IoT Devices: “Smart” Security Testing

Levi Seibert

Abstract

IoT (or smart) devices are physical devices that are connected to the Internet. They are increasingly common in today’s technological world, and as such, have become a major risk in terms of cybersecurity. Much research has been done in the field of IoT attacks and testing; however, much of the testing does not focus on the security of the devices, but rather their functionality. Therefore, many of these devices are still very vulnerable. Once the functionality of these devices is better understood, some of the existing vulnerabilities become more obvious. The only real way to protect these devices from the known vulnerabilities is by integrating (or improving) security testing during the development of the systems.

Introduction

IoT stands for the “Internet of Things”. The Internet of Things is a network of Internet-connected physical devices, called “things”; basically, it consists of everyday tools that have been reimagined with Internet technologies [1]. Examples of IoT devices include smart speakers, smart appliances, smart thermostats, pretty much “smart’ anything.

IoT devices are commonplace in today’s technological world. According to some estimates, there are between 12.2 and 14.4 billion active IoT devices across the globe. By 2025, that number is expected to increase to ~27 billion [2]. Since all these devices are, by the nature of IoT, connected to the Internet, they are at high risk of cyberattacks. As such, the devices need to be better tested from a security standpoint.

This paper explores some of the vulnerabilities that are existent in IoT or “smart” devices and how to mitigate exploitations. In the next section, some related research is discussed. Next, an analysis of IoT risks as well as a threat model are presented. The following section considers some specific IoT attacks and vulnerabilities. Then, the solution (integrating secure testing) is examined. Finally, the conclusion will summarize the topics that have been considered.

Background

As expected, the prevalence of IoT devices has led to a plethora of potential attacks and vulnerabilities. As such, many research teams have investigated some of these attacks and how they can be exploited against smart devices.

Acar, Huang, Li, Narayana, and Feamster presented two web based IoT attacks that target IoT devices with HTTP servers on their Local Area Networks; namely, exploiting HTML5 MediaError error messages in Chrome and Firefox and using DNS rebinding to access local devices [3]. Many IoT devices have HTTP servers as part of their functionality, and therefore are at risk of this vulnerability.

Luo, Zhao, Lu, Sagduyu, and Xu proposed an attack on IoT devices by making use of machine learning in an adversarial manner. The machine learning model is created in such a way as to infer outcomes and manipulate outputs of devices, which has the potential of wrecking complete havoc to IoT systems [4].

Dang, Li, Liu, Zhai, Chen (Qi), Xu, Chen (Yan), and Yang presented their studies on the threat of fileless attacks (attacks that exploit vulnerabilities already present on a device) on Linux-based IoT devices. Fileless attacks have been on the rise, and IoT devices are often vulnerable because of their lack of sophisticated firewalls and antivirus tools [5].

Mitev, Miettinen, and Sadeghi presented a man-in-the-middle attack that targets smart device "Skills" (and their counterparts in other OSs). The goal of this attack is to combine malicious back-end Skill functionalities with inaudible attack techniques in order to allow an adversary to control and hijack communications between the user and the Skills on an IoT smart device, thus allowing for all manners of maliciousness [6].

Sugaware, Cyr, Rampazzi, Genkin, and Fu presented an attack called LightCommands that injects commands to voice-controllable systems (like smart speakers) via a laser. By targeting the MEMS microphones in several of these systems, the researchers were able to inject commands into devices from a considerable distance and through glass windows, allowing for a semi-remote attacker to potentially execute heinous commands [7].

OConnor, Jessee, and Campos considered the security issues with IoT companion apps and how vulnerabilities in smart home devices can fall victim to man-in-the-middle attacks. Since IoT devices often communicate sensitive data with a mobile application, an attacker intercepting this traffic would have access to dangerous information and may be able to cause serious harm [8].

Obviously, the IoT is a complex umbrella of devices. Not all systems are the same, nor do they work in the exact same fashion (although they do share similar functionality, as described below). That being said, much research has been done in order to standardize the testing of these devices.

Behnke, Thamsen, and Kao presented an automated testing framework for IoT applications, known as Héctor. This framework is based on three guiding factors: flexible experiment definitions, extensible platform support, and arbitrary testbeds [9].

Truong, Berardinelli, Pavkovic, and Copil presented modeling techniques for Cyber-Physical Systems (CPS) and Internet of Thing (IoT) systems, as well methods of testing their uncertainties. Their process of testing is separated into four parts: modeling, extracting model information, generating test configurations, and deploying the system under test [10].

