Securing IoT Networks

Student Andrei Vlădescu
Professor Dr. Ion Bica
Faculty of Automatic Control and Computers
University POLITEHNICA of Bucharest
vladescu.andrei@techbattlefield.ro

January 30, 2024

Abstract

This study investigates the efficacy of Moving Target Defense (MTD) strategies in bolstering the security of Internet of Things (IoT) networks. With the escalating integration of diverse and resource-constrained IoT devices into critical infrastructures, traditional security measures prove insufficient against evolving threats. Our research assesses MTD techniques, such as dynamic network reconfiguration and software diversity, demonstrating their effectiveness in enhancing resistance to both known and zero-day attacks. The adaptive nature of MTD mechanisms ensures continuous adjustments, providing robust protection for systems, such as IoT devices. These findings provide insights for developers, security engineers, and researchers in search of adaptive solutions to safeguard IoT ecosystems from the ever-changing threat landscape.

1 Introduction

The surge in Internet of Things (IoT) devices has transformed our daily lives, linking everything from home gadgets to industrial equipment. According to [15], the number of these IoT devices will almost double from 2023 to 2030, to a number of aproximately 30 billion devices. While this connectivity brings convenience, it also raises important security concerns. IoT devices, now included inside many homes and facilities, require a focus on security to address potential risks. This paper aims to explore the security aspects of IoT devices, emphasizing the need for practical measures. One such important security measure is called Moving Target Defense (MTD).

MTD methodology typically falls under the category of proactive security measures, as it aims to evade security threats, thus making it more difficult for an attacker to deliver a malicious payload.

The objective of this paper is to provide a overview of the existing body of knowledge. By scrutinizing contemporary literature, research papers, and case studies, this document aims to focus on the current state of MTD in IoT, elucidating the key principles, challenges, and potential strongpoints for future exploration.

2 MTD Strategy applied in IoT

MTD can best be described as the classical game of "shell game" [4], in which a stone is hidden under three cups or shells, and the gambler tries to guess the location of the stone, after they were shuffled.

In the words of [11], moving target defenses have been proposed as a way to make it much more difficult for an attacker to exploit a vulnerable system by changing aspects of that system to present attackers with a varying attack surface. The goal of a diverse defense is to make the attack target unpredictable, making it more difficult to deliver a malicious packet. Such strategies are already in place in a computer, such as address space layout randomization (ASLR), instruction set randomization (ISR), honeypots (decoy nodes) or honeynets (decoy networks). As such, different parameters can be varied so that the attacker will either miss his target, or worse, hit a decoy target.

In the space of IoT, the targets are usually small nodes, with a low power consumption, sometimes in hard-to-reach locations. Some of these attacks, that aim to bring down nodes will catastrophically affect these devices, as they may be either something trivial as a light bulb, or something that belongs to a city's infrastructure [2]. These MTD methodologies must be effective, in relation to a certain attack, must not add much overhead, as IoT devices are computationally slow, and they will possibly consume more power only by adding a framework specifically for this defense mechanism.

3 State-of-the-Art MTD Techniques

In the works of [16] it is argued that MTD can be classified into four big categories. This is not the only article that tackles MTD categorization, others have done so too, in [14] or [10], but [16] draws a better distinction between MTD strategies.

3.1 Software-based Diversification

The first technique which is discussed is software-based diversification. Existing works achieve this technique by manipulating the programs or compilers to produce diveresification. Via software manipulation, the security can be enhanced by input rectification, since excision may limit the payload type. An implementation is SOAP [6] a software for rectifying input based on constraints.

Compiler-generated diversity attains it's purpose by producing internally different program variants, but with the same functions. Such works are presented by [11], in Chapter 4: massive-scale software diversity (MSSD) and malt-variant execution environment (MVEE).

3.2 Runtime-based Diversification

Several defense techniques address attacks by introducing diversification into runtime environments. One such technique is address space layout randomization (ASLR) [1]. ASLR randomizes the processes' address space, so that the attacker cannot use hard-coded addresses of a reconnoitered function or variable. Another such defense mechanism is instruction set randomization (ISR) [5]. ISR encodes machine code with a compile-time randomized key. At runtime, the instructions are deobfuscated, so that the program can go freely. As an attacker, hardcoding a machine-code instruction will lead to the illegal instruction error, and the program will terminate, thus protecting it.

3.3 Communication Diversification

Communication diversification techniques safeguard systems from network-related attacks by concealing internal information and communication protocols. One such implementation of this scheme is the mutable networks (MUTE) architecture, as it's presented in chapter nine of [11]. MUTE enables networks to change their configurations such as IP address and routes randomly and dynamically while preserving the requirements and integrity of network operation. Adding decoys, the attacker can be tricked into fingerprinting the nodes on the network with wrong information.

