communicate with those enclaves still operating? What is the cascading effect of MTD system failures?

What is the effectiveness of the MTD system?
Does the MTD technique do what it's supposed to, when it matters the most? If the approach does not expectantly thwart attack, then it should with high probability reduce the risk of attack. This consideration is heavily dependent on the threat surface, the approach, and the number of available parameters to modify to increase entropy for the attacker. Furthermore, it is also dependent on attacker sophistication, known attack vectors, and likely the most the vexing factor, unknown attack vectors. When unknowns are included in the problem space, the solution space tends toward intractability. It is our opinion that this notion impacts the determination of adversary work factors for MTD approaches. Approaches to measure the effectiveness of the MTD are not described in this paper.

Using the discussion points above, we have distilled the areas of concern to derive a set of metrics to measure defensive work factors for an MTD. Table 1 below attempts to generalize this set; the metrics are not meant to describe a finality, but rather an initial template to build upon. As MTDs evolve with technological advances, so should the defense work factors.

Table 1: MTD Defensive Work Factors

Category	Metric	Units
OPEX Implementation	Operator workload	man-hours
	Operator learning-curve	man-hours
	Third-party O&M	cost
CAPEX Implementation	New procurement	cost
	Retrofit	cost
	Upgrade	cost
Performance-Network	Packet latency	time
	Packet jitter	time
Performance-Host	Process execution	time

	Memory consumption	bytes
	CPU consumption	percentage
Performance-Application	Transaction time	time
Network-Services Impact	Applications	Application-based
Host-Services Impact	Applications	Application-based
Network Scalability	Number of nodes	count
Host Scalability	Amount of memory	bytes
	Number of processors	count
	Number of applications	count

We've identified four areas that should be investigated when considering deployment of MTDs. The first is best measured with man-hours and cost. The third area, investigates stability, is a topic for a future research paper. The fourth area, on the effectiveness of defense provided by MTDs, is ongoing research by numerous authors. The remainder of this paper will present methods the MTD effects on system performance.

IV. Defensive Work Factors: Effects on System Performance

APM and NPM focus on monitoring, at a system level, the operation and availability of the distributed application and supporting components such as network, databases, middleware, and services. APM and NPM have the objective of identifying and measuring the individual transaction of an application and identifying and measuring the components of a transaction. As an example, a user may engage an enterprise application via the browser on his local workstation. In this case APM will collect transactional performance data originating at the user's browser. As the transaction engages the server middleware and application additional performance details are collected and reported. Next the application may initiate a connection through the network to a backend database. Performance data from the complete transaction is reported to a central monitoring console. Thus, if an MTD is deployed in any component or system supporting the transaction, its impact on that transaction will be measured.

Numerous host-based MTD approaches will impact host resources such as CPU cycles, memory, or, possibly, disk activity or storage access. APM systems provide for extensive instrumentation of the host system and applications running on the host. The extensive instrumentation can provide detail breakdowns of the response time of transactions and, in cases, detailed response times of the transaction components. Transactions that require network access for external service calls include details on network access and response times. The APM system can produce an extensive volume of time-stamped host-operation and performance data that can be used for comparative analysis.

Similar to APM, NPM can provide details on the host system's interaction with the network and access to remote services and data. Transaction delays associated with network roundtrip times and times to access authentication servers, DNS, databases, and web application servers are available to include in analysis. The NPM data is also time-stamped and can be correlated with the host-based APM data collection results.

In assessing the system performance impacts of an MTD approach that varies host or network based parameters the analysis approach begins with establishing a baseline of a system operation without MTD employed. The baseline performance should include measuring performance of those characteristics that are expected to be impacted by the MTD. Host-based MTD approaches that randomize memory location should include CPU and memory usage along with transaction times for various time-buckets so comparative analysis can be performed when host-based MTD is employed and enabled. The impact to host performance may not be dramatic over the short period but may be more impactful over longer periods of host operation.

Baseline data should also be collected for network performance. Delays associated with host access to remote services and data requiring network access should be collected. This data becomes the baseline which to compare data collected with network-based MTD approaches employed.

With the system fully instrumented and baseline performance data collected the system can be outfitted with the MTD approach. System runs similar to those done for baseline data collection are executed with MTD approach employed. Data is collected and comparative analysis is performed to identify the performance impact of the MTD approach. Furthermore the instrumented system can be used for tuning the MTD approach to identify acceptable operation with acceptable performance impacts.

V. Defensive Work Factor: In-place Security Mechanisms

Identifying the cost of an information system security mechanism is dependent upon an organization's deployment strategy and how effective the security mechanisms are at achieving security objectives. In general, the goal is to create a value proposition which results in a greater value in security than it costs to deploy and manage the security mechanisms. Prior to MTD approaches the value proposition of various types of security mechanisms could be evaluated independently in that the security mechanism acted independently in its ability to perform its security function. However, with the introduction of MTD, parameter values that, independent of MTD, were available to the in-place security mechanism become a parameter value that changes with some MTD approaches. In cases, with MTD the in-place security mechanism cannot depend on a static parameter value.

An example is deploying an IP address randomizing MTD approach with an in-place NIDS. NIDS can detect probes,

scans, malicious and anomalous activity across a network and can also serve to identify general traffic patterns for a network. Depending on the system architecture, cases can occur where the NIDS cannot depend on a static source and destination IP address in order to perform its security operations. In this case, the cost of deploying the IP address randomizer should include its negative impact on the in-place NIDS.

In the case of network-based MTD that affects a parameter used by an in-place security mechanism the in-place security mechanism should be placed on the non-impacted side of the MTD defense. This assures the in-place security mechanism can perform its function on various parameters that have not been affected by the MTD. If an MTD is employed that affects parameters that an in-place security mechanism uses and the in-place security mechanism cannot be placed on the private side (i.e., unaffected side) of the MTD than the in-place security mechanism will be affected. The affect must be thoroughly analyzed and if a reliable analysis cannot be performed the impact should be considered as a worst case result. In cases, a worst case result is the complete removal of the in-place security mechanism.

VI. Applying Defense Work Factors to a Use Case

As a proof of concept for our approach we create a use case where a network-based MTD approach is deployed in a realistic scenario. The use case scenario is an IP address randomization MTD approach leveraging software defined networking (SDN). The MTD system was developed for Industrial Control System applications, specifically Supervisory Control and Data Acquisition (SCADA). The MTD requires Openflow-compatible switches for endpoint network entry. It is these endpoint switches (either installed on the host as virtual switches, or positioned in front of the host as virtual or physical switches) that are controlled to install and strip randomized IP addresses.

To deploy the system, a virtualization platform was used [9]. Endpoints were created as light-weight Linux virtual machines; Open vSwitch (OVS) processes were executed on the Linux endpoints. The SDN Openflow controller was also deployed in the network, to connect to the endpoint OVS and control IP address assignment. As a network-based MTD approach, defensive work factor metrics to capture are Performance-Network, Performance-Application and Network Scalability; for this particular use case we have scoped the tests to just NPM; thus, NPM-type instrumentation of the system is used to collect performance data. To gather Performance-Network data, endpoint agents were used to generate and digest traffic between endpoints. An NPM collector virtual machine was deployed to gather the performance data, normalize it and parse it for statistical analysis.

To meet the testing requirements, a partial factorial experiment design was utilized. Factors for the experiment are shown in Table 2 below.

Table 2. NPM Experiment Parameters

Factorial Experiment Parameters

Experiment Size	10-to-10 Endpoints
Protocols	TCP, UDP
Randomization Technique	IP Address Randomization
Randomization Intervals (s)	10, 30, 60, Random [30,60]
Test Duration (s)	{240, 480}

Baseline measurements for each of the factorial combinations were captured without the MTD system running. The baseline data are then compared to the data captures with the MTD system running. Table 3, Table 4 and Figure 1 describe the bandwidth percentages compared to baseline date for TCP and UDP. Figures 2 depicts the jitter measurements for UDP traffic.

Table 3. TCP Bandwidth Consumption

Test Length	Randomization Interval	BW Percentage of Baseline
240s	10s	0.7117
	30s	0.9087
	60s	0.9523
	[30,60]s	0.9558
480s	10s	0.7246
	30s	0.9104
	60s	0.9673
	[30,60]s	0.9533

Table 4. UDP Bandwidth Consumption

Test Len.	Rand. Interval	Fixed BW (Mb)	BW Percentage of Baseline
240s	10s	{1,10,50,100}	{0.9998, 0.9993, 0.9928, 0.9838}
	30s	{1,10,50,100}	{1, 1, 1, 0.9957}
	60s	{1,10,50,100}	{0.9999, 1, 1, 0.9978}
	[30,60]s	{1,10,50,100}	{1, 1,1, 0.9972}
480s	10s	{1,10,50,100}	{0.9998, 0.9998, 0.9962, 0.9862}
	30s	{1,10,50,100}	{1, 1,1, 0.9954}
	60s	{1,10,50,100}	{0.9999, 1,1, 1}
	[30,60]s	{1,10,50,100}	{1, 1,1, 0.9972}

