Lightweight Security for IoT Systems leveraging Moving Target Defense and Intrusion Detection

Van-Tien Nguyen, Renzo Efrain Navas, Guillaume Doyen

Lightweight Security for IoT Systems leveraging Moving Target Defense and Intrusion Detection

Van-Tien Nguyen[1], Renzo E. Navas[2], Guillaume Doyen[2]
[1]*INSA Toulouse, LAAS-CNRS, Toulouse, France* [2]*IMT-Atlantique, IRISA-CNRS, Rennes, France*
van-tien.nguyen@laas.fr, {renzo.navas, guillaume.doyen}@imt-atlantique.fr

***Abstract*—As more and more devices have communication capabilities, our world is becoming increasingly interconnected. This paradigm is called the Internet of Things (IoT). Most IoT devices have limitations in memory, computing capacity, and energy, thus making impossible to integrate fully-fledged secured solutions into them. Intrusion Detection Systems (IDS) and Moving Target Defense (MTD) are two acknowledged cyber defense techniques that have attracted researchers' attention but need to fit within the constraints of IoT systems. In this paper, based on our previous MTD work, we propose an in-node MTD strategy exhibiting hybrid (i.e., event- and time-based) movement. We specifically explore the MTD interaction with a lightweight detection mechanism to provide reactive defense on top of the by-design proactive-time-based MTD. We implemented and evaluated our proposal in a real IoT platform exposed to a Reduction-of-Quality (RoQ) attack by measuring the round-trip time and packet-loss rate of the system in four scenarios. Notably, we compared our proposal against a time-based-only MTD alternative, which demonstrates the promising results of our hybrid strategy.**

***Index Terms*—IoT, Moving Target Defense, Intrusion Detection System, Lightweight**

I. Introduction

IoT devices are becoming ubiquitous around the world: more than 75 billion devices are predicted by 2025 [1]. Unfortunately, most of them are vulnerable in terms of security because their limitations in energy, memory, or computing power do not allow them to integrate acknowledged and powerful security solutions. As a promising way to overcome this issue, the Moving Target Defense (MTD) [2] cyber defense paradigm, introduced in 2009, can be leveraged. Its application to constrained IoT systems has recently gained interest but is still in development [3]. In an MTD system, some properties change based on time or events in order to increase the attacker's effort. A property changed by MTD, called a Moving Parameter (MP), can be for instance an IP address. Time-based MTD stands for a proactive mechanism and changes the MP value every MTD period independently of any attack occurrence. By contrast, Event-based movement is reactive and changes the MP value if a given event happens such as an attack, for instance. Finally, a hybrid-based MTD considers both types of triggers to operate an MP movement. Time-based MTD is simpler but it gives attackers an amount of exploitation time until the MTD period ends. Event-based MTD is more adaptive in preventing attacks. This movement, however, requires knowledge from other components to recognize special events.

Intrusion Detection System (IDS) stands for the acknowledged reactive technique to secure current information systems. IDS are especially becoming cleverer thanks to artificial intelligence techniques supported by advances in platform/hardware [4]. Nevertheless, fully-fledged IDS solutions embedded into constrained IoT systems are not considered as a realistic and efficient solution due to the gap between the high resource consumption needed by IDS and the low resource limitation of IoT systems. As such, if lightweight in-node IDS solutions can be considered for IoT, they can hardly bring a satisfying security level due to their computation limitations to operate in a standalone way.

As a consequence, coupling MTD with a lightweight in-node IDS can be a relevant way to benefit from these two security mechanisms. An IDS can trigger event-based movement, thus reducing the time allowed to an attacker in a given configuration. Less straightforward is that an IDS can benefit from MTD: MTD system-state knowledge can be used as an input for intrusion detection techniques. In the literature, there are some works [5], [6] that propose the integration of IDS and MTD in IoT but they do not consider the fruitful interaction between these two components for an in-node IoT solution. This motivates us to propose a lightweight solution leveraging reactive MTD interacting with a detection scheme. The key contributions of this work are the following:

1) We propose a lightweight in-node security solution at the network layer for resource-constrained IoT systems. The proposal considers the interaction between a hybrid-based (time- and event-based) MTD strategy and a threshold-based intrusion detection mechanism.
2) We empirically evaluate the network performance of an IoT system implementing our MTD proposal in several scenarios and we also provide the source code of our experiments.
3) We validate the benefit of the hybrid strategy versus a time-based-only MTD solution in our network setting.

The rest of this paper is structured as follows. Section II introduces related work on MTD and IDS in IoT. Section III presents background on IANVS: an MTD framework we consider in our subsequent contributions. Section IV details our proposal. In Section V, we conduct some experiments to evaluate our MTD strategy in a real IoT platform. Finally, in Section VI we briefly conclude and discuss future work.

II. Related Work

MTD and IDS are two concepts of interest in computer security, for a decade for the former and more than three decades for the latter. While MTD attempts to increase the complexity for attackers in learning the system, IDS tries to acknowledge the attacker's presence as soon as possible. Despite a plethora of works on each concept for IoT [3], [7], the combination of both techniques inside an IoT node has not been largely considered in the literature.

In 2019, Giraldo et al. [5] proposed an MTD strategy for a cyber-physical system that randomly turns the sensor data on/off. This MTD scheme reveals the information of stealthy false-data injection attacks performed by a set of compromised nodes, thus improving the accuracy of intrusion detection. However, it does not profit from any interaction between the IDS and the MTD which is time-based only. In 2021, Chen et al. [6] considered an IDS system based on a Moving Target IPv6 Defense (MT6D) mechanism. In this proposal, the authors used an IDS component to detect the attacker's speed in port scanning attacks. Then, they adapted the MTD period to reduce the ability of the attacker to explore the current IPv6 addresses. The greater the cyber threat detected by IDS, the shorter the MTD period length needed. If there is no attack behavior determined, the system does not need to waste resources to rapidly transform the devices' IP. Despite the relevance of this approach, the dynamic interval strategy did mitigate only the reconnaissance phase of an attack. Indeed, although the shorter interval (MTD period) gives the attacker less time to study the system, there remains some chance for the attacker to discover the target MP value and perform some other kinds of attack afterward, which stands for a general limitation of time-based MTD strategies. The survey on proactive and reactive MTD by Cho et al. in 2020 [8] concluded that future research should focus on more adaptive MTD mechanisms whose concepts and techniques are still in their infancy in today's cutting-edge MTD technology. It is particularly critical for the defender to understand and learn an attacker's action or system security conditions in order to make decisions for the best MTD deployment.

As illustrated by this section, even if some early work considers MTD and IDS techniques for an overall security reinforcement, their efficient combination and integration into an IoT node has not been fully addressed. In addition, this in-node combination is an efficient way to intrinsically secure IoT devices without relying on external actors. Such statements motivate us to propose a reactive security strategy for resource-constrained environments that leverages the benefit of both MTD and IDS.

III. Background: IANVS

In 2020, Navas et al. proposed IANVS, an MTD framework for the constrained IoT [9]. IANVS consists of four main components: (1) an Authenticated Key Establishment mechanism (AKE), (2) an Authenticated state Synchronization mechanism (Auth-SYNC), (3) a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG), and (4) a Moving Parameter Mapping (MP-Map). This framework allows to decide *how* and *when* to change the system's MP with strong cryptographic guarantees. The Auth-SYNC component allows every MTD (distributed) entity to agree on a same system state value (e.g., *wall-clock* time) which will allow them to produce the same MP's value at the same moment. The AKE is a component to agree on a fresh shared secret, and the CSPRNG (e.g., a stream cipher) assures the strong cryptographic guarantees of the MP's new values. Finally, the MP-Map maps the CSPRNG's output (i.e., raw bits) to a value in the MP domain (e.g., a valid IP address).