3.4 Dynamic Platform Techniques

Dynamic platform techniques (DPT)[10] change platform properties to stop attacking processes, like temporal changes (virtual machine rotation) or diversity (multiple variants of execution). Intersecting dynamic techniques at compiler-level are not discussed in this subsection, as they have been already covered. Talent [9] is a migration-based technique that leverages OS-level virtualization to create a virtual execution envirnoment for migrating a running application across different platoforms and preserving the state of the execution. DPT implementations can also be achieved by switching between different types of servers (e.g. Flask or Apache), as proposed in [13].

3.5 Security Model

The security model of identify, protect, detect, respond and recover (IPDRR) [8] is a traditional security model, that consists of standards, guidelinesand best practices.

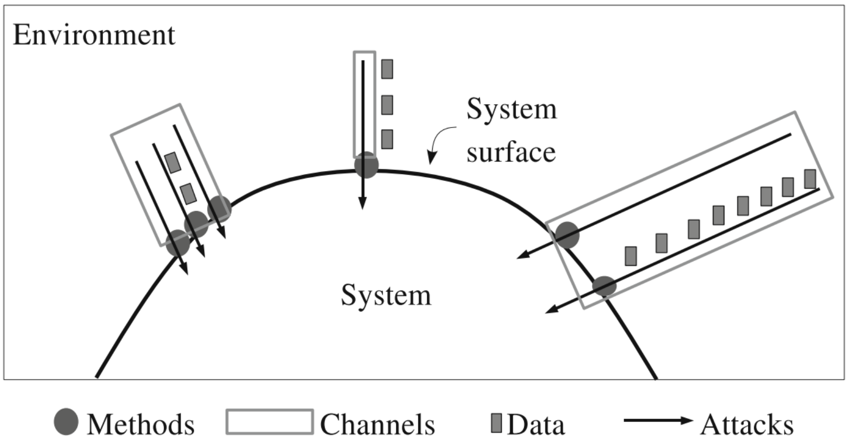

Figure 1: System's Attack Surface[11]

While the framework may be used in MTD, we can derive a better model for this usecase. New security models have been developed and measured [3] [12] [7]. To have a better grasp of the problem encountered, an attack surface definition is in due. A system's attack surface is the subset of the system's resources that an attacker can use to attack the system. The taxonomy of the attack surface may be categorised into more smaller elements.

3.5.1 Entry Points

A system's entry points are methods, such as the system's API, that receive data items from the environment. For instance, a direct entry point, like a method receiving input from a user or reading a configuration file, can be invoked by a user or system in the environment, read from a persistent data item, or invoke another system's API. Indirect entry points receive data from direct entry points.

3.5.2 Exit Points

The system's exit points are methods, such as the one writing to a log file, that send data items to the environment. A direct exit point, like a method invoked by a user or system receiving data results, can write to a persistent data item or invoke another system's API with data items as input. An indirect exit point sends data to a direct exit point.

3.5.3 Channels

Systems have communication channels, like TCP/UDP sockets or RPC end points, which users or external systems use to interact. Attackers exploit these channels to connect to the system and invoke methods, adding another potential avenue for attacks.

3.5.4 Untrusted Data Items

An attacker utilizes persistent data items to either indirectly inject data into the system or receive data indirectly from it. Examples of persistent data items include files, cookies, database records, and registry entries. For instance, a system may read from a file after an attacker writes into it, or vice versa. Consequently, persistent data items serve as another vulnerability for potential attacks on a system.

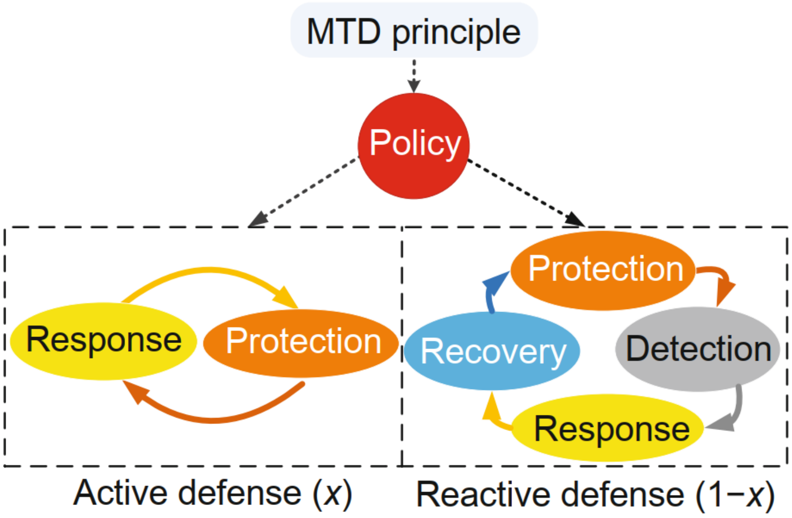

Figure 2: New Security Model for MTD[3]

A new security model encompasses both active defense and reactive defense processes. In the active mode, defense operates independently of network status, periodically or irregularly shifting the attack surface. As a result, detection and recovery links are not required. In the reactive mode, the defense process is initiated by security alerts, aligning with the IPDRR model. x is the measurement of active defenses, and 1-x is the reactive defense measurement. A defense mechanism may employ a security system with different x, thus changing the dynamic.

4 Conclusion and Further Work

In conclusion, this review highlights the critical significance of moving target defense in reinforcing the security posture of IoT devices. This examination of existing literature, case studies, and research findings aims to find a good balance for the problematic landscape of IoT security. By implementing these measures, the defense mechanisms act as a robust deterrent against malicious activities targeting the interconnected nature of IoT devices. Looking ahead, the identified gaps and limitations in current MTD strategies offer valuable insights for focused research initiatives. Given the dynamic and evolving nature of both IoT and cybersecurity, ongoing refinement of new security solutions for IoT is imperative.

References

[1] Pax project aslr documentation. https://pax.grsecurity.net/docs/aslr.txt. Original ASLR implementation.

[2] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K.-C. Wang. Review of internet of things (iot) in electric power and energy systems. *IEEE Internet of Things Journal*, 5(2):847–870, 2018.

[3] G.-l. Cai, B.-s. Wang, W. Hu, and T.-z. Wang. Moving target defense: state of the art and characteristics. *Frontiers of Information Technology Electronic Engineering*, 17:1122–1153, 11 2016.

[4] 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, 2019.

[5] J. Kennan and V. P. Kemerlis. Instruction set randomization: An updated implementation. 2020.

[6] F. Long, V. Ganesh, M. Carbin, S. Sidiroglou, and M. Rinard. Automatic input rectification. In *2012 34th International Conference on Software Engineering (ICSE)*, pages 80–90, 2012.

[7] P. K. Manadhata and J. M. Wing. An attack surface metric. *IEEE Transactions on Software Engineering*, 37(3):371–386, 2011.

[8] N. I. of Standards and Technology. Framework for improving critical infrastructure cybersecurity, 2021.

[9] H. Okhravi, A. Comella, E. Robinson, S. Yannalfo, P. Michaleas, and J. Haines. Creating a cyber moving target for critical infrastructure applications. In J. Butts and S. Shenoi, editors, *Critical Infrastructure Protection V*, pages 107–123, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[10] H. Okhravi, T. Hobson, D. Bigelow, and W. Streilein. Finding focus in the blur of moving-target techniques. *IEEE Security Privacy*, 12(2):16–26, 2014.

[11] S. J. A. K. G. V. S. C. W. X. S. W. e. Pratyusa K. Manadhata, Jeannette M. Wing (auth.). Moving target defense: Creating asymmetric uncertainty for cyber threats. Advances in Information Security №54. Springer, 2011.

[12] A. K. G. V. S. V. S. C. W. X. S. W. e. Pratyusa K. Manadhata (auth.), Sushil Jajodia. *Moving Target Defense II: Application of Game Theory and Adversarial Modeling*. Advances in Information Security 100. Springer-Verlag New York, 1 edition, 2013.

[13] A. Saidane, V. Nicomette, and Y. Deswarte. The design of a generic intrusion-tolerant architecture for web servers. *IEEE Transactions on Dependable and Secure Computing*, 6(1):45–58, 2009.

[14] N. Saputro, S. Tonyali, A. Aydeger, K. Akkaya, M. A. Rahman, and S. Uluagac. *A Review of Moving Target Defense Mechanisms for Internet of Things Applications*, pages 563–614. 2020.

[15] L. S. Vailshery. Number of iot connected devices worldwide 2019-2023, with forecasts to 2030. 2023. Accessed: January 2024.

[16] J. Xu, P. Guo, M. Zhao, R. Erbacher, M. Zhu, and P. Liu. Comparing different moving target defense techniques. *Proceedings of the ACM Conference on Computer and Communications Security*, 2014:97–107, 11 2014.