SEPP 2.0—Advanced IoT Hacking Scenarios for Hands-On Security Education

Dominic Hauser, Ostbayerische Technische Hochschule Regensburg, Germany
Julian Graf, Ostbayerische Technische Hochschule Regensburg, Germany
Sebastian Fischer, Ostbayerische Technische Hochschule Regensburg, Germany

The Asian Conference on Education & International Development 2026
Official Conference Proceedings

Abstract

This paper presents SEPP 2.0 (Security Education and Penetration-Testing Platform), the second stage of a practical teaching platform designed to strengthen IoT security education through direct interaction with real devices. Building on the first SEPP version, which was successfully presented and published at IAFOR (Hauser et al., 2025), SEPP 2.0 extends the concept with a larger set of structured, course-aligned exercises within the IoT Security program at Ostbayerische Technische Hochschule Regensburg (OTH Regensburg). The platform integrates real IoT devices such as smart plugs, light bulbs, smart locks, and a Raspberry Pi into a portable suitcase environment. Each exercise is designed to make typical security weaknesses and attack steps tangible for students. The sequence of tasks reflects realistic phases of a security assessment—from information gathering and configuration review to network and communication analysis. Common tools are applied in a guided and safe context to observe network behavior, detect insecure communication, and understand reproducible attack patterns such as replay or denial-of-service. SEPP 2.0 connects these technical elements with reflective learning on current standards and regulations, including ETSI EN 303 645 and the EU Cyber Resilience Act. This combination helps students not only identify vulnerabilities but also translate them into technical and organizational protection requirements. Beyond its local use at OTH Regensburg, SEPP 2.0 was developed with transferability in mind. Its modular structure and detailed documentation allow other universities to adopt, adapt, and expand the platform to fit their own cybersecurity or engineering curricula, making SEPP 2.0 a sustainable model for practice-oriented security education.

Keywords: IoT security, penetration testing, hands-on education, smart home, cybersecurity

iafor
The International Academic Forum
www.iafor.org

Introduction

Advantages and Disadvantages of IoT

The Internet of Things (IoT) has been continuously expanding in recent years, with smart home devices becoming increasingly present in both private homes and industrial environments. These technologies offer numerous advantages, such as more efficient communication, automation of daily routines, and optimization of supply chains in various sectors.

Despite these benefits, the security risks associated with IoT should not be overlooked. To support their wide range of use cases, IoT devices rely on multiple communication protocols and interfaces and must be compatible with numerous platforms such as HomeKit, Alexa, Nest, or SmartHomeAssistant. In theory, this complexity demands frequent vulnerability checks and continuous firmware updates. However, in practice, many devices, especially older or low-cost ones, no longer receive updates, leaving them exposed to potential attacks. Users are often unaware of these risks, making the devices an attractive target for attackers.

SEPP—A Hands-On Platform for Teaching IoT Security

To effectively address the growing risks associated with IoT technologies, it is essential to equip future computer science professionals with a thorough understanding of relevant protocols and the ability to identify potential vulnerabilities. At OTH Regensburg, an IoT security course is already offered, covering a wide range of technologies such as SSH, Nmap, Wi-Fi, Bluetooth Low Energy (BLE), and Wireshark from a theoretical perspective.

However, studies have shown that theory alone is often insufficient to truly internalize the complexity of IoT security. Practical, hands-on exercises, especially those focused on common vulnerabilities such as insecure communication or user tracking, have proven to significantly improve students' ability to understand and mitigate real-world threats (Kolias et al., 2016). Gamified approaches such as Capture the Flag (CTF) competitions further support this goal by increasing motivation and encouraging analytical thinking (Legg et al., 2021). Furthermore, integrating real devices into course curricula has been shown to substantially improve understanding of IoT architectures and security concepts (Sahrani et al., 2024).

To complement the theoretical foundation of the IoT security course with applied experience, the Security Education and Penetration-Testing Platform (SEPP) was developed. As introduced in the first publication, SEPP provides a dedicated learning environment in the form of a portable suitcase containing various IoT devices, including smart plugs, light bulbs, Nuki locks, cameras, and a Raspberry Pi. These components are integrated into a custom-built network to simulate realistic smart home scenarios for practical exercises.

Key Findings and Motivation for Further Development

In the initial implementation of SEPP, a series of practical exercises was tested with a group of students enrolled in the IoT security course. These exercises included typical attack scenarios such as port scanning, traffic analysis, DoS attacks, and replay attacks using tools like Nmap, Wireshark, Netcat, and Scapy. All students were able to complete the tasks with only minimal guidance and successfully applied their theoretical knowledge in a practical environment.

The results confirmed that real-world scenarios and hands-on challenges can significantly improve learning outcomes in the field of cybersecurity. Students reported that especially the direct interaction with real devices helped them understand how vulnerabilities work and how different protocols behave under attack conditions. Although the initial results proved to be very positive, more exercises and tests are needed to confirm and consolidate these findings.

Based on this feedback, the platform is now being expanded. The second part of this work introduces additional exercises, structured along the typical steps of a hacking process as used in the IoT Security course. Each phase is accompanied by one or more hands-on tasks that deepen students' understanding of both technical tools and attack logic. The goal is to gradually build a more structured, flexible, and comprehensive version of SEPP that supports a wider range of use cases and difficulty levels.

Exercises

Existing Exercises

This section provides an overview of the existing and partially tested exercises from the previous version of the SEPP platform. All exercises were designed with the aim of introducing students to common cybersecurity concepts and tools, such as port scanning, packet analysis, traffic monitoring, and DoS attacks with real IoT devices using well-known tools like Nmap, Wireshark, Netcat, and Scapy.

For a detailed description of each exercise, please refer to the previous publication (Hauser et al., 2025).

The exercises included:

- Network scanning with Nmap to identify IoT devices and open ports within a local subnet
- SYN-flood DoS attack using Python and Scapy to simulate denial-of-service behavior on a smart plug
- Traffic monitoring with Wireshark to observe SYN/SYN-ACK patterns and analyze device responses
- Unencrypted command injection using Telnet to directly control a smart bulb via JSON messages
- Replay attacks by resending captured packets to trigger actions on IoT devices
- Encryption testing using Netcat and Wireshark to determine whether communication is in plaintext
- App traffic analysis by capturing packets between smartphone apps and their associated devices.

These exercises proved highly effective in helping students understand the practical use of the listed tools and provided valuable insight into areas that should be prioritized and further developed in the next version of the SEPP project.

Exercise Design Aligned to the Course Structure

Since the whole SEPP project is designed to the practical-part of an already existing course of the OTH Regensburg, the decision was made to create the next exercises aligned to the content

of the course. This should enable a flawless integration into the course, after successfully testing the created exercises.

As SEPP is intended to complement the practical part of the existing IoT Security course at OTH Regensburg, the next set of exercises was aligned with the structure and content of the course. This alignment is expected to enable a more flawless integration into the existing contents followed by a detailed evaluation in the upcoming semester.

Course Chapter 1: IoT and Shodan

This chapter is primarily intended to introduce students to the fundamentals of IoT security and provide an initial overview of common vulnerabilities, device characteristics, and technical limitations.

Among other topics, it covers the OWASP Top 10 vulnerabilities, a regularly updated website with a list of the most critical security issues in the field. The list includes threats such as insecure network services, insufficient authentication, or lack of encryption, which are especially relevant in the IoT context (OWASP, n.d.).

Students also get a first look at the Shodan.io platform. Shodan is a search engine that indexes internet-connected devices and services, allowing users to find unsecured systems, exposed ports, and even specific IoT hardware worldwide. It demonstrates how easy it can be to discover vulnerable devices on the internet with just a few search queries (Shodan, n.d.).

To better illustrate the real-world impact of insecure IoT systems, the Mirai botnet is also introduced in this chapter. Mirai infected over 600,000 devices at its peak by exploiting a hardcoded list of 64 default username and password combinations. The source code of the original botnet is publicly available and serves as a clear example of how poor default security settings can be abused at scale (Antonakakis et al., 2017).

Exercise of This Chapter

Students are encouraged to become familiar with the Shodan.io platform in order to gain an understanding of the general number of vulnerabilities present and the typical number that occur per device. They are also motivated to search for familiar devices within the platform.

Course Chapter 2: Standards and Regulation

In this part of the course, students are introduced to existing IoT security standards and legal requirements that are relevant for consumer devices in Europe and beyond. The goal is to familiarize students with both the technical side and the regulations, as is the case in professional practice.

One of the most relevant and widely accepted standards is ETSI EN 303 645, which defines basic cybersecurity requirements for consumer IoT devices. The standard includes provisions such as the requirement of unique passwords per device, a public vulnerability disclosure process, secure software updates, and minimized attack surfaces. It also addresses data protection, telemetry handling, and secure communication.

The chapter also addresses the Cyber Resilience Act (CRA), a regulation introduced by the European Union aimed at improving the cybersecurity of digital elements in connected products across their entire lifecycle. It outlines requirements for manufacturers to ensure secure design, development, and maintenance, and places greater responsibility on vendors (European Commission, 2025).

To make the content more tangible, several concrete provisions are discussed in the course: for example, ensuring password strength and uniqueness, managing vulnerabilities within 90 days, and enabling secure remote updates through best-practice cryptographic methods.

By the end of this chapter, students should understand the importance of legal and regulatory frameworks and should be able to prescribe technical requirements for a standard such as ETSI EN 303 645 or the Cyber Resilience Act (European Commission, 2025).

Exercise of This Chapter

Given the theoretical nature of this chapter, which is situated in the early stages of the course, there is no practical exercise for now. The students have not yet acquired certain essential knowledge, which will be imparted in the ensuing chapters.

Course Chapter 3: Information and Configuration

This chapter of the course is designed to provide the students with an initial in-depth insight into the side of IoT devices and their potential vulnerabilities. In order to achieve this objective, the first step is to describe the information typically provided by the manufacturer or retailer. This information includes technical details, the current software version, the duration of device updates, and an address to contact the manufacturer to report any security vulnerabilities.

In addition, students are shown how and where to search for already known vulnerabilities related to specific devices. The course introduces platforms such as https://cve.mitre.org and https://www.cvedetails.com, which allow targeted searches for reported vulