Model-Enabled IoT Penetration Testing

Justin Wasser

University of Maryland Global Campus

INFA 670: Information Assurance Capstone

Professor Boyles

3/20/2024

Contents

Abstract

The Internet of Things (IoT), presents security challenges due to its assortment of devices and configurations. Moreover, the realm of IoT is expanding rapidly and novel IoT devices are commonplace. Furthermore, IoT devices are often connected to larger networks and therefore can serve as a gateway for malicious actors into said networks if not properly secured. Therefore, effective penetration testing of IoT devices is a crucial aspect of a comprehensive network security architecture. Lastly, some aspects of traditional penetration testing methods are not well-suited for securing IoT devices and networks, and therefore new approaches such as combining aspects of “Model-based Security Testing (MBST)” (Lonetti et al., 2023) with automated and manual penetration testing methods may be preferable.

Introduction

The “Internet of Things (IoT)” (Chougule, n.d.) can be simply defined as any non-traditional computing device that is connected to the Internet and leverages that connection to send and receive data (Chougule, n.d.). Almost any device you can think of today has the potential to become an IoT device simply by giving it the capability to connect to the internet (EC-Council, 2023). Moreover, the IoT’s architecture varies depending on the device in question, its specific application, and its environment, and therefore for this discussion, it is sufficient just to note that IoT devices rely on connections with various other computing entities (servers, other IoT, etc.) to function (Bella et al., 2023; OWASP, n.d.). Additionally, the IoT domain is relatively new compared to traditional computing infrastructures and it is also expanding rapidly as shown by its revenue increasing more than three times from 2013 to 2020 and with around 30 billion active IoT devices projected by 2030 (Chougule, n.d.; EC-Council, 2023).

Moreover, IoT devices include products that directly affect an individual's ability to maintain their privacy (IoT devices store and transmit personal data), safety (smart home/smart lock/smart car), or even their life (smart pacemaker) (Chen et al., 2018; Hardaway, 2023). Furthermore, these are just examples of personal IoT products, however, the same concerns apply on a grander scale when considering the security of IoT devices that are a part of industrial systems (Bella et al., 2023). To that point, due to their prevalence and the sensitive data/systems they interact with assuring the security of IoT devices is of the utmost importance (Chen et al., 2018). However, realizing this goal will require overcoming the various challenges IoT environments present (Chen et al., 2018).

IoT Environmental Challenges

For starters, the innovation that drives the proliferation of the Internet of Things (IoT) is increased connectivity, however, that innovation also creates more opportunities for vulnerabilities to exist among the increasing devices and connections (Chen et al., 2018). To that point, IoT devices incorporate a wider range of technologies and architectures compared to traditional computers and therefore it is harder to create a universal security architecture for IoT devices (Bella et al., 2023; EC-Council, 2023). Moreover, many of these devices' placement can be defined as simultaneously "easily accessible by an attacker" (Bures et al., n.d.) yet "difficult to check by the service provider" (Bures et al., n.d.). Moreover, these challenges are exacerbated by the business considerations that drive many IoT devices to be deployed rapidly (less testing) and at the lowest cost possible (i.e. less capable hardware), which further restricts the security capabilities of these devices (Bures et al., n.d.). These hardware constraints (processing and/or power related) make it difficult if not impossible to update a portion of said devices and "can lead to the implementation of lightweight authorization and security algorithms, exposing these IoT devices as a weak entry point to the whole network" (Bures et al., n.d.).

Furthermore, there is a greater range of communication protocols used across IoT devices which calls into question the security of "the lower physical layers (hardware, network protocols, operational systems, application servers etc.)" (Bures et al., n.d.), meaning that often for IoT devices/networks each of those lower layers must be tested to ensure that they are secure, which is both complex and expensive (Bures et al., n.d.). Lastly, IoT devices often do not rely on a secure framework from which to build and deploy devices, which increases the chance of vulnerabilities being present (Bures et al., n.d.; Lonetti et al., 2023). To that point, all these factors lead to the current situation where "57 percent of IoT devices are susceptible to medium- and high-severity exploits" (EC-Council, 2023). Therefore, preventing the occurrence of many

vulnerabilities in IoT devices is not always feasible given the current IoT environment, therefore the task of identifying and exploring the potential harm associated with various vulnerabilities, i.e. pen-testing, is paramount (Bures et al., n.d.; Lonetti et al., 2023).

Automated vs Human Penetration Testing

Penetration testing (pentesting) is a form of "security testing" (Lonetti et al., 2023) that seeks to mimic the attacks a system will likely face from malicious actors (Chen et al., 2018; Lonetti et al., 2023). Furthermore, the purpose of performing penetration testing is to identify vulnerabilities in a system/network and to explore how damaging exploitation of a given vulnerability may be to a system/network (EC-Council, 2023; Lonetti et al., 2023). Moreover, pentesting can be performed manually or it can be automated, and there are strengths and weaknesses associated with each approach (Johnson, 2022). To that point, the main benefit of automated pentesting is that it is much more efficient than pentesting performed by humans (Johnson, 2022; Pope, n.d.). Since automated pentesting can be performed significantly faster and at a lower cost compared to manual pentesting, it can be performed more often (Johnson, 2022). Moreover, the types of penetration tests that mesh well with automation currently are tests that do not require complex reasoning i.e. inferences, deductions, etc. (Chen et al., 2018; Johnson, 2022; OccamSec, 2023; Pope, n.d.).

Some aspects of testing an IoT device's interface, system, and network fall under this definition as they could also be tested using static analysis, "semi-formal" (US Department of Homeland Security, 2006), and/or "formal methods" (US Department of Homeland Security, 2006) (Chen et al., 2018; Lonetti et al., 2023; US Department of Homeland Security, 2006). For example, "security researchers use automated tools to carry out three types of specialized PT: interface testing, transportation testing, and system testing" (Chen et al., 2018). To that point,

automated pentesting of interfaces can be used to check “input validation mechanisms” (Chen et al., 2018), which are the origin of some of the most prevalent and harmful vulnerabilities associated with IoT devices/networks such as ““insecure web interface” and “insecure network services,”” (Chen et al., 2018) (Chen et al., 2018). Moreover, automated pentesting is capable of testing many of the protocols used to relay information within the IoT environment (Chen et al., 2018). Automated testing of these protocols includes checking for “misuse issues and design flaws in communication protocols and weak cryptographic schemes” (Chen et al., 2018) (Chen et al., 2018). Additionally, an IoT device’s system can also be tested using automated tools that check for “implementation flaws, insecure system settings, and other known vulnerabilities” (Chen et al., 2018) (Chen et al., 2018). However, automated penetration testing does have its limitations (Johnson, 2022; OccamSec, 2023; Pope, n.d.).

Automated pentesting is a relatively new and still developing technology, and therefore at this stage, it is not well suited for use in all IoT security domains as “Pen tests on wireless networks, web apps and social engineering, for example, aren't supported by most tools” (Johnson, 2022). Therefore, performing manual pentesting is a critical part of securing IoT environments (Johnson, 2022; OccamSec, 2023). Moreover, besides not applying to all IoT testing domains, the main weakness of automated pentesting is its lack of ingenuity, which is a human penetration tester’s strength (Chougule, n.d.; Johnson, 2022). To that point, a human conducting penetration testing can engage in social engineering tactics to learn about or even access a given system (Johnson, 2022). They can also use their skill to “find cleverer vulnerabilities and attacks that automated tests may miss, such as blind SQL injection attacks, logic flaws and access control vulnerabilities” (Johnson, 2022). Therefore, the most effective penetration testing methodology is one that employs both automated and manual penetration

testing to complement one another, thereby creating a more layered security architecture (Chen et al., 2018; Johnson, 2022; OccamSec, 2023).

IoT Penetration Testing Methodology

A general methodology for penetration testing comes from NIST, and while it is not specifically designed for IoT environments, its basic outline of activities is still applicable (Scarfone et al., 2008). According to Scarfone et al. (2008), the four parts of penetration testing are planning, discovery, attack, and reporting. Moreover, like many aspects of penetration testing the planning segment becomes more complex when dealing with the IoT (OWASP, n.d.; Scarfone et al., 2008). To that point, "Many taxonomies exist aiming to classify IoT attacks according to different dimensions such as the adopted wireless communication technologies [61], the different layers of the IoT technology [58,62] or the vulnerability object (i.e. devices, network, software or data)" (Lonetti et al., 2023). This diverse environment of technologies makes the planning of a penetration test more difficult as there is a lack of uniformity regards the terminology used to describe what components and processes need to be tested (Lonetti et al., 2023).

This is significant because, "During the preparation of a penetration test, a series of important decisions need to be made, which have a major impact on the test procedure and consequently the test results. Part of these decisions is to clarify what should be tested (*scope of the test*)" (OWASP, n.d.). Moreover, making these determinations is less obvious when dealing with IoT devices and their potentially numerous and changing interactions with supporting computing infrastructure (Chougule, n.d.). To that point, various hardware, interfaces, and connections are considered part of the relevant security testing scope that a single IoT device

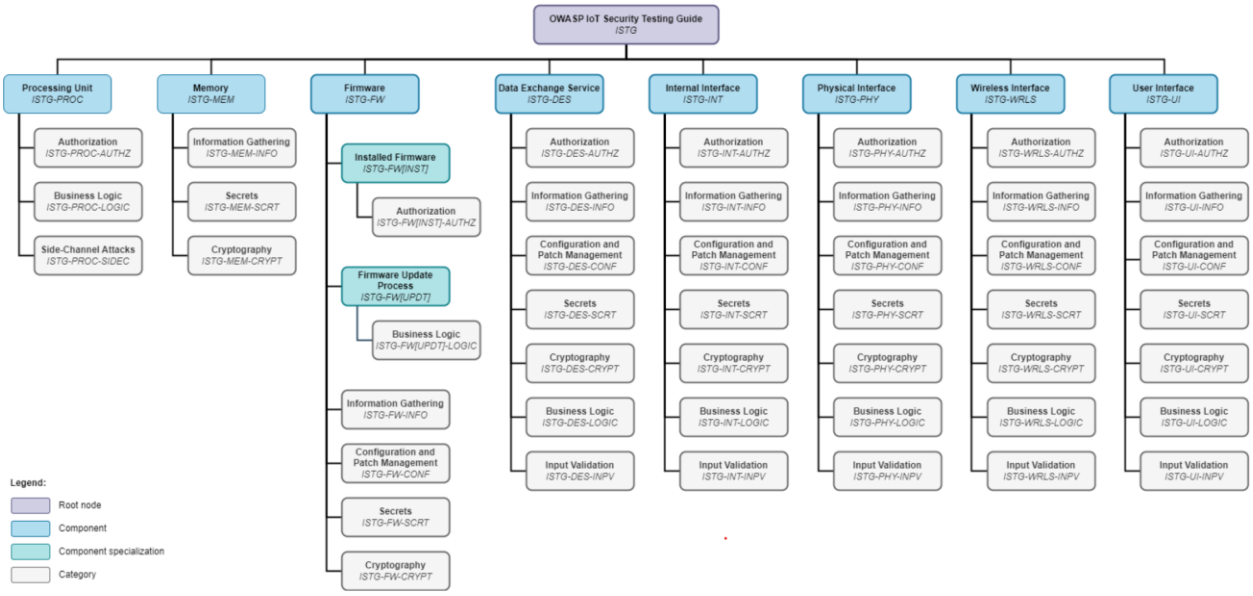
undergoing penetration testing may be subjected to, including the device interactions with its supporting IoT network but excluding the IoT network in general (OWASP, n.d.).

Furthermore, one penetration testing framework designed specifically for the IoT comes from OWASP, and it involves creating a penetration test based on the characteristics of two main components, which are the IoT device(s), and the attacker(s) involved (OWASP, n.d.). To that point, the characteristics of the IoT device in question dictate the types of attacks and attackers it is likely (or can) face, and therefore tests should be tailored to that device (OWASP, n.d.). The benefit of OWASP's framework is it is not designed with a specific IoT device or environment in mind, meaning it can be applied to many different IoT environments (OWASP, n.d.). However, because there are vast amounts of tests that can be performed and because IoT devices and their supporting infrastructure can vary widely, it may be difficult and time-consuming to figure out what specific tests to perform in a given situation (OWASP, n.d.). The OWASP's proposed solution to this problem is to take a "systematic approach" (OWASP, n.d.) to select which tests to incorporate into a pentest plan, starting with general IoT characteristics to create a base penetration test and then adding more individualized tests to create the penetration test for a specific IoT device/environment (OWASP, n.d.).

However, many factors must be considered when creating a pentest plan using the OWASP framework including all the physical components of an IoT device, how it communicates internally, how it communicates with other entities, and detailed characteristics of potential attackers (OWASP, n.d.). To illustrate just how complex this can become, for just a single IoT device the following elements must be considered for pen testing: "Processing unit" (OWASP, n.d.), memory, "Installed firmware" (OWASP, n.d.), "Firmware update mechanism" (OWASP, n.d.), "Data exchange service" (OWASP, n.d.), "Internal interfaces" (OWASP, n.d.),

"Physical interfaces" (OWASP, n.d.), "Wireless interfaces" (OWASP, n.d.), and "User interfaces" (OWASP, n.d.) (OWASP, n.d.). Moreover, each of the elements listed also contains numerous sub-elements, which Figure 1 documents (OWASP, n.d.).

Figure 1: OWASP IoT Device Penetration Testing Domains



(OWASP, n.d.)

Additionally, the different ways each element may be subjected to an attack must be considered from the attacker's point of view, including how (from what distance) an attacker can access the device in question, and what "authorization" (OWASP, n.d.) or amount of privileges a given user (attacker) can obtain (OWASP, n.d.). Moreover, both the access and authorization of a given user (attacker) are assessed on a 1-4 scale with 4 being the highest and most problematic (OWASP, n.d.). Furthermore, making these determinations is not always straightforward as authorization may vary across IoT elements and therefore, "the impact of authorization access levels on the test scope always depends on the specific implementation of the business logic and the authorization/permission scheme per component" (OWASP, n.d.) (OWASP, n.d.). Therefore,

