SEPP 4.0 - Evaluation of Hands-On IoT-Security Exercises

Louis Ebneth and Sebastian Fischer
Faculty of Computer Science and Mathematics
Ostbayerische Technische Hochschule Regensburg
Regensburg, Germany
e-mail: louis1.ebneth@st.oth-regensburg.de | sebastian.fischer@oth-regensburg.de

***Abstract*—Teaching Internet of Things (IoT) security effectively remains a challenge due to the gap between theoretical concepts and their application in real-world systems. This paper presents the fourth iteration of the Security Education and Penetration-Testing Platform (SEPP) project, a gamified, hands-on learning environment that uses vulnerabilities in real IoT devices to teach IoT security concepts to computer science students. Over the course of the past semester, a weekly exercise session was conducted, during which ten previously developed exercise sheets were evaluated through observations and student questionnaires. The results show high student engagement, strong perceived learning outcomes, and clear benefits from working with realistic attack scenarios and real IoT devices. However, the evaluation also revealed areas in need of improvement, including device selection, sequencing of the exercises, clarity of instructions, preparation of the exercise sessions, and time management. Based on these findings, this paper details the next steps in the further development of the SEPP platform and the continuous improvement of practical IoT security education at the OTH Regensburg.**

Keywords-Internet of Things; IoT Security; Cybersecurity Education; Hands-On Learning.

I. Introduction

The Internet of Things (IoT) has become an integral part of our daily lives. From smart home appliances, to wearables, and industrial equipment, the ability to connect devices and systems across the internet has opened up new possibilities for automation, control, and monitoring in all aspects of life [1]. The heterogeneous nature of the IoT ecosystem and the rapid development of IoT devices in recent years have resulted in a lack of robust security mechanisms and created vulnerabilities that expose IoT systems to a wide range of cyberthreats [2][3].

A. SEPP - Security Education and Penetration-Testing Platform for IoT

The SEPP platform presents a gamified approach to teaching IoT Security to computer science students by providing the students with a way of engaging with real-world examples of the cybersecurity risks introduced in the IoT Security lecture through a series of hands-on exercises [4][5].

The platform simulates a home environment equipped with various IoT devices that contain security flaws, such as smart plugs, security cameras, or light bulbs. Each device is associated with one or more exercise sheets that have been created by students as part of their bachelor's thesis or study project.

B. Evaluation of the Exercises

Over the course of the past semester, two students conducted a weekly exercise session, where they evaluated the ten exercise sheets based on observations during the session and a questionnaire that the participating students had to fill out at the end of the session [6][7].

Section 2 provides a brief overview of the exercise contents. Section 3 discusses the first evaluation results. Section 4 outlines the next steps in the further development of the SEPP platform.

II. Exercises

The exercise sheets were designed as the practical part of the IoT Security lecture and are therefore aligned to complement its contents. The students work with the industry-standard penetration testing tools introduced in the lecture and perform a wide variety of different attacks and audits on real IoT devices.

The exercises cover a wide range of protocols used in IoT devices, including Wi-Fi, Bluetooth, Bluetooth Low Energy (BLE), Telnet, Secure Shell (SSH), Message Queuing Telemetry Transport (MQTT), Transport Layer Security (TLS), and more.

Exemplary tasks include:

1) Determining the Internet Protocol (IP) address of a smart outlet by conducting a network scan using the network exploration tool Nmap.
2) Capturing control commands sent to an IoT light bulb via the Telnet protocol using Wireshark and subsequently performing a replay attack by retransmitting the captured packets with Python.
3) Performing an Address Resolution Protocol (ARP) spoofing-based Man-in-the-Middle (MITM) attack on a Siemens LOGO! DDC module using Ettercap to sniff and intercept its network traffic.
4) Executing Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks on various devices using a Python script.
5) Gaining access to a web interface protected by default credentials through a dictionary-based brute-force attack with the Python package Selenium.
6) Hijacking the active user session of a web interface by recreating a session cookie from a Wireshark capture containing a transmission of the session ID.

III. Discussion of the First Results

In general, the feedback on the exercises was found to be positive. The connection to the IoT Security lecture was perceived as apparent, and the students stated that they were able

to apply what they had learned. The students highly praised the hands-on approach and especially emphasized the joy of being able to compromise real IoT devices, particularly when the instructions provided them with very limited information about the device to start with. The different attack vectors covered in the exercises were reviewed as interesting and educational, offering clear perspectives on cybersecurity issues in IoT devices. For most exercises, the instructions were rated as understandable and easy to follow, and one exercise even received praise for its puzzle-like design. The students were surprised by the ease of using tools like Nmap and Wireshark to collect the data necessary to perform the various attacks. The simplicity of various attacks was also pointed out. This included the ability to perform a DoS attack with a simple Python script and the ability to perform a MITM attack with a single Ettercap command.