IANVS was instantiated and evaluated at the network layer [9] and at the physical layer [10]. In both proposals, the MP was changed proactively at the end of a static period of time (i.e., the MTD period) to mitigate attacks such as port scanning and jamming. For the network layer instantiation, Navas et al. [9] studied the probability of an attacker to discover an open port for different values of the MP's value range (i.e., Shannon's entropy), the MTD period length, and the attacker's speed. At constant MP entropy, the key factor of the attacker/system interaction is the ratio $\#attacks/MTD\_period$. The choices to improve the system's resiliency are increasing the MP entropy or reducing the MTD period. Furthermore, event-based movement is identified as another dimension for the defender's side but it is proposed as a future direction. In the rest of this work, we explore the coupling of an IDS system with an hybrid-based movement using IANVS.

IV. A Lightweight MTD–IDS solution for the constrained IoT

In this section, we present a security solution that combines a IANVS-based MTD scheme and a lightweight IDS mechanism for constrained IoT nodes. As a research methodology, we considered an empirical approach driven by experiences performed in an IoT testbed implementing IANVS. As such, in the following, we first introduce the type of attack we address. Then, we present the overall architecture of our proposal, focusing on the MTD and IDS main components.

A. Attacker model

In our experimental testbed, the attacker performs a reduction-of-quality (RoQ) attack [11], which stands for a particular Denial-of-Service (DoS) attack, operating in two phases: (1) Reconnaissance, and (2) Exploitation.

1) Reconnaissance phase (R. phase): The attacker is located outside of a network domain to secure. Consequently, he is not aware of the open port in an IoT server device used by legitimate clients to access a shared resource, except from conducting a port scan. This attacker tries to be stealthy by sending scanning packets at a low rate.

2) Exploitation phase (E. phase): Once the attacker discovers the open port, he sends high-rate application packets to the IoT server's open port to drastically increase the response time of legitimate clients' requests or, even worse, make the IoT node down.

Since the IoT server is deployed in a constrained device, we should consider the impacts of the R. phase on the system's availability. If it has a significant effect on the system's performance, the attacker could decide to stay in the R. phase and achieve his purpose rather than trying to reach the E. phase.

B. Overall architecture

As depicted in Fig. 1, our solution comprises an IDS and an MTD as the two main components. The MTD aims to move a parameter that might be an attack target. The MTD decisions are based on time (the System's Internal State component implementing the IANVS operations exposed above) or based on events (e.g., IDS alerts). The MTD-IDS interaction operates as follows: On the one hand, IDS detects abnormal activities and alerts the MTD which, in turn, moves the MP. On the other hand, after moving, the interaction between the attacker and the system is disrupted and the IDS observes and gains knowledge from this change.

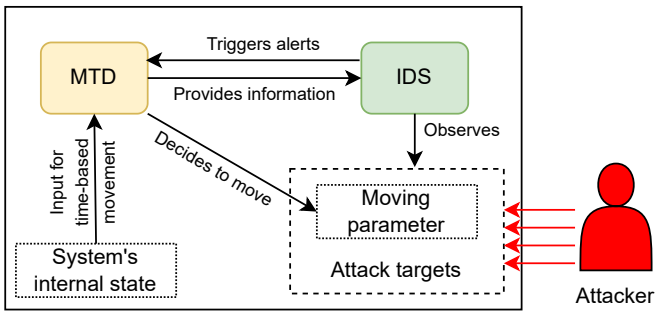


Fig. 1: Overall architecture of our proposal

1) MTD component: We propose a reactive MTD strategy that changes the MP's value based on two factors: time, and abnormal events. Fig. 2 illustrates the different situations our MTD scheme can handle. To understand how it operates, the following definitions are required:

- *Universe*: An abstraction of an input (u) to calculate the value of the MP. All universes have the same notion of time (t). A MP $p(u,t)$ is calculated by a function of u and t. At a given time t_1, the MP would have one value in one universe but different in another.
- *System's presence*: At a given time t, the MP uses only one universe as input. In other words, at any time, there exists only one active universe for the MTD system.
- *Jump*: Refers to a movement of the MP.
- *Normal behavior*: The behavior and interactions between legitimate components of the system.
- *Abnormal behavior*: The behavior or interaction that is diagnosed as an intrusion action.