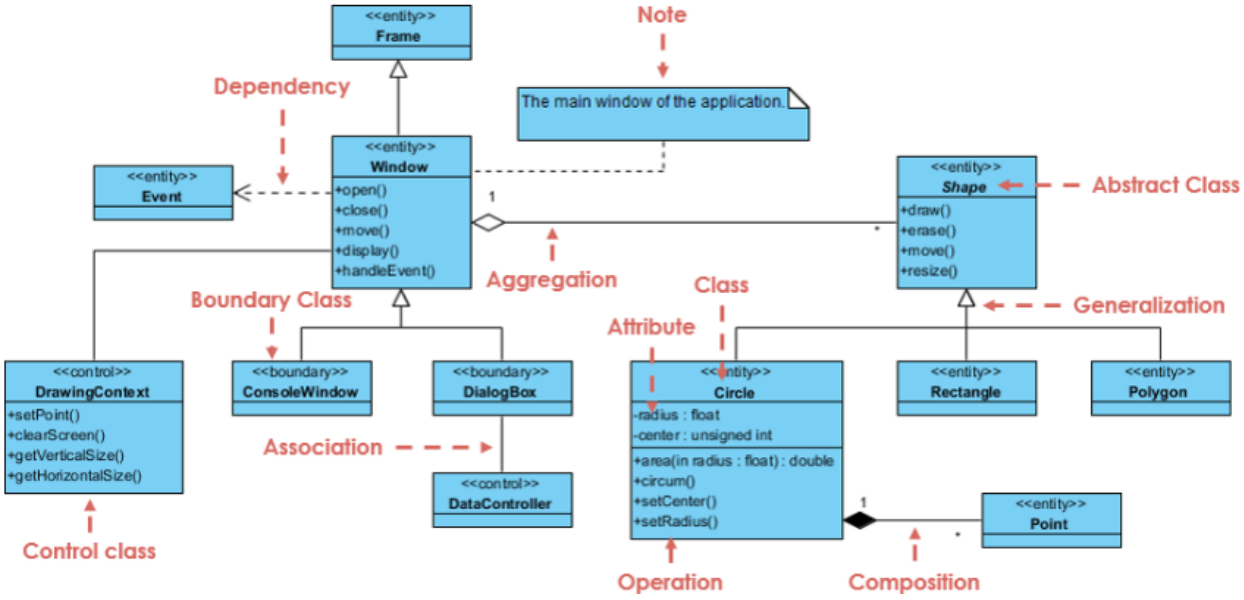
this approach may generally still result in significant time dedicated to test planning, without even beginning to consider the security of the supporting IoT network (OWASP, n.d.). Therefore, an emerging solution to this problem that builds on the framework laid out by OWASP may be found within “Model-based Security Testing (MBST)” (Lonetti et al., 2023) (OWASP, n.d.).

Model-Based Security Testing (MBST)

For starters, MBST or “Model-based Security Testing” (Lonetti et al., 2023) is a type of test methodology under the same security testing umbrella as penetration testing (Lonetti et al., 2023). Moreover, like penetration testing, MBST “addresses the security requirements of the SUT, such as authentication, authorization, confidentiality and integrity of exchanged data” (Lonetti et al., 2023). However, unlike pentesting, MBST uses models of “System under Test (SUT) and/or of its environment” (Lonetti et al., 2023) to automatically create and perform relevant security tests (Lonetti et al., 2023). Moreover, these tests do not need to be created for every new IoT device/environment because the MBST employs “Model-driven Engineering (MDE) technologies to target low-level technical aspects of devices, programming languages or protocols” (Lonetti et al., 2023). To that point, MDE uses “abstraction” (Khalil & Dingel, 2018) to remove features of IoT devices to the point where they can be compared to other IoT devices that normally could not be compared due to their different features i.e. hardware components, communication/network protocols, and/or “programming languages” (Lonetti et al., 2023) (Khalil & Dingel, 2018; Lonetti et al., 2023). Furthermore, this is accomplished in part by “using different formalisms such as Unified Modeling Language (UML) based diagrams, timed automata, or Colored Petri Nets (CPNs)” (Lonetti et al., 2023). Moreover, once the various modeling techniques just mentioned have been utilized to create a generalized outline of the IoT

device and/or network, specific system/communication/testing constraints can be defined as appropriate for a given component/connection (Lonetti et al., 2023). An example of how UML can be used to describe a generic version of a system/application (in this case a GUI) is found in Figure 2 (Visual Paradigm, n.d.).

Figure 2: UML Diagram of Generic GUI



(Visual Paradigm, n.d.)

Furthermore, the ability of MBST is not limited just to modeling an IoT device or network, as there are methods for modeling malicious actor's behavior as well (Lonetti et al., 2023). To that point, "timed automata have been used as a target formalism to model and verify attack trees" (Lonetti et al., 2023), with "the root of the attack tree models the main goal of the attacker, whereas the leaves correspond to the basic attacks" (Lonetti et al., 2023). Next, a collection of various attack trees (corresponding to different attacker goals) can be compiled "into a network of price timed automata" (Lonetti et al., 2023), where different weights are attributed to different potential attacks (Lonetti et al., 2023). Considerations that may be

weighted include the likelihood of an attack, the potential harm that could occur, the number of steps in an attack pattern before harm occurs, the cost of conducting a specific test, and other factors (Lonetti et al., 2023). Furthermore, this newly created network (price timed automata) can be leveraged to automatically create system/network component tests (Lonetti et al., 2023).