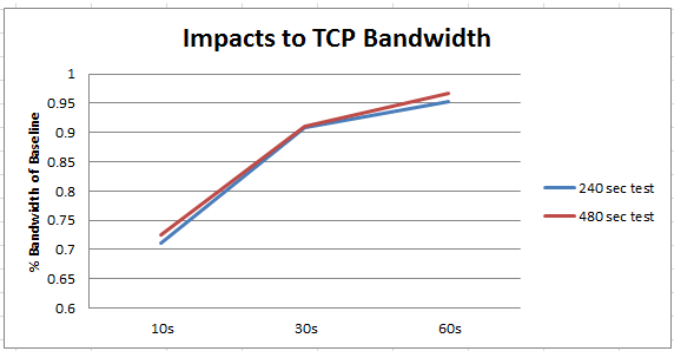

Figure 1. TCP Bandwidth Impacts per Randomization Interval

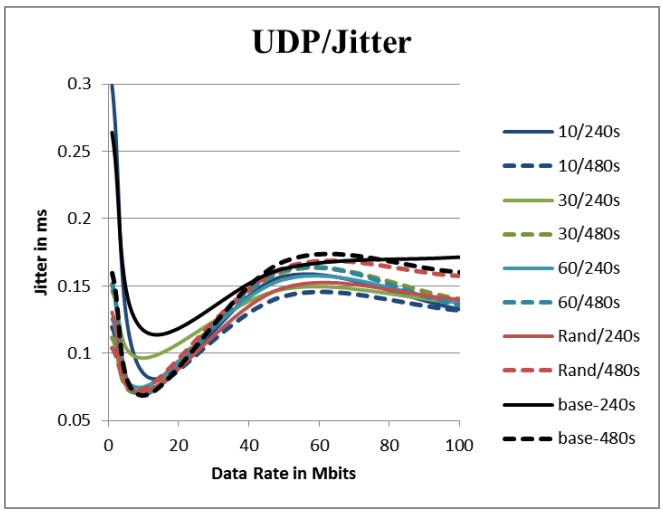

Figure 2. UDP Jitter Measurements

Peak Endpoint Memory Consumption: 24599472B
Peak Endpoint CPU Consumption: 2% Utilization

Given the bandwidth tests, we may observe that the MTD performs best with low-bandwidth systems, such as SCADA networks. With such systems, defenders may be able to increase the amount of randomization (that is, randomization intervals) to better protect their system(s). A bit loss study affirms this projection, as longer randomization times and higher bandwidths diverge from baseline measurements. The MTD approach appears to have little impact on jitter, as the measures closely follow the baseline measurements. Finally, peak memory usage and CPU utilization are not excessive, and would bear little burden on host systems.

VII. Conclusion and Further Study

The research presented in this paper focuses on the development of creating a framework that describes the costs of deploying moving target defense (MTD) approaches. The authors proposed and describe a Defensive Work Factor approach that describes the costs of deploying an MTD from three views:

- What does implementation impose?
- How does the MTD system affect performance?
- What is the stability of the MTD System?

A fourth view should also be considered for an overall evaluation of the MTD approach:

- What is the effectiveness of the MTD system?

Each of the defensive views are defined in the paper and a description of how to apply APM and NPM instrumentation approaches to provide quantitative values for metrics measuring MTD effects on host and network performance is presented. A use case applying the defensive work factor approach is provided that demonstrates the effectiveness of the approach. Baseline test metrics are collected and compared to metrics with the MTD employed. Our use case deploys an IP address randomization MTD that uses software defined networking (SDN) technologies.

The research team recognizes the value of MTD technologies and will continue to develop metrics that describe the costs of deploying MTD and the impact MTD has on preventing malicious attacks on a system or preventing data comprise. Further studies will include assessing MTD on the stability of the system and further development of metrics describing effectiveness of preventing malicious activity on an MTD-protected system.

References

[1] Chavez, A., Hamlet, J., Lee, E., Martin, M., and Stout, W. (2015) "Network Randomization and Dynamic Defense for Critical Infrastructure Systems," Sandia National Laboratories Report – SAND2014-16446PE, 2015.

[2] M. Torgerson, "Security Metrics for Communication Systems," 12th International Command and Control Research and Technology Symposium, Newport, Rhode Island, June 19-21, 2007.

[3] G. Cybenko, J. Hughes, "No Free Lunch in Cyber Security," MTD Workshop, Scottsdale, Arizona, November, 2014.

[4] Dunlop, Matthew; Groat, Stephen; Urbanski, William; Marchany, Randy; Tront, Joseph, "MT6D: A Moving Target IPv6 Defense," MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011 , vol., no., pp.1321,1326, 7-10 Nov. 2011

[5] E. Al-Shaer and Q. D. J. Jafarian, "On the Random Route Mutation Moving Target Defense," in National Symposium on Moving Target Research, June 2012.

[6] K. S. Quan Jia and A. Stavrou, "Motag: Moving target defense against internet denial of service attacks." In International Conference on Computer Communications and Networks (ICCCN), 2013.

[7] H. Okhravi, M.A. Rabe, et.al., "Survey of Cyber Moving Targets," Lincoln Laboratory – Massachusetts Institute of Technology Technical Report, September 2013.

[8] H. Shacham, M. Page, B. Pfaff, E. Goh, N. Modadugu, D. Boneh. "On the effectiveness of address-space randomization." ACM Conference on Computer and Communications Security (CCS), CCS '04, pages 298–307, New York, NY, USA, 2004.

[9] Urias, V.; Van Leeuwen, B.; Richardson, B., "Supervisory Command and Data Acquisition (SCADA) system cyber security analysis using a live, virtual, and constructive (LVC) testbed," MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012 , vol., no., pp.1,8, Oct. 29 2012-Nov. 1 2012.

[10] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: transparent moving target defense using software defined networking," in Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12. New York, NY, USA: ACM, 2012, pp. 127–132.