In a normal state, the system stays in the main universe u_0. At the end of each MTD period, the MP will move according to IANVS [9]. IANVS uses *wall-clock* time as an input to calculate a new value of the MP. We call this movement a *T-move* (Time-based move). When the system detects an anomaly, IANVS changes the MP's value using a new universe u_1. We call this action an *E-move* (Event-based move). When an MTD period ends, the system always moves back to the main universe u_0. One important design decision of our proposal is that the T-move forces the system, and consequently the MP, to move back to the main universe. The purpose of this action is to help a legitimate user of the system to know what time t and universe u use as inputs to calculate the current MP's value, without additional universe synchronization overhead.

2) IDS component: Given the RoQ attack we aim to detect, we consider the number of incoming packets per second as the main metric of our detection mechanism. A legitimate client will send requests to a server to get resource information (e.g., temperature). The server defines a threshold value for the total number of requests received in one second. If the server receives more packets than the threshold, it reacts by informing the MTD component. We use this computationally inexpensive threshold-based detection mechanism to cope with the constraints of our IoT in-node IDS setting. Similar methods like the CUSUM statistical change detection method [12] are widely used and recognized in the literature.

Algorithm 1 MTD-based algorithm for intrusion detection

```
while True do
    if system.getTime == MTD.next_period then
        universe_id ← 0
        MTD.move(universe_id)
    end if
    if IDS.observe == "suspected activity" then
        universe_id ← universe_id + 1
        MTD.move(universe_id)
        if IDS.observe == "No more suspected activity"
then
            IDS.alert == "Attack"
        else
            IDS.alert == "No Attack"
        end if
    end if
end while
```

Algorithm 1 describes the details of this operation. One can see that the key point of our detection method consists of monitoring the new rate of incoming packets on a given IoT server port, once a movement is performed given the detection threshold crossing due to a formerly too-high rate of packets in the previous universe. Indeed, thanks to the IANVS substrate, a legitimate client leverages a method to find the new open port within a small amount of time (using the Auth-SYNC and the new universe identity). Thus, after an E-move due to a threshold crossing, the rate of incoming packets on the novel IoT server port value will only be that of the legitimate client. If the rate is still over the detection threshold, the latter must be updated to reflect this legitimate activity. On the contrary, the attacker needs non-negligible time to perform a new R. attack to find the MP's new value. As such, the receiving packet rate

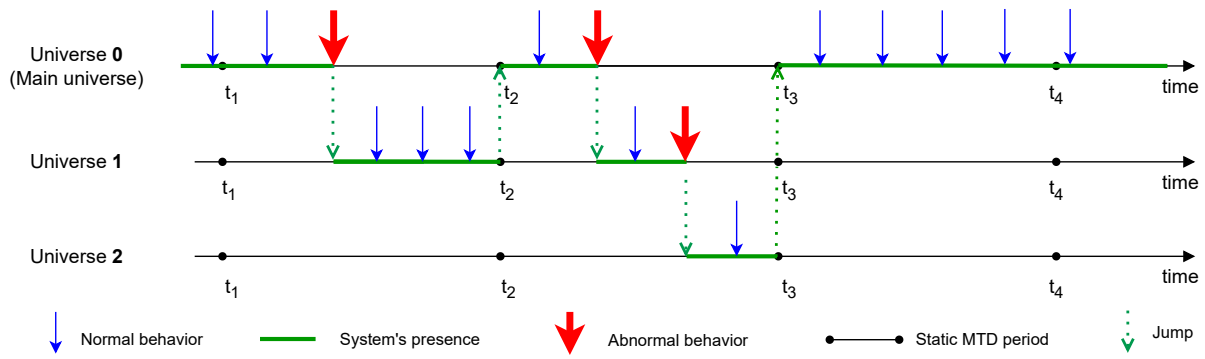


Fig. 2: Strategy of our hybrid-based MTD

would be reduced significantly right after the MP movement since only preserving the legitimate packet rate. Consequently, the abnormal behavior is considered finished if the new packet rate right after the E-move is much smaller than the last one, right before this movement.