Moreover, the described model can be supplemented with "security functional test cases" (Lonetti et al., 2023), which then allows it to produce "abstract test cases" (Lonetti et al., 2023) (Lonetti et al., 2023). Additionally, modeling frameworks such as "Colored Petri Nets (CPNs)" (Lonetti et al., 2023) "have been used for instance to model trusted authentication architectures for IoT applications and verify by model-checking that these architectures satisfy a set of security properties" (Lonetti et al., 2023). Therefore, the capabilities of MBST just described are well-suited to improve various stages of IoT penetration testing (Lonetti et al., 2023; Rak et al., 2022).

Model-Enabled IoT Penetration Testing

Aspects of MBST simultaneously conform with the requirements of a sound IoT device penetration testing methodology while also enhancing its potential efficiency and effectiveness (Lonetti et al., 2023; OWASP, n.d.; Rak et al., 2022). To that point, the models created by MBST can be applied to a wide range of penetration testing activities for IoT devices, regardless of their diverse architecture via the use of MDE (Lonetti et al., 2023). For example, modeling can be used to create "a network of price timed automata" (Lonetti et al., 2023) which can be leveraged to automatically create IoT device/network component tests (Lonetti et al., 2023; Rak et al., 2022). Furthermore, some of the tests derived from the security models of a given IoT device/environment can be performed without human input by automated pentest tools, such as interface testing related to "input validation mechanisms" (Chen et al., 2018), which would

mitigate some of the most prevalent and harmful vulnerabilities associated with IoT devices/networks i.e. ""insecure web interface" and "insecure network services,"" (Chen et al., 2018) (Lonetti et al., 2023; Marijan et al., 2017; Rak et al., 2022). As a result, the proposed methodology i.e., model-enabled IoT penetration testing, would greatly reduce the time that human penetration testers would need to spend on both planning and testing in an IoT environment, which would lead to lower overall costs and possibly more frequent penetration testing (Chen et al., 2018; Johnson, 2022; Lonetti et al., 2023; Marijan et al., 2017; Rak et al., 2022). Stated another way, the combination of MBST techniques with penetration testing would result in automating much of the pentest planning and facilitate the use of automated penetration testing tools to find the more straightforward IoT vulnerabilities, while manual pentesting would only be required to explore the most complex and novel vulnerabilities (Chen et al., 2018; Johnson, 2022; Lonetti et al., 2023; Marijan et al., 2017; Rak et al., 2022).

Additionally, the joining of model-based security testing with IoT penetration testing enables superior "test case design in comparison to the previously used model-free test scripts, and also increased the fault-detection effectiveness" (Lonetti et al., 2023), and "automated traceability of requirements, smooth re-generation [sic] of test procedures in regression testing, intuitive and more efficient analysis of test results" (Lonetti et al., 2023). Moreover, the proposed methodology allows IoT penetration testing to be conducted earlier in the SDLC as the use of modeling allows for vulnerabilities to be "directly addressed during the design phase" (Mahmoodi et al., 2018) (Lonetti et al., 2023; Mahmoodi et al., 2018). Furthermore, when manual penetration testing activities uncover new information, those findings can be incorporated into the model of a given IoT device/environment to improve the model's

performance going forward (Rak et al., 2022). The stages of the described model-enabled IoT penetration testing framework are illustrated in Figure 3 (Rak et al., 2022).

Figure 3: Model-Enabled IoT Penetration Testing Methodology

(Rak et al., 2022)

Therefore, MBST by way of MDE facilitates the creation of various models that describe an IoT device, its network, and the types of attacks it may face across all its components, connections, and interfaces to produce logical (cost-effective) test cases for penetration testers to execute (Lonetti et al., 2023; Mahmoodi et al., 2018). Additionally, where applicable, penetration test cases produced by the proposed methodology can be executed automatically via the "TITAN test suite optimization technology" (Marijan et al., 2017) so long as "test adapters are used, containing all the functions defined in the model" (Lonetti et al., 2023).

Lastly, the testing architecture produced allows tests created for one IoT device/network to apply to different types of IoT devices and networks (Lonetti et al., 2023). This is of note because this capability directly coincides with the goals set out in OWASP (n.d.) *IoT Security Testing Guide*, which states that an effective penetration testing methodology for IoT devices must allow for "the comparison of test procedure (test steps/cases) and results regardless of specific technologies or device types (*comparability*)" (OWASP, n.d.) (OWASP, n.d.).

Furthermore, the proposed methodology also meets the other three requirements listed in OWASP (n.d.) *IoT Security Testing Guide*, which is that the methodology be capable of adding more tests as needed "*expandability*" (OWASP, n.d), facilitate the use of a common testing language "*comprehensibility*" (OWASP, n.d), and finally can be adopted within current pentesting frameworks, i.e. "*efficiency*" (OWASP, n.d.) (Lonetti et al., 2023; OWASP, n.d.; Rak et al., 2022).

Conclusion

In conclusion, it has been demonstrated that the IoT will likely continue to grow rapidly throughout the rest of the decade, moreover, securing the ever-expanding catalog of IoT devices and their supporting networks poses a significant challenge due to their often diverse attributes across a range of technologies (Chen et al., 2018; Chougule, n.d.; EC-Council, 2023). Therefore, to address the challenges facing penetration testing of IoT devices and networks, the proposed model-enabled IoT penetration testing methodology, which conforms to OWASP (n.d.) *IoT Security Testing Guide* was presented (OWASP, n.d.). To that point, the proposed methodology would address many of the limitations related to performing penetration testing of IoT devices and networks by creating a framework that can automatically generate cost-effective test cases, perform some of those tests via automation, and is capable of incorporating feedback from penetration tests performed by humans to improve its model(s); while functioning across a diverse range of IoT devices and networks (Lonetti et al., 2023; Mahmoodi et al., 2018; Marijan et al., 2017; Rak et al., 2022). The result of adopting this methodology would be a potentially significant decrease in time spent by human penetration testers on the planning, attack, and reporting/analysis stages of penetration tests, in addition to an overall more effective IoT penetration testing process (Lonetti et al., 2023; Scarfone et al., 2008).

Lastly, additional areas of interest include using advanced AI, specifically deep learning algorithms to replicate some of the more creative penetration testing activities that currently only human testers can conduct (Cui et al., 2018; Johnson, 2022; Singapore Computer Society, 2021). This could potentially be applied to replace some activities that currently only human penetration testers are capable of, i.e. the execution of complex/novel attacks (Cui et al., 2018; Johnson, 2022; Lonetti et al., 2023; Pope, n.d.; Singapore Computer Society, 2021). Moreover, this capability could theoretically be integrated into a model-enabled penetration testing methodology to further automate the penetration testing process (Lonetti et al., 2023).

Bibliography

Bella, G., Biondi, P., Bognanni, S., & Esposito, S. (2023). PETIoT: PEnetration Testing the Internet of Things. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2302.04900

Bures, M., Cerny, T., & Ahmed, B. S. (n.d.). *Internet of Things: Current Challenges in the Quality Assurance and Testing Methods*. Retrieved March 11, 2024, from https://arxiv.org/ftp/arxiv/papers/1805/1805.01241.pdf

Chen, C.-K., Zhang, Z.-K., Lee, S.-H., & Shieh, S. (2018, April). *Penetration Testing in the IoT Age* (J. Voas, Ed.). https://ieeexplore.ieee.org/document/8352070

Chougule, S. (n.d.). *IoT Device Penetration Testing*. Retrieved March 13, 2024, from https://owasp.org/www-chapter-pune/meetups/2019/August/IoT_Device_Pentest_by_Shubham_Chougule.pdf

Cui, L., Yang, S., Chen, F., Ming, Z., Lu, N., & Qin, J. (2018). A survey on application of machine learning for Internet of Things. *International Journal of Machine Learning and Cybernetics*, *9*(8), 1399–1417. https://doi.org/10.1007/s13042-018-0834-5

EC-Council. (2023, June 28). *IoT Penetration Testing: How to Perform Pentesting on a Connected Device*. Cybersecurity Exchange. https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/iot-penetration-testing-how-to-secure-your-connected-devices/

Hardaway, J. (2023, October 17). *A Pacemaker Is An Example Of What Type Of IoT Device?* Robots.net. https://robots.net/tech/a-pacemaker-is-an-example-of-what-type-of-iot-device/#:~:text=With%20the%20advent%20of%20Internet%20of%20Things%20%28IoT%29

Johnson, K. (2022, February 8). *Pros and cons of manual vs. automated penetration testing*. SearchSecurity. https://www.techtarget.com/searchsecurity/feature/Pros-and-cons-of-manual-vs-automated-penetration-testing

Khalil, A., & Dingel, J. (2018). Optimizing the Symbolic Execution of Evolving Rhapsody Statecharts. *Advances in Computers, 108*, 145–281. https://doi.org/10.1016/bs.adcom.2017.09.003

Lonetti, F., Bertolino, A., & Di Giandomenico, F. (2023). Model-based security testing in IoT systems: A Rapid Review. *Information and Software Technology*, *164*. https://doi.org/10.1016/j.infsof.2023.107326

Mahmoodi, Y., Reiter, S., Viehl, A., Bringmann, O., & Rosenstiel, W. (2018, August 1). *Attack Surface Modeling and Assessment for Penetration Testing of IoT System Designs*. IEEE Xplore. https://doi.org/10.1109/DSD.2018.00043

Marijan, D., Liaaen, M., Gotlieb, A., Sen, S., & Ieva, C. (2017). *TITAN: Test Suite Optimization for Highly Configurable Software*. https://doi.org/10.1109/icst.2017.60

OccamSec. (2023, August 7). *The Human Touch in Cybersecurity: Why AI Can't Fully Replace Penetration Testers - OccamSec*. Occamsec.com. https://occamsec.com/the_human_touch_in_cybersecurity/#:~:text=But%20as%20we%20journey%20deeper

OWASP. (n.d.). *OWASP IoT Security Testing Guide*. Owasp.org. Retrieved March 12, 2024, from https://owasp.org/owasp-istg/01_introduction/index.html

Pope, J. (n.d.). *Human vs AI In Pen Testing*. Cyber Smart Consulting Ltd. Retrieved March 14, 2024, from https://cybersmartconsulting.com/ai-in-pen-testing/#:~:text=AI%20Pen%20Testing%20Systems

Rak, M., Salzillo, G., & Granata, D. (2022). ESSecA: An automated expert system for threat modelling and penetration testing for IoT ecosystems. *Computers & Electrical Engineering*, *99*, 107721. https://doi.org/10.1016/j.compeleceng.2022.107721

Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). *Special Publication 800-115 Technical Guide to Information Security Testing and Assessment Recommendations of the National Institute of Standards and Technology*. NIST. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

Singapore Computer Society. (2021). *Simplifying the Difference: Machine Learning vs Deep Learning*. https://www.scs.org.sg/articles/machine-learning-vs-deep-learning

US Department of Homeland Security. (2006, September). *Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software* (S. T. Redwine, Jr., Ed.). Learn.umgc.edu. https://learn.umgc.edu/content/enforced/383271-022073-01-2195-GO1-9041/Common_Body_of_Knowledge2007.pdf?_&d2lSessionVal=yHNXTIp6y56ZPEX8jKq29unVQ&ou=313879&_&d2lSessionVal=GjydqAcr8UMZkMm5yxUyya8no&ou=383271&ou=930353

Visual Paradigm. (n.d.). *UML Class Diagram Tutorial*. Visual-Paradigm.com. Retrieved March 17, 2024, from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/