Dias, Ferreira, and Sousa discussed the Continuous Integration (CI)/Continuous Delivery (CD) of IoT devices. The research team developed three related strategies that help solve the problem of reproducible testing methodologies: using real device testbeds, using simulation-based testing, and updating software with a middleman [11].

Freitas and Lelli presented a mapping of existing research in regard to using Machine Learning (ML) algorithms to test IoT applications. It was discovered that Deep Learning was the most used technique, and most articles were focused on identifying security attacks [12].

Pontes, Lima, and Faria proposed a pattern-based IoT testing approach based on five test patterns that they identified as recuring throughout IoT systems. The five test patterns/strategies identified by the research group are testing periodic readings, testing triggered readings, testing alerts, testing action, and testing actuators [13].

While there has been much work done in testing the functionality IoT devices, it does appear that testing methods designed to assess the security of these systems is lacking, which is an unfortunate and dangerous situation, considering even just a fraction of the vulnerabilities that are currently existent (or in research).

Risk Analysis and Threat Model

While researchers have considered several attacks (and mitigations) against IoT devices, there are still numerous other ways that these computers can be exploited. By the nature of their

Internet-based functionality, they are at major risk of attacks. Because IoT is so broad, there are specific attacks that may affect some devices while not others and vice versa.

System functionality

As was previously mentioned, the field of IoT is very broad and diverse. There are many devices that do very different things; however, they all have the same basic flow of functionality. As shown in Figure 1, most IoT devices have a default “waiting” stage where they wait for a predetermined condition (a wake word or some other action). Once this event occurs, the device moves into a data-receiving state, in which input is received from either the user or the environment. After the input is completed, the information is processed, and the action is performed and/or a response is provided. Then, the output is displayed, or the environment is changed. Finally, the device returns to the waiting state. Once again, this is a very broad overview of the system functionality; however, it does summarize the general process of IoT devices.

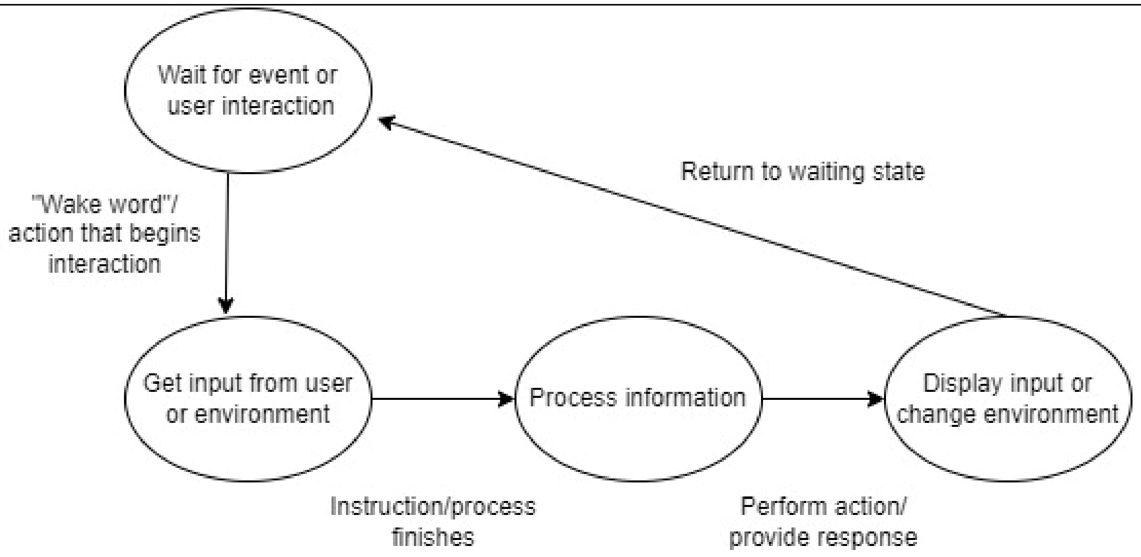
Wait for event or user interaction
"Wake word"/ action that begins interaction
Return to waiting state
Get input from user or environment
Process information
Display input or change environment
Instruction/process finishes
Perform action/ provide response

FIGURE 1 An example state transition diagram for IoT devices.

Threat Paths

When the functionality of IoT devices is further broken down, the flow of the data becomes a key factor to consider. It has been determined that most IoT devices have six processes that make up the entire flow of the data: Sense, Transport, Store, Analyze, Control, and Share [14]. This data flow is shown in Figure 2.