V. Evaluation

The IANVS framework was previously implemented and proven effective to mitigate some attacks when proactive MTD is used. In this section, we extend this framework and implement our proposal to conduct dedicated evaluations assessing the trade-off of MTD settings as well as validating the benefit of reactive MTD in a resource-constrained IoT system. Our source code can be found online[1].

A. Experimental Setup and Scenario

1) Setup: We use the setup from [9]. Three entities are involved: (1) A *LoPy4 IoT node* running a Constrained Application Protocol (CoAP) Server. The LoPy4 does not have the capabilities to deploy advanced networking components such as a firewall or to adapt IDS solutions like Bro, Snort, or Suricata. (2) A *CoAP Client* deployed in a Dell desktop with an Intel Core i5-3470 4xCPU @ 3.20GHz, running Ubuntu 20.04.4. (3) An *Internal Attacker* deployed in an MSI laptop with an Intel Core i5-10210U 8xCPU @ 1.6GHz, running Ubuntu 20.04.4. All entities are connected to a LAN using a Linksys E900 Router. The LoPy4 uses a WiFi interface, while the rest Ethernet.

2) Relevant parameters: The client sends CoAP GET requests periodically to the server in order to get temperature information. The server's UDP port number is the MP that can be calculated by the client and server. We use 10 bits for the port value range. IANVS uses a secret pre-shared key stored on both client and server. Both sides use the Network Time Protocol to synchronize the internal system's clock. Both client and attacker send standard CoAP GET requests so the system can not distinguish them based on the packet content.

[1]https://github.com/Tien-V-Nguyen/MTDIDS2023/

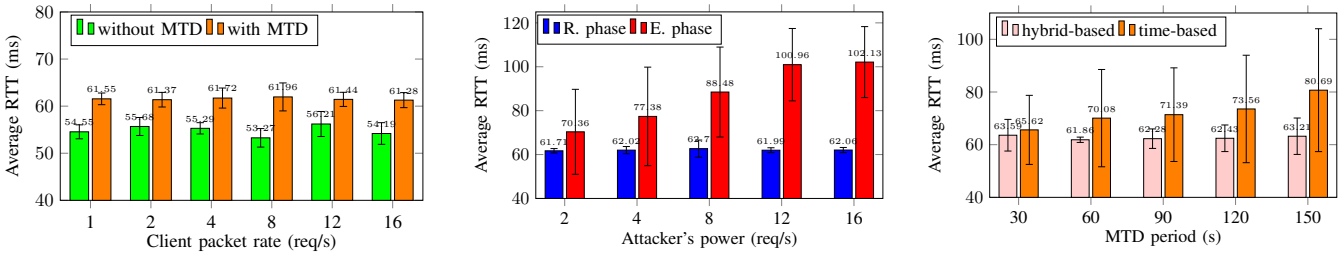
3) Metrics: M. Ashouri et al. [13] summarized the most relevant metrics used in the IoT literature for performance evaluation purposes. Among them, energy/power consumption, CPU utilization, and Round-Trip Time (RTT) are the most acknowledged. In our use case, we prioritize the networking component of the system. On the one hand, the average RTT of the client's requests would be affected significantly when the attacker tries to flood messages to the target port and causes an increase in the server's load in handling incoming requests resulting in a delay of the server processing time. On the other hand, the switching time when changing the port number induces a small period of no processing for incoming packets and it may increase the Packet Loss Rate (PLR) of client's requests. Therefore, we select the average RTT and the PLR of legit clients as the metrics to evaluate the performance of our MTD coupled with IDS proposal.

