MTD in Plain Sight: Hiding Network Behavior in Moving Target Defenses

Tina Moghaddam
The University of Queensland
Brisbane, Australia
t.moghaddam@uq.edu.au

Guowei Yang
The University of Queensland
Brisbane, Australia
guowei.yang@uq.edu.au

Chandra Thapa
CSIRO Data61
Marsfield, Australia
chandra.thapa@data61.csiro.au

Seyit Camtepe
CSIRO Data61
Marsfield, Australia
seyit.camtepe@data61.csiro.au

Dan Dongseong Kim
The University of Queensland
Brisbane, Australia
dan.kim@uq.edu.au

***Abstract*—Virtual IP shuffling moving target defenses (MTD) reduce the attacker's success probability by imposing an unknown window in which they have to complete their attacks on a particular address. Previous work has shown that an attacker who knows the MTD window can greatly increase their attack success rate, and that enough information is leaked onto the network by an MTD trigger for this to be detectable by an attacker analyzing network traffic. In this work, we propose a way to hide when the MTD triggers by generating traffic that mimics the symptoms of the real MTD trigger in the network. These 'mimic' trigger events occur at different times in the MTD interval, thereby deceiving the attacker into detecting the wrong window size. In this paper, we 1) introduce the proposed hiding scheme, 2) discuss three methods of generating the mimic trigger events in our traffic, 3) implement and test one such scheme to show that it is effective at fooling the attacker, and 4) discuss the further challenges that need to be overcome to make this method a viable defense strategy.**

***Index Terms*—Machine learning, moving target defense, network security, adversarial examples**

I. Introduction

Virtual IP Shuffling moving target defenses (MTD) are a class promising proactive defense techniques that assign virtual IP addresses (vIP) to network hosts, and change these vIPs periodically. They reduce the attacker's chances of success by imposing a finite attack window before they lose the information they had collected about the network, and have been shown to be effective against a variety of reconnaissance techniques [1]. However, new work has shown that the most efficient types of MTD implementations leak information onto the network that allows the attacker to detect the MTD interval and trigger time using machine learning [2]. With this information the attacker can plan their attacks to maximize their attack window. This reduces the effectiveness of MTD and means shorter MTD intervals are required to achieve the same security benefit [3].

Adversarial examples have been well researched in other domains as a way for attacker to fool machine learning models [4]. We take inspiration from these types of attackers and apply this idea from the defenders point of view, with the aim of obfuscating the MTD trigger. In our proposed scheme, by generating 'mimic' triggers, the network fools the attacker's model into detecting the incorrect MTD interval and hence removes any advantage that knowing it would provide. To the best of our knowledge, there is no prior work on hiding the operation of an MTD in a computer network from an attacker.

II. Masking MTD

A. Threat Model

The MTD trigger is detectable from the network traffic due to the unique fingerprint that the installation of new forwarding rules creates. The usual scheme of implementing the MTD is completely asynchronous and rules are installed on an as-needed basis. Therefore, alternative schemes for implementing the MTD which do not leave a unique fingerprint are necessarily less efficient. This leads to the idea of obfuscating the MTD trigger not by removing this fingerprint, but by generating additional events with such a fingerprint to hide legitimate triggers.

The attacker is assumed to be outside of the SDN network and able to eavesdrop the connections between a legitimate client and the network. By clustering this eavesdropped data, the attacker is able to detect the MTD trigger interval T as well as the absolute time the MTD triggers with some accuracy. Given this information, the attacker can then time their attacks to maximize their attack window, increasing the chances of attack success [3].

B. Design Parameters

Creating adversarial samples to circumvent machine learning models has been well explored in literature and is generally feasible for simple models [4]. However, there is a significant challenge in creating the adversarial sample's features in the input. In our domain, the issue is that these adversarial situations need to arise within the network before feature extraction by the attacker. This is difficult to achieve because the network has no control over the features that the attacker uses. However, it can have control over the traffic inside the

2833-292X/24/$31.00 ©2024 IEEE
DOI 10.1109/DSN-S60304.2024.00022

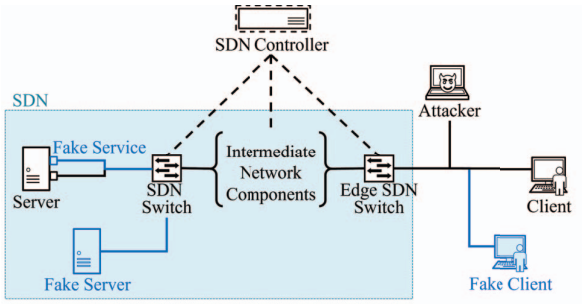

Fig. 1. Our proposed approach: Generating fake MTD triggers in an SDN environment.