There are two potential external entities for an IoT device, a user and/or the environment. Either way, data is sensed from one of these entities and transported to some sort of storage resource (often a cloud device). From there, the data is analyzed, control decisions are made, and data is shared with the originator(s); that is, the user and/or the environment.

Of particular importance, especially when considering the potential threats with IoT devices, are the boundary points (indicated with dotted vertical lines in Figure 2). These are the points in the flow when the data moves between devices, and thus because extra vulnerable to attacks.

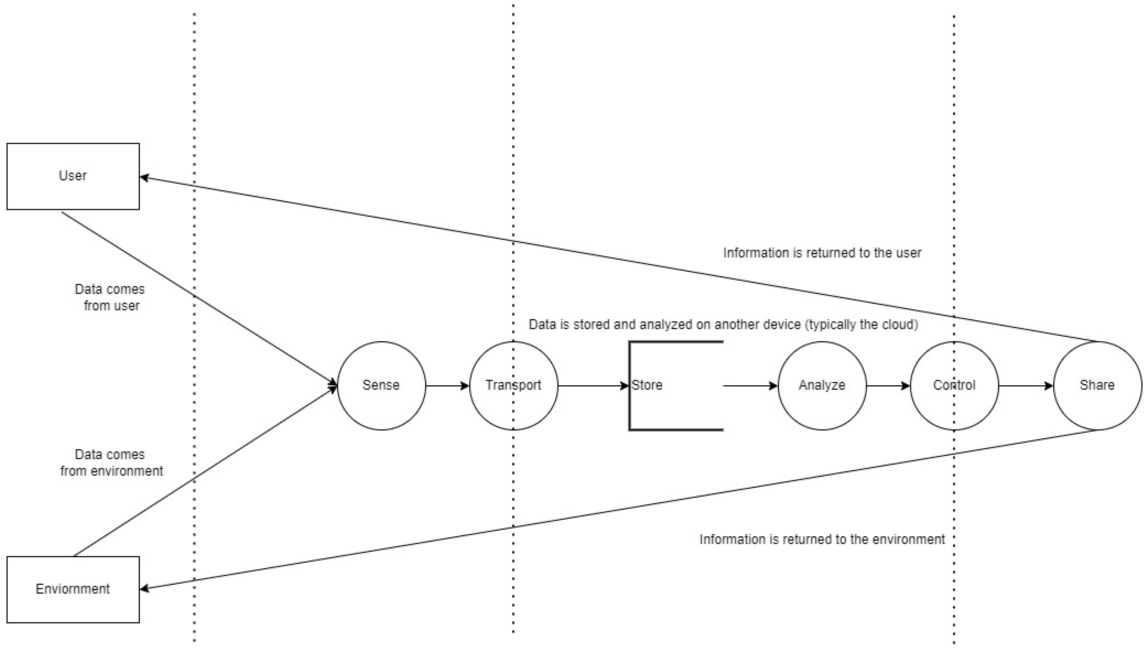
User
Data comes from user
Data comes from environment
Enviornment
Information is returned to the user
Data is stored and analyzed on another device (typically the cloud)
Sense
Transport
Store
Analyze
Control
Share
Information is returned to the environment

FIGURE 2 An example data flow diagram for IoT devices.

Vulnerabilities and Attacks

There are obviously numerous points of failure in smart devices. Several of them (fileless attacks, light commands, etc.) have already been discussed based on existing research that has been done. There are a few attacks that are more general and more common (note that these vulnerabilities are based off their CAPEC descriptions as defined by MITRE):

One of the most common attacks against IoT devices is that of trying common or default passwords; that is, brute-forcing logins based on easily guessed passwords [15]. Once an attacker gains access to a single device, they can wreak havoc across the entire network. This attack is so prevelant simply for the fact that smart devices are often very lacking in authentication policies. The Mirari botnet is a stellar example of this type of attack [16].

In an adversary-in-the-middle (also called man-in-the-middle) attack, the attacker sits between two connected devices. He/she intercepts all traffic between the two devices. The attack can be either passive (the attacker only observes the traffic) or active (the attacker alters the traffic) [17]. A couple of man-in-the-middle attack examples have already been discussed.

Flooding is another attack that can occur. In this case, the attacker will repeatedly send a large number of interactions to a host. This has the potential of shutting down the host from other connections [18]. Common examples of this are DoS's and DDoS's, where servers are shut down from a large number of requests from a client.

