If You Can't Measure It, You Can't Improve It: Moving Target Defense Metrics

Stjepan Picek
Cyber Security Research Group,
Delft University of Technology
Delft, The Netherlands
stjepan@computer.org

Erik Hemberg
MIT, CSAIL
Cambridge, MA, USA
hembergerik@csail.mit.edu

Una-May O'Reilly
MIT, CSAIL
Cambridge, MA, USA
unamay@csail.mit.edu

ABSTRACT

We propose new metrics drawing inspiration from the optimization domain that can be used to characterize the effectiveness of moving target defenses better. Besides that, we propose a Network Neighborhood Partitioning algorithm that can help to measure the influence of MTDs more precisely. The techniques proposed here are generic and could be combined with existing metrics. The obtained results demonstrate how additional information about the effectiveness of defenses can be obtained as well as how network neighborhood partitioning helps to improve the granularity of metrics.

KEYWORDS

Moving Target Defense, Metrics, Objective Space, Network Neighborhood Partitioning

1 INTRODUCTION

Cyber attacks are becoming more serious with constant improvements in attackers' strategies. In most of the modern systems there is a significant amount of asymmetry between the attackers and defenders. That asymmetry usually works in the favor of the attacker since he can decide what and when to attack while the defender has most often either a static defense or a defense that is changing in a predefined manner. One category of defender actions that encompasses a wide range of potential mechanisms to balance the asymmetry is called the **Moving Target Defense** (MTD) [1]. MTD represents an approach of adding dynamics to the system through changing its configuration through the time.

To evaluate the effect of a certain MTD in a qualitative manner can be a difficult task on its own and good metrics are always needed in order to conduct the MTD effectiveness evaluation. Accordingly, there are numerous works investigating new metrics and moving target defenses [3, 5–9].

We recognize two broad MTD metrics problems that we aim to address here. The first problem stems from the fact that no single metric is powerful enough to capture all relevant behaviors of a network and changes incurred by a defense. We consider MTDs on a more general level and propose a set of metrics that are able to capture additional information about the network and influence of MTDs. For these metrics we draw inspiration from the domain of fitness landscape analysis for the metaheuristic algorithms. There, such analysis is traditionally used in order to analyze the characteristics of various classes of problems as well as a tool in the design of new optimization algorithms [4].

The second problem lies in the size of a network under investigation. If a network is large and the MTD effect relatively small (or local), a metric will reflect that change but there will be no information exactly where the change occurred. If we assume that the attacker can be present in any node of a network, an MTD can be evaluated as a successful one (based on the magnitude of the change it caused) but in fact be useless if all changes are done in different (wrong) parts of a network. Accordingly, we propose the technique we call Network Neighborhood Partitioning where a change is captured only if it occurs in a certain neighborhood of a selected node.

2 PRELIMINARIES

Let N be a graph representing the physical topology of a network. The nodes in the network can be switches, hosts, firewalls, routers, etc. We continue by calling all nodes hosts, and specifying the difference as necessary. We represent the network N as an n-tuple: $< H, R, C, RH >$. Here, H represents the hosts, R represents the resources, C represents the connections between hosts (i.e., their adjacencies), and RH represents the resources available for each host.

We consider several general Moving Target Defenses (since the metrics we propose are of a general nature, other MTDs can be also used):

- Add/remove a host (honeypot). A honeypot is isolated and monitored by defense system and users accessing it are being recognized as attackers since legitimate users do not enter honeypots. Each honeypot has resources just as the real hosts do.
- Change a connection. This action corresponds to rerouting/mutating traffic between hosts.
- Change a resource. This action corresponds to migration based techniques.

We emphasize that we list MTDs that are connected with the description of a network as used in this paper. Naturally, a network is a much more complex system and it can consist of more elements and consequently different MTDs then become relevant. Still, those

MTD'17, October 30, 2017, Dallas, TX, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5176-8/17/10…$15.00
https://doi.org/10.1145/3140549.3140558

differences do not influence the applicability of approaches we present here.