B. Results

This section presents the results of five experiments. Each experiment is composed of several configurations. We ran each configuration at least two times for ten or more minutes. In the first experiment, we assess the efficiency of the proposed IDS component in simple use cases. In the remaining four, we focus on the MTD evaluation and measure the RTT and PLR of legitimate clients' requests in different scenarios.

1) ***Experiment 1****. Detection performance:* In this evaluation, we assess the performance of the proposed IDS mechanism. The attacker's packet rates are fixed at 6 req/s in R. phase and 16 req/s in E. phase. MTD-movement is hybrid and the MTD period length is 60 seconds. We define three dynamic client use cases:

1) *Slight rise*: Client starts sending packets at a rate of 0.5 req/s and doubles the rate every 30 seconds. If the rate exceeds 2 req/s, it goes back to 0.5 req/s.
2) *Intensive rise*: Client starts sending packets at a rate of 0.5 req/s and doubles the rate every 30 seconds. If the rate exceeds 8 req/s, it goes back to 0.5 req/s.
3) *Random*: Client starts sending packets at a rate of 0.5 req/s and randomly chooses a new rate every 30 seconds. The domain of packet rate $\in \{0.5, 1, 2, 4, 8\}$.

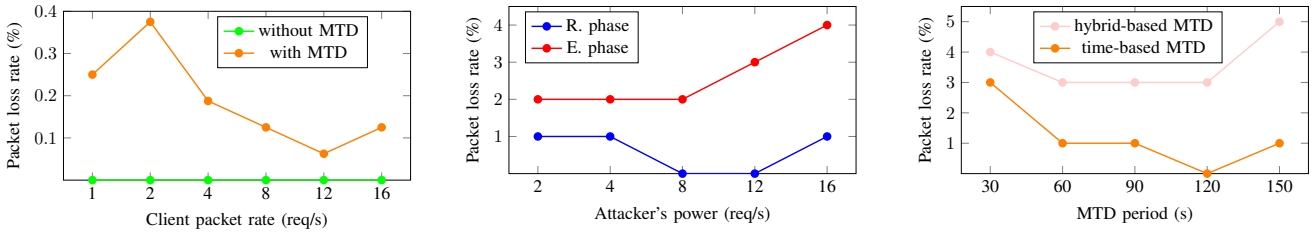


(a) Exp. 2: Client's RTT of systems with and without MTD (No attack)

(b) Exp. 3: Client's RTT under two attack phases (No MTD)

(c) Exp. 4: Client's RTT for hybrid- and time-based MTD (Under attack)

Fig. 3: RTT empirical results for Experiments 2-4



(a) Exp. 2: Client's PLR of systems with and without MTD (No attack)

(b) Exp. 3: Client's PLR under two attack phases (No MTD)

(c) Exp. 4: Client's PLR for hybrid- and time-based MTD (Under attack)

Fig. 4: PLR empirical results for Experiments 2-4

For each use case, we calculate the True Positive Rate (TPR) and the False Positive Rate (TNR) of the IDS as:

$$TPR = \frac{TP}{TP+FN} \; ; \; FPR = \frac{FP}{TN+FP}$$

Where: TP (True Positive) is the detection outcome where the IDS correctly predicts the real attack; FP (False Positive) is the outcome where the IDS predicts a normal behavior as an attack; TN (True Negative) is the outcome where the IDS correctly predicts the legitimate behavior; and FN (False Negative) is the outcome where the IDS predicts an attack as normal behavior. When the client raises its packet rate or when the exploitation attack comes, the system performs an E-move. After this move, the IDS component observes and detects the anomalies. The results are presented in Table I.

TABLE I: Exp. 1. Performance of the proposed IDS for three dynamic client cases

Case	Number of E-moves	Number of real attacks	TP	TN	FP	FN	TPR	FPR
Slight rise	20	11	11	9	0	0	100%	0%
Intensive rise	18	10	10	8	0	0	100%	0%
Random	13	9	9	4	0	0	100%	0%

The results are perfectly accurate (zero FP nor FN). This good performance allows us to consider further experiments in the next parts of the paper. However, the use cases in this first experiment are not diverse. A more extensive campaign should be done to have stronger statistical guarantees.