The following subsections highlight the core issues that were found during the evaluation and how to best address them.

A. Device Choice

The choice of the IoT device was found to be suboptimal for some exercises.

For example, the students pointed out that determining the success of a DoS attack on a smart Wi-Fi outlet is rather difficult if there is no device plugged into it that would indicate the failure of sending control sequences from the smartphone application to the outlet. Another exercise had the students perform a replay attack on an IoT light bulb. However, the device was unable to handle the increased number of requests created from multiple groups of students doing this at the same time. The attack therefore resulted in a DoS scenario, with the bulb becoming unresponsive for a few minutes.

The victim devices used in these exercises therefore need to be reevaluated. They should provide clear feedback on the success of an attack whenever possible and should not exhibit any unintended behavior.

B. Repetition of Attacks

The steps of first identifying a device on the network with Nmap and subsequently performing a DoS attack on it were included in four of the ten exercise sheets. Repeating the same attack, especially in an equivalent fashion, should be avoided, as this does not provide any additional educational value.

C. Arrangement within the Semester

The first exercise used Nmap to identify a smart outlet, launched a DoS attack, and analyzed the resulting traffic with Wireshark. The second exercise covered vulnerabilities in Wi-Fi Protected Access (WPA) and Wi-Fi Protected Setup (WPS) and again required device identification using Nmap. However, the chapters that introduce Nmap, WPA, and Wireshark have not been discussed in the lecture at this point of the semester.

This emphasizes the need to re-evaluate the arrangement of the exercises on the semester timeline, as it is not feasible to have the students working with tools before they have been introduced in the lecture.

D. Instructions and Guidance

All instructions should be clear and easy to follow.

Over the course of the evaluation, multiple errors in the exercise instructions were identified that need to be corrected. This includes incorrect sample commands and Python scripts that did not produce the desired results. Additionally, since the exercise sheets were created at home by different students, discrepancies arose between the instructions and the final computer lab setup. This includes incorrect Wi-Fi Service Set Identifiers (SSIDs) and passwords, as well as mismatched Media Access Control (MAC) addresses, all of which need to be corrected as well. In some cases, the students remarked that the Python scripts they had to complete as part of the exercise did not contain any information about what to do beyond a *"TODO"* comment marking the line. Additional comments that explain what is to be done need to be added to these scripts. Some instructions were also found to be too vague and need to be concretized. One example had students analyze a Wireshark network capture without specifying what to look for in the packets, another step only referred to *"any Wi-Fi device"*, which left the students wondering which one to use.

The tools and Python libraries that are used in the exercises, but not covered by the lecture, should be introduced with a short description of the tool and should provide concrete example commands. This can be done in short on the instruction sheet itself or on a separate information sheet for a more extensive resource. Exemplary resources include Ettercap, Reaver, Selenium, and Bleak.

Some students also provided feedback on the lack of such information and examples for the tools that were introduced in the lecture. However, this could easily have been solved by attending the lecture or taking a look at the course material. In addition, online research and AI tools, such as ChatGPT, can provide further guidance.

E. Understanding

The goals and key takeaways of each exercise should be clear. This was not always the case.

One exercise had the students read out information from a Bluetooth smart lock using the nRF Connect app. The subsequent steps of the exercise did not utilize this information further or explain how it could be used, which left the students wondering about the relevance of this step. Another exercise had the students remark that a session hijacking attack on a Siemens LOGO! DDC module was too easy, as the session ID was part of the Uniform Resource Locator (URL) of the web interface and could therefore easily be captured with Wireshark, and that the web interface did not provide feedback when the attack was successful.

It should be made clear that these are vulnerabilities in real IoT devices and not custom-engineered lab creations.

F. Preparation of the Exercise Sessions

In an ideal situation, all devices would be firmly integrated into the SEPP platform and configured as necessary for the

exercises. However, this is not yet possible due to the still-evolving nature of the project.

The exercise leader must therefore be provided with a clear and easy-to-follow setup guide for each exercise that outlines what has to be prepared and checked before the exercise session. This is currently only the case for three out of the ten exercises, with one of the guides being received as insufficient.