3 NEW METRICS AND NETWORK NEIGHBORHOOD PARTITIONING

In order to define MTD metrics based on the fitness landscape, we start with some basic definitions. Note that we use standard definitions that are adapted to our network model and MTDs. A neighborhood represents all hosts that can be reached from a certain host in one move, i.e., where there is a direct connection between the hosts. Fitness landscape FL is an n-tuple $< N, f >$ where N represents the network and f is the function representing the evaluation metric. The objective space O consists of the results of metrics (i.e., solutions as we refer them in the rest of the paper) evaluated on a network N. In order for a metric to produce different results, the network configuration needs to change. Those changes can be done by running MTDs.

3.1 Distance Definition

We can have a number of different distance measures with respect to the effect of applying an MTD. Formally defined, a distance d on a set S is a mapping $S \times S \rightarrow R^+$ such that:

- $d(x, y) = 0$ if and only if $x = y$.
- $\forall x, y \in S \times S : d(x, y) = d(y, x)$.
- $\forall x, y, z \in S \times S \times S : d(x, y) + d(y, z) \geq d(x, z)$.

3.2 Metrics in the Objective Space

First, we briefly discuss the difference between metrics in the network space (i.e., “standard" metrics) and metrics in the objective space. Intuitively speaking, “standard” metrics work on elements of the network (e.g., the mean of path lengths metric [2]), which are also elements that the attacker can attack. The metrics in the objective space work on the results of metrics in the network space (recall, those results are called solutions). The metrics in the objective space can give further insights about the quality of conducted MTDs as observed through applied metrics.

When discussing metrics in the objective space, we require a number of network configurations in order to obtain a set of solutions. To obtain those network configurations, we can:

1. Run an MTD a number of times always starting from the same initial network configuration. This option works only if the MTD results in random changes, otherwise the solutions will be the same.
2. Run an MTD a number of times always starting from the same initial population. For each run allow different number of MTD iterations. This technique can be used when MTDs produce random changes but can be misleading if the MTD has a notion of goodness and can converge since then more iterations will usually mean better solutions.
3. Run the same MTD on a number of different randomly created networks. This option is less useful when considering a certain defined network configuration but can be helpful in assessing the general effectiveness of MTDs.
4. Partition network into a number of subnetworks where each solution corresponds to one partition.

The **amplitude** $Amp(N)$ represents the distribution of solutions in the objective space, i.e., the difference between the best and the worst solution:

$$Amp(S_N) = \frac{|S_N|(\max_{s \in S_N} f(s) - \min_{s \in S_N} f(s))}{\sum_{s \in S_N} f(s)}. \tag{1}$$

Here, S_N represents the set of different network configurations, $|S_N|$ is the cardinality of that set, and $f(s)$ represents the solution obtained by evaluating the network.

The **relative variation of the amplitude** Δ_{Amp} of a randomly generated network N_R and network N is defined as:

$$\Delta_{Amp} = \frac{Amp(N_R) - Amp(N)}{Amp(N_R)}. \tag{2}$$

Positive value means that the amplitude of a random network is larger than the amplitude of a network N, which means that the random network configurations have larger variations of solutions than MTD is able to produce.

The **average gaps of the relative gaps** G between a set of solutions and the best known solution equals:

$$G = \frac{\sum_{s \in S_N}(f(s^*) - f(s)}{|S_N| \cdot f(s^*)}, \tag{3}$$

where $f(s^*)$ represents the best known solution. The smaller the gap, the less diversity of solutions.

The **average length of the walks** $L_{avg}(S_N)$ is the number of network changes (e.g., MTD iterations) needed before the solution reaches some user-defined value σ (ideally, an optimal value):

$$L_{avg}(S_N) = \frac{\sum_{s \in S_N, f(s) \leq \sigma} len(s)}{|S_N|}. \tag{4}$$

Here, $len(s)$ denotes the number of times network changed before reaching the value σ. The average length of a walk can be used to infer information about the ruggedness of the landscape. The shorter the walk, the less effort needed to reach a certain network configuration.