Finally, there is an attack that overflows device buffers. This occurs when there are no bounds checking and the attacker injects or manipulates code that overflows some buffer. The result of this attack may be crashing the device or allowing for modified execution [19]. An example of this is the Devil’s Ivey attack, which allowed for remote code execution in certain models of security cameras [20].

All of these attacks specifically target the IoT device itself, rather than the storage and analyzation systems. While these systems are also vulnerable, the primary focus of this paper is on the IoT systems themselves and the vulnerabilities that are relevant to them.

Integrating Secure Testing

In order to mitigate these vulnerabilities (and many more), IoT developers need to shift their focus from quantity to quality. The current popularity of these systems is very lucrative for developers; however, mass producing them without regard for security is a major concern. In fact, the exploitation of devices can have significant financial consequences, especially if the fault occurs because of a manufacturer defect.

In order to help prevent these situations, developers need to make security a priority. Manufacturers need to amend their Software Development Life Cycle in order to focus on producing safe products. This is known as implementing the Secure Software Development Cycle.

There are numerous ways to shift towards a security-focused development methodology but the most important task that must be focused on is testing. Several of the previously mentioned research articles focus on the testing of IoT devices, however most of that work focuses on testing the functionality of the products, rather than their defenses. There are many different methods of testing these devices (testbeds, simulations, etc.), but regardless of the methodology selected, testers need to keep security in the front of their minds.

Testers can make use of black-box (where they have no inside information on the device), white-box (where the tester is directly familiar with its inner-workings), or gray-box testing (a combination of white and black-box techniques). Each of these have advantages and disadvantages, but gray-box testing is often considered to be the best of both worlds.

Testers also need to be sure that they test the systems at various stages. Unit tests need to take place when the individual components are being produced (this testing is often done by the developers themselves). Once the individual pieces are all produced and put together, system, or end-to-end, testing needs to take place to ensure that the system works as is intended, without security flaws.

Every component of IoT devices needs to be tested; however, for some of the more common attacks, described earlier, a tester can easily check to see if the vulnerabilities are present. If they are, then there are fairly simple mitigations that can be implemented. For example, in order to prevent man-in-the-middle attacks, devices should ensure that entire communications are encrypted and that the participating devices are verified. To prevent buffer overflows, simple bounds checking can be added to the program. In order to guard against flooding attacks, developers should add throttling techniques. And, to address default username/password exploitations, devices need to be shipped with either unique and random login credentials, or mandate the credentials be changed after a short period of time.

At the very least, IoT manufacturers need to have a good patch management system in place, in order to efficiently and effectively strengthen the defense of vulnerable devices, especially after new faults are discovered.

Summary/Conclusion

IoT devices are overtaking the globe. The Internet-connectivity of all these devices lends itself to allowing for numerous vulnerabilities. In order to prevent these attacks, developers need to make security testing a priority in their development, rather than focusing on just mass-producing devices. After all, strengthening the security of IoT devices is the “smart” thing to do.

References

[1] Oracle, “What is IoT?,” *Oracle*. [Online]. Available: https://www.oracle.com/internet-of-things/what-is-iot/. [Accessed: August 26, 2022].

[2] M. Hasan, “State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally,” *IoT Analytics*, May 18, 2022. [Online]. Available; https://iot-analytics.com/number-connected-iot-devices/. [Accessed: August 26, 2022].

[3] G. Acar, D. Y. Huang, F. Li, A. Narayanan, and N. Feamster, “Web-based Attacks to Discover and Control Local IoT Devices,” Proceedings of the 2018 Workshop on IoT Security and Privacy (IoT S&P '18), pp. 29-35, August 2018. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/ 10.1145/3229565.3229568. [Accessed: Oct. 10, 2022].

[4] Z. Luo, S. Zhao, Z. Lu, Y. E. Sagduyu, and J. Xu, “Adversarial Machine Learning Based Partial-model Attack in IoT,” Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning (WiseML '20), pp 13-18, July 2020. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/ 10.1145/3395352.3402619. [Accessed: Oct. 10, 2022].

[5] F. Dang, Z. Li, Y. Liu, E. Zhai, Q. A. Chen, T. Xu, Y. Chen, and J. Yang, “Understanding Fileless Attacks on Linux-based IoT Devices with HoneyCloud,” Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '19), pp 482-493, June 2019. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/ 10.1145/3307334.3326083. [Accessed: Oct. 10, 2022].