*2) **Experiment 2**. MTD Implementation's Cost:* In our use case, the CoAP service is initialized and bound to a specific port number which cannot be dynamically changed during the server's execution. To overcome this limitation, our server-side code creates a port redirection between the dynamic MTD port and the static CoAP port. This internal redirection can have side effects on the packet delivery delay which needs to be assessed. As such, we vary the client's packet rate to measure the impact of the client's speed on both MTD and non-MTD servers. For each client's rate value, we conduct two tests of 10 minutes: one for a server without MTD and the other for a server with MTD. The MTD period length is 60 seconds.

The results are shown in Fig. 3(a) and Fig. 4(a). A change in the client's speed rate affects neither RTT nor PLR significantly. In a non-MTD IoT, the average RTT is around 55ms while the use of MTD raises the RTT by about 10%. For the PLR results, we believe the T-move creates a short period of time where the server side drops packets.

*3) **Experiment 3**. Impacts of Reconnaissance and Exploitation Phases:* In this experiment, we evaluate the impact of each attack phase independently on the RTT and PLR. The R. phase is conducted by an attacker who sends scanning packets to a close port and the E. phase is instantiated by an attacker who continuously sends valid CoAP packets to the open port. The client rate is 3 req/s and the MTD period length is 60 seconds.

The results are shown in Fig. 3(b) and Fig. 4(b). The impact of the R. phase is negligible: the average RTT of client requests under the R. phase remains similar to the system without any attack (Fig. 3(a)). The E. phase, however, makes noticeable differences in both RTT and PLR on the client side.

When the attacker's speed increases, the QoS of our system is affected significantly and the server is going to crash several times when the flooding speed exceeds 16 req/s. This result motivates defenders to prevent attackers from reaching the E. phase.

4) ***Experiment 4****. Improvement and Costs of the Proposed Reactive MTD:* In this section, we evaluate the RTT and PLR of our novel hybrid-based MTD-IDS strategy vs a time-based-only MTD. The experiment is conducted with a constant client packet rate of 3 req/s, the attacker rate is 6 req/s in the R. phase and 16 req/s in the E. phase. Both schemes, time-based with static periodic MTD and hybrid-based (event- and time-based) MTD, are assessed for different lengths of the MTD period $\in \{30, 60, 90, 120, 150\}$. Each test lasts 10 minutes.

The results are shown in Fig. 3(c) and Fig. 4(c). The hybrid-based MTD brings a strong benefit since it ensures that the average RTT is always stable and close to the RTT of the system without attacks (Fig. 3(a)). Nevertheless, the trade-off of this hybrid-based MTD lies in the increase of the PLR when compared to the time-based MTD. Once the IDS detects an anomaly, the MTD performs an E-move and the client loses at least the next request.

5) ***Experiment 5****. Variable Client Traffic Use Cases:* In previous sections, the scenarios consisted of an attacker and a static client who always sends packets at a constant rate. We now evaluate our proposal with dynamic clients who change the packet rate over time with the same parameters and for the same three dynamic client uses case as presented in Experiment 1.

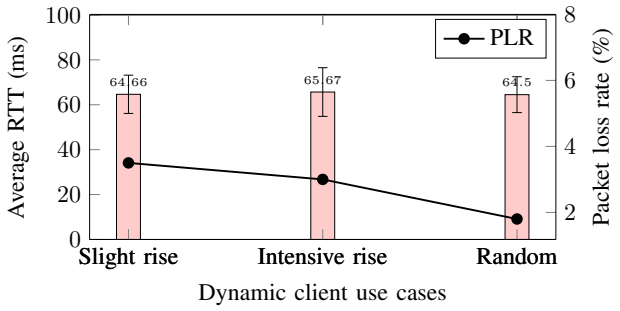


Fig. 5: Exp. 5. Performance of the hybrid scheme with three dynamic client use cases (Under attack).

The results are shown in Fig. 5. We can see that the average RTT remains stable but its standard deviation is larger than in the static client scenarios (Fig.3(c)). The PLR varies around 2-3 %, which is not significantly different from the static use case (Fig.4(c)).

VI. Conclusion

In this work, we proposed a security solution for constrained IoT systems by coupling MTD and IDS approaches which need lightweight adaptations in the context of resource-constrained IoT nodes. Moreover, we used MTD and IDS in synergy: each enhances and relies on the other. The MTD component is hybrid, exhibiting both reactive and proactive movement, a quality under-explored in the literature. We conducted several empirical assessments to measure the trade-offs of different MTD settings in terms of RTT and PLR. The results validate the improvement of our proposal in terms of RTT and the accuracy of our in-node IDS method. The main disadvantage of our hybrid MTD strategy is the slight increment in PLR compared to a time-based MTD.

In future work, Moving Parameters from non-network layers can be taken into account like data representation, software, or runtime environment. The evaluation campaign can take into account other metrics like CPU usage, RAM, or energy consumption. Also, the proposed in-node IDS scheme could be validated in more realistic scenarios (i.e., a real application) to obtain more robust statistical guarantees.

Acknowledgment

This work is partially supported by ICO (Institut Cybersécurité Occitanie), funded by Région Occitanie, France.

References

[1] I. Statista, "Internet of things (iot) connected devices installed base worldwide from 2015 to 2025 (in billions)," *URL: https://www. statista. com/statistics/471264/iot-number-of-connecteddevicesworldwide/(Consulté 17/05/2020)*, 2018.

[2] F. Chong, R. Lee, A. Acquisti, W. Horne, C. Palmer, A. Ghosh, D. Pendarakis, W. Sanders, E. Fleischman, H. Teufel III *et al.*, "National cyber leap year summit 2009: Co-chairs' report," *NITRD Program*, 2009.

[3] R. E. Navas, F. Cuppens, N. B. Cuppens, L. Toutain, and G. Z. Papadopoulos, "Mtd, where art thou? a systematic review of moving target defense techniques for iot," *IEEE internet of things journal*, vol. 8, no. 10, pp. 7818–7832, 2020.

[4] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of ids for iot: recent updates, security issues, and challenges," *Archives of Computational Methods in Engineering*, 2021.

[5] J. Giraldo, A. Cardenas, and R. G. Sanfelice, "A moving target defense to detect stealthy attacks in cyber-physical systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 391–396.

[6] Y.-Y. Chen, I.-H. Liu, C.-C. Wu, C.-G. Liu, and J.-S. Li, "Dynamic interval strategy for mt6d in iot systems," in *2021 International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)*. IEEE, 2021, pp. 36–39.

[7] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and M. Guizani, "A survey on iot intrusion detection: Federated learning, game theory, social psychology, and explainable ai as future directions," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4059–4092, 2022.

[8] 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.

[9] R. E. Navas, H. Sandaker *et al.*, "Ianvs: A moving target defense framework for a resilient internet of things," in *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020, pp. 1–6.

[10] R. E. Navas, F. Cuppens, N. B. Cuppens, L. Toutain, and G. Z. Papadopoulos, "Physical resilience to insider attacks in iot networks: Independent cryptographically secure sequences for dsss anti-jamming," *Computer Networks*, vol. 187, p. 107751, 2021.

[11] M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the transients of adaptation for roq attacks on internet resources," in *Proceedings of the 12th IEEE International Conference on Network Protocols, 2004. ICNP 2004.* IEEE, 2004, pp. 184–195.

[12] E. S. PAGE, "CONTINUOUS INSPECTION SCHEMES," *Biometrika*, vol. 41, no. 1-2, pp. 100–115, 06 1954. [Online]. Available: https://doi.org/10.1093/biomet/41.1-2.100

[13] M. Ashouri, F. Lorig, P. Davidsson, and R. Spalazzese, "Edge computing simulators for iot system design: An analysis of qualities and metrics," *Future Internet*, vol. 11, no. 11, p. 235, 2019.