G. Wi-Fi Issues

The original setup of the SEPP had a Raspberry Pi 5 hosting a Wi-Fi network to which the students and IoT devices were connected. This network was found to be rather inconsistent and unable to support the desired number of devices. Midway through the semester, a GLMT300N-V2 Mini Smart Router was added to serve as an access point and replace the Raspberry Pi's Wi-Fi.

H. Timely Constraints

The weekly exercise session has a duration of 90 minutes. The time required for the individual exercise sheets varied greatly. Some exercise sheets were too extensive to be completed in 90 minutes, while the students finished others within 45 minutes. The length of existing exercises must be adjusted to approximately match the length of the exercise session. Future exercise sheets should also be designed with the time frame in mind and need to be tested for compliance prior to their inclusion in the course.

Three areas were clearly identified where time could be saved during the sessions:

1) Provisioning of Necessary Tools: About a third of the first exercise session was lost because the students had to install the required tools. The resources and tools required for the exercise should therefore be communicated in advance of the weekly session. Later sessions had a message posted on the E-learning platform a couple of days prior, that provided the students with a list of the necessary tools and instructed them to install these. Tools that are used repeatedly across most exercises, such as Nmap and Wireshark, can be communicated to all students in the first lecture at the beginning of the semester. Alternatively, every student can be tasked with installing a Kali Linux virtual machine in a free virtualization software like VirtualBox, as this distribution already contains most of the required tools. For this, links to setup tutorials should be provided. Another option is to provide students with ready-to-go native Kali Linux laptops and Android phones in the computer lab. This would also eliminate issues, such as a virtual machine not having access to the Wi-Fi adapter of the host device, and therefore being unable to switch it into monitor mode, or issues with the iOS version of the nRF Connect app not displaying certain information, such as the Bluetooth address of an IoT device.

2) Handling of Lengthy Subtasks: Some steps, such as a full Nmap User Datagram Protocol (UDP) port scan, are known to take quite a while. The instructions did not mention limiting the range of ports to scan, which caused the students to wait for the full scan to finish. This was especially prominent if the results of such a step were needed for the further steps of the exercise. Appropriate tips should be added to steps like this in order to minimize wasted time.

3) Distribution of Files: The students work with numerous Python scripts throughout the exercises. In some cases, these scripts were provided only as plain text or images on the exercise sheets, which the participants had to manually transcribe into their editors. One exercise also involves a Java-based server application for which only a compiled **.jar* file and screenshots of the source code were available.

A central Git repository is proposed to manage and efficiently distribute all source files. This repository could also be extended to hold and distribute the exercise sheets.

IV. Conclusion and Future Work

The initial results of the evaluation indicate that the SEPP platform is a promising and well-received tool to educate future IT professionals about IoT cybersecurity. However, they also show that it is still an evolving project that is far from complete and will require continuous improvement in the future. The issues mentioned in Section 3 will be addressed over the course of the next semester, as the lecture is only held in the winter term.

The results of this paper can also be used outside the SEPP platform to better design similar practice-oriented exercises based on this experience. Therefore, the results not only serve to improve the platform, but can also help other educators with the development of better tasks.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, Retrieved: 2026.02.18, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2010.05.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128610001568.

[2] R. H. Weber, "Internet of Things – New security and privacy challenges," *Computer Law & Security Review*, vol. 26, no. 1, pp. 23–30, 2010, Retrieved: 2026.02.18, ISSN: 2212-473X. DOI: https://doi.org/10.1016/j.clsr.2009.11.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0267364909001939.

[3] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015, Retrieved: 2026.02.18, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2014.11.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128614003971.

[4] D. Hauser, J. Graf, S. Fischer, and R. Hackenberg, "SEPP: Security Education and Penetration-Testing Platform for IoT," in *Proceedings of The European Conference on Education 2025*, Retrieved: 2026.02.18, 2025, pp. 443–453. DOI: https://doi.org/10.22492/issn.2188-1162.2025.36.

[5] D. Hauser and S. Fischer, "SEPP 2.0 - Advanced IoT Hacking Scenarios for Hands-on Security Education," 2025, Unpublished.

[6] I. Edelmann, "IoT Vulnerabilities: Evaluation and Improvement of the Exercises," Unpublished bachelor's thesis, 2025.

[7] L. Herrmansdörfer, "Evaluation and Analysis of Cyber-Security Challenges in Relation to the Internet of Things," Unpublished bachelor's thesis, 2025.