[6] R. Mitev, M. Miettinen, and A. Sadeghi, “Alexa Lied to Me: Skill-based Man-in-the Middle Attacks on Virtual Assistants,” Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS '19), pp. 465-478, July 2019. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/10.1145/3321705.3329842. [Accessed: Oct. 10, 2022].

[7] T. Sugawara, B. Cyr, S. Rampazzi, D. Genkin, and K. Fu, “Light Commands: Laser-Based Audio Injection Attacks on Voice-Controlled Systems,” 29th USENIX Security Symposium, 2020.

[8] T. J. OConnor, D. Jessee, and D. Campos, “Through the Spyglass: Towards IoT Companion App Man-in-the-Middle Attacks,” Cyber Security Experimentation and Test Workshop (CSET '21), pp. 58-62, August 2021. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/10.1145/ 3474718.3474729. [Accessed: Oct. 10, 2022].

[9] I. Behnke, L. Thamsen, and O. Kao, “Héctor: A Framework for Testing IoT Applications Across Heterogeneous Edge and Cloud Testbeds,” In Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC '19 Companion), pp. 15-20, December 2019. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/10.1145/3368235.3368832. [Accessed: Oct. 31, 2022]

[10] H. Truong, L. Berardinelli, I. Pavkovic, and G. Copil, “Modeling and providing IoT Cloud Systems for Testing Uncertainties,” In Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2017), pp. 96-105, November 2017. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/10.1145/3144457.3144490. [Accessed: Oct. 31, 2022].

[11] J. P. Dias, H. S. Ferreira, and T. B. Sousa, “Testing and Deployment Patterns for the Internet-of-Things,” In Proceedings of the 24th European Conference on Pattern Languages of Programs (EuroPLop '19), Article 16, pp. 1-8, July 2019. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/ 10.1145/3361149.3361165. [Accessed: Oct. 31, 2022].

[12] L. Freitas and V. Lelli, “Using Machine Learning on Testing IoT Applications: a systematic mapping,” In Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia '22), pp. 348-358, November 2022. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/ 10.1145/3539637.3558049. [Accessed: Oct. 31, 2022].

[13] P. M. Pontes, B. Lima, and J. P. Faria, “Test Patterns for IoT,” In Proceedings of the 9th ACM SIGSOFT International Workshop on Automating TEST Case Design, Selection, and Evaluation (A-TEST 2018), pp. 63-66, November 2018. [Online serial]. Available: Association for Computing Machinery, https://doi-org.athens.idm.oclc.org/10.1145/ 3278186.3278196. [Accessed: Oct. 31, 2022].

[14] Breadware, “How Do IoT Devices Work?,” *Breadware*, Oct. 17, 2022. [Online]. Available: https://breadware.com/2022/10/how-do-iot-devices-work/. [Accessed Dec. 1, 2022].

[15] CAPEC, “CAPEC-70: Try Common or Default Usernames and Passwords,” *mitre.org*, Oct. 21, 2021. [Online]. Available: https://capec.mitre.org/data/definitions/70.html. [Accessed: Aug. 26, 2022].

[16] M. Kan, “IoT botnet highlights the dangers of default passwords,” *CSO*, Oct. 4, 2016. [Online]. Available; https://www.csoonline.com/article/3127263/iot-botnet-highlights-the-dangers-of-default-passwords.html. [Accessed: Aug. 30, 2022].

[17] CAPEC, “CAPEC-94: Adversary in the Middle (AiTM),” *mitre.org*, Oct. 21, 2021. [Online]. Available: https://capec.mitre.org/data/definitions/94.html. [Accessed: Aug. 26, 2022].

[18] CAPEC, “CAPEC-125: Flooding,” *mitre.org*, Oct. 21, 2021. [Online]. Available: https://capec.mitre.org/data/definitions/125.html. [Accessed: Aug. 26, 2022].

[19] CAPEC, “CAPEC-100: Overflow Buffers,” *mitre.org*, Oct. 21, 2021. [Online]. Available: https://capec.mitre.org/data/definitions/100.html. [Accessed: Aug. 26, 2022].

[20] R. Vamosi, “Devi’s Ivy security vulnerability leaves IoT devices at risk,” *Synopsys*, July 19, 2017. [Online]. Available: https://www.synopsys.com/blogs/software-security/devils-ivy/ [Accessed: Dec. 1, 2022].