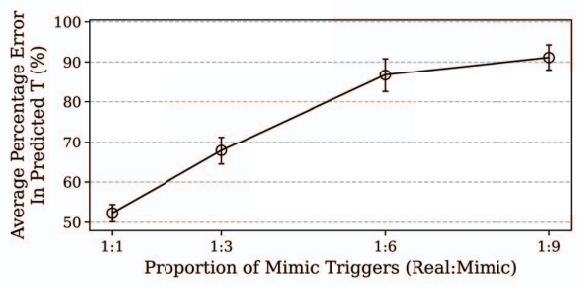

Fig. 2. Effect of the proportion of real and mimic triggers. The legitimate MTD interval is $180s$.

network. Therefore, our approach is to re-create the features present when the MTD triggers in the network as closely as possible. At the same time we want to minimize the disruption to the normal operations of the network. Therefore, the mimic triggers need to be generated in ways that (a) are as transparent as possible to the clients and servers, and (b) have as little impact as possible on performance and incur minimum extra cost.

III. Implementation and Feasibility

In this section we present some possible ways to create the required features by creating traffic in the network. To do this we employ either a fake client, a fake server, or a fake service. Figure 1 depicts a simplified version of the SDN network showing the two ends of communication and the placement of these fake components. The operation of each of these schemes is as follows.

Fake client: There is a fake client in the path of the attacker so that their traffic is sniffed along with legitimate network traffic. The client sends a request to the server, triggering new rules to be installed. This method is completely transparent to the server and legitimate client, however the placement of clients is critical, and this method incurs additional load on the network servers.

Fake server: There is a fake server inside the network, and connections to it cause additional rules to be installed. Here the load on the client and network is unchanged, and legitimate connections with the legitimate server are unaffected. However, running an additional server is costly.

Fake service: Legitimate servers run an additional service which the client can connect to causing the installation of rules. This causes additional load on servers but allows the real and fake service to be isolated without the need for an additional server. However, it is further from normal operation, so it may be distinguishable by the attacker.

IV. Preliminary Experiments

We implemented the fake client scheme for the feasibility study. This scheme creates the least deviation from the normal network operation. To make them indistinguishable to the attacker, the fake client makes requests for webpages on the server that are similar to those of a real client.

We used an MTD interval (legitimate) of 180s, and collected network traffic over 48 hours for each experiment. As the baseline, with no mimic triggers, the attacker could find whether the MTD had triggered with an accuracy of 0.92 ± 0.08. This allowed it to estimate the MTD interval to $179s \pm 3s$, which gives an average error of 1%. We then added mimic triggers with an interval of $M = 20s$, meaning for each real trigger there were nine mimic triggers. We found that the attacker clustered these mimic triggers together with the real triggers with an accuracy of 0.91, which is similar to the accuracy for the baseline. This means the mimic triggers fool the attacker effectively. With the mimic triggers included, the overall accuracy of the attacker at detecting whether the MTD has triggered or not was 0.5 ± 0.2 when averaged over all trials, which is to say random. Note here that the legitimate triggers are still detected accurately, but the fake triggers are also detected, as expected.

The effect of the proportion of real to mimic triggers on the attacker's ability to detect the MTD interval T was also investigated, with the results shown in Figure 2. We can see that as the proportion of fake triggers increases, so does the average percentage error in predicting T, however even a one to one ratio leads to an error of 50%. Increasing the proportion of mimic triggers should have an associated performance cost which also needs to be investigated.

V. Discussion and Future Work

The preliminary results have shown that it is possible to fool the attacker into detecting the MTD interval incorrectly. There is a performance cost associated with adopting an MTD, which means operators are constrained by the performance trade-off which they must balance when choosing their interval. However there is also a performance overhead in implementing the mimic triggers. As future work, the security-performance trade-off for mimicking the MTD needs to be investigated with a comparison to the performance degradation from legitimate MTD triggers. Additionally, the fidelity of fake services and servers can be reduced in order to improve performance, but must be balanced in order to continue to be indistinguishable to the attacker. Questions about the performance of these schemes must be answered in order to to identify the conditions where this obfuscation is useful.

REFERENCES

[1] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense," *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[2] T. Moghaddam, G. Yang, C. Thapa, S. Camtepe, and D. D. Kim, "POSTER: Toward intelligent cyber attacks for moving target defense techniques in software-defined networking," in *Proceedings of the ACM Asia Conference on Computer and Communications Security*, 2023, pp. 1022–1024.

[3] T. Moghaddam, M. Kim, J.-H. Cho, H. Lim, T. J. Moore, F. F. Nelson, and D. D. Kim, "A practical security evaluation of a moving target defence against multi-phase cyberattacks," in *Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2022, pp. 103–110.

[4] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.