Autocorrelation function measures the correlation of the solutions with a distance equal to d:

$$\rho(d) = \frac{\sum_{s, t \in N \times N,\ dist(s,t)=d} (f(s) - \bar{f})(f(t) - \bar{f})}{n \cdot \sigma_f^2}. \tag{5}$$

Here, n is the number of solutions with a distance d. When considering neighborhood solutions, ρ value close to 0 indicates that the variation between two neighbors is equal on average to the variation between any two solutions. This metric belongs to the interval $[-1, 1]$ where the closer is the absolute value to 1, the larger is the correlation between the solutions.

3.3 Network Neighborhood Partitioning

One important aspect of the network is its size. A metric result for a small network can significantly differ when compared with a metric result for a large network. Since most of the attacker models assume that the attacker can be in any node of the network, having MTDs that influence only nodes that are far from the attacked node can be misleading when assessing the security of a network. We propose to partition the network with respect to the neighborhoods as defined through the neighborhood size parameter. We assume that all the nodes in the network are connected. The first step in our algorithm is to decide on the size of the neighborhood, which can be any value between 1 and the maximal distance in the network. Once that is selected, partitioning can start where each partition consists of all the elements in the neighborhood of a randomly selected element. The algorithm runs either until all elements are considered or some predefined number of partitions is created and evaluated.

We give the pseudocode for the Network neighborhood Partitioning in Algorithm 1. If the neighborhood size results in splitting the network to a partition that consists of only one node (i.e., if the neighborhood size is n and there are elements at the distance $n + 1$ connected only to a node in that partition) then those leaf nodes can be included in the partition. Alternatively, we create a partition with the neighborhood size $n - 1$. The choice of the technique depends on the required number of partitions.

Algorithm 1 Network Neighborhood Partitioning algorithm.

```
Input:
$list_{all}$ – a list containing all nodes of a network,
$M$ – metric that is being evaluated, e.g., the mean of path lengths
find max neighborhood size $T_{max}$
select neighborhood size $T$ where $T \leq T_{max}$
repeat
    randomly select node $i$ from $list_{all}$
    remove node $i$ from $list_{all}$
    form partition $P$ starting at node $i$ of size $T$
    remove all nodes belonging to partition $P$ from $list_{all}$
    measure $M$ for partition $P$
    store metric result for partition in $V_i$
until $|list_{all}| = 0$
```

The network partitioning algorithm can be also adjusted to capture different parts of a network:

(1) Run the algorithm a number of times (user-defined parameter) where any of the nodes can be in more than one partition. With it, we ensure that each partition consists of all the nodes up to the maximal neighborhood size T.
(2) Run the algorithm a number of times (user-defined parameter) where every partition has exactly D nodes. Each of the nodes can be in more than one partition. By doing so, we ensure that the subnetworks are of the same size which can help in highly connected networks.

4 SIMULATION RESULTS

We consider as the metric in the network space the mean of path lengths [2]. The mean of path lengths len is the arithmetic mean for all path lengths. We emphasize that the used metric serves only as an example to run our analysis and is consequently of secondary importance.

The mean of path lengths before the MTD (iteration 0) equals 4.17 and after the MTD (iteration 1) equals 4.04. Based on those results, we assume that the attacker has a slightly shorter path to attack on average after the MTD. Now, we let MTD run for several more iterations and we give results in Table 1. For comparison purposes, we apply MTD_2 on the same network. We emphasize that the results we give here serve only for illustration purposes and they do not represent any specific MTD. Consequently, we are not concerned how a certain MTD is working or whether it actually makes a good defense. The only important aspect is that by applying MTDs we are obtaining different network configurations.

Table 1: Mean of path lengths, MTD_1 and MTD_2.

Iteration	$len(MTD_1(N))$	$len(MTD_2(N))$
0	4.17	4.17
1	4.04	4.21
2	4.16	4.16
3	4.00	4.29
4	4.28	4.46
5	4.46	4.07
6	4.42	4.15
7	4.38	4.46
8	4.29	4.20
9	4.16	4.17

Table 2: Results for the metrics in the objective space.

	$Amp(N)$	Δ_{Amp}	G	$L_{avg}(S_N)$	$\|\rho(1)\|$
MTD_1	0.11	-0.30	0.05	0.50	0.57
MTD_2	0.09	-0.01	0.05	0.40	0.73

In Table 2 we give results for metrics in the objective space for defenses MTD_1 and MTD_2. The amplitude of a random set of networks (10 random networks of the same cardinality as network N) equals 0.083. We select the value σ equal to 4.3 when calculating the average length of a walk. When working with the autocorrelation function, we select the distance d equal to 1 where we consider the number of hosts in the network as a distance measure. Consequently, we consider only those iterations where the difference between the number of hosts for network configurations equals 1, which is 5 network configurations for the first MTD and 7 configurations for the second MTD.

The first MTD results in a larger amplitude, meaning the diversity of solutions is larger. As a consequence, the attacker would need to adapt to more diversity, which would make the attack more difficult. When considering the relative variation of the amplitude, we see that both defenses behave better than random networks but the first defense has a larger variation. The average gaps metric does not reveal any differences between MTDs when calculating with a precision of two decimal places. Still, we know the smaller the gap, the less diversity in configurations obtained by running MTD. The average length of the walks shows that the first MTD requires larger number of steps to reach a predefined objective value. Finally, the autocorrelation function shows that the variation of solutions between two neighbors is closer to the variation of any two solutions for the first MTD, which means that such obtained solutions are more difficult to attack.

Based on the results, we can assume that both defenses behave better than simply generating random networks but the first defense MTD_1 offers more diversity in configurations and is somewhat slower in reaching predefined objective value as indicated by the average length of the walks metric.

Table 3: Mean of path lengths for a partitioned network.

Iteration	$len(MTD_1(P_1))$	$len(MTD_1(P_2))$	$len(MTD_1(P_3))$
0	2.50	1.67	2.4
1	2.50	3.14	2.38
2	2.67	2.23	2.38
3	2.67	2.85	2.20
4	2.50	2.85	2.90
5	2.80	2.45	2.90
6	2.80	3.10	2.25
7	2.67	2.23	2.90
8	2.50	3.14	2.38
9	2.67	2.30	2.85

Next, we repeat the experiments but now we also use the Network Neighborhood Partitioning algorithm. Note that in the parentheses we write randomly selected nodes from which the distances are taken to form neighborhoods. When calculating the mean of path lengths we calculate them only from those randomly selected nodes to all the other nodes in a partition.

- $P_1(5) : 0, 1, 2, 3, 4, 5, 6, 7, 8.$
- $P_2(11) : 9, 10, 11, 12.$
- $P_3(13) : 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23.$

The partitions when considering the network after the first iteration of MTD_1 are:

- $P_1(5) : 0, 1, 2, 3, 4, 5, 6, 7, 8.$
- $P_2(11) : 9, 10, 11, 12, 15, 16, 17, 25.$
- $P_3(13) : 13, 14, 18, 19, 20, 21, 22, 23, 24.$

We repeat the same evaluation process but we consider each partition as a separate network and we display results here for MTD_1 only. Since the attacked host (e.g., host 0) is now located in only one partition (P_1), we randomly select the attacker node for each partition.

Each partitions' solution represents one solution in S_N. As before, we create a random network in order to calculate Δ_{Amp} but we also partition that random network and obtain the amplitude of 0.1. For calculating the length of the walks, we set σ value to 2.7 and investigate how long MTD needs to reach that value for each partition. The obtained results are given in Table 4. From them we can infer how effective is the defense when measured through the mean of path lengths metric. Here, we can also clearly see the benefit of the Network Neighborhood Partitioning algorithm.

Let us assume the attacker is located in host 0. When looking the whole network (Table 1) after the first iteration of MTD_1, we see the mean of paths length changes which means the network changed and the attacker would need to adapt to that change. When considering information obtained from partitions, we see that the mean of path lengths did not change for the first partition and changed only marginally for the third partition. Since the attacker is in P_1, his immediate neighborhood does not change, which means he can use a large portion of knowledge obtained before MTD was run.

Stepping away from the metrics in the objective space, we briefly discuss the benefit of the Network Neighborhood Partitioning algorithm when considering metrics in the network space. After obtaining the metrics' results for each partition, we take for instance their median value as the indicator of the change done by an MTD. Comparing the results obtained after the first iteration of MTD_1 for the whole network (Table 1) and median of values obtained after the first iteration of MTD_1 for network partitions (Table 3) we can observe that the resulting change is smaller in magnitude in the second case, which suggests that the change was not significant enough to cover all regions of the network.

Table 4: Results for the metrics in the objective space for the partitioned network.

Iteration	$Amp(N)$	Δ_{Amp}	G	$L_{avg}(S_N)$	$\lvert\rho(1)\rvert$
9	0.21	-1.1	0.09	4	0.45

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented several metrics that can infer more information about the effectiveness of MTDs. Besides that, we propose the Network Neighborhood Partitioning algorithm that enables us to achieve better granulation of metrics' results and consequently describe the influence of an MTD more precisely. The simulation results show our metrics are able to characterize several aspects of MTDs which is especially apparent when combined with the Network Neighborhood Partitioning algorithm.

We consider this work to serve as a proof of a concept. In future work, we plan to address the applicability of techniques presented here to real world cases. This would include conducting experiments where MTDs are combined in order to countermeasure risks. Finally, more experiments are needed to determine whether the Network Neighborhood Partitioning can cause situations where the MTDs are actually working worse, resulting in a downgrade of the usability of a network for legitimate users.

ACKNOWLEDGMENTS

This work has been supported in part by Cybersecurity@CSAIL initiative. Additionally, this work has been supported in part by Croatian Science Foundation under the project IP-2014-09-4882.

REFERENCES

[1] Qi Duan, Ehab Al-Shaer, and Jafar Haadi Jafarian. 2013. Efficient Random Route Mutation considering flow and network constraints. In *IEEE Conference on Communications and Network Security, CNS 2013, National Harbor, MD, USA, October 14-16, 2013.* IEEE, 260–268.

[2] N. Idika and B. Bhargava. 2012. Extending Attack Graph-Based Security Metrics and Aggregating Their Application. *IEEE Transactions on Dependable and Secure Computing* 9, 1 (Jan 2012), 75–85.

[3] P. K. Manadhata and J. M. Wing. 2011. An Attack Surface Metric. *IEEE Transactions on Software Engineering* 37, 3 (May 2011), 371–386.

[4] El-Ghazali Talbi. 2009. *Metaheuristics - From Design to Implementation.* Wiley.

[5] Joshua Taylor, Kara Zaffarano, Ben Koller, Charlie Bancroft, and Jason Syversen. 2016. Automated Effectiveness Evaluation of Moving Target Defenses: Metrics for Missions and Attacks. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense (MTD '16).* ACM, New York, NY, USA, 129–134.

[6] Sridhar Venkatesan, Massimiliano Albanese, George Cybenko, and Sushil Jajodia. 2016. A Moving Target Defense Approach to Disrupting Stealthy Botnets. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense (MTD '16).* ACM, New York, NY, USA, 37–46.

[7] Lingyu Wang, Anoop Singhal, and Sushil Jajodia. 2007. Toward Measuring Network Security Using Attack Graphs. In *Proceedings of the 2007 ACM Workshop on Quality of Protection (QoP '07).* ACM, New York, NY, USA, 49–54.

[8] Kara Zaffarano, Joshua Taylor, and Samuel Hamilton. 2015. A Quantitative Framework for Moving Target Defense Effectiveness Evaluation. In *Proceedings of the Second ACM Workshop on Moving Target Defense (MTD '15).* ACM, New York, NY, USA, 3–10.

[9] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese. 2016. Network Diversity: A Security Metric for Evaluating the Resilience of Networks Against Zero-Day Attacks. *IEEE Transactions on Information Forensics and Security* 11, 5 (May 2016), 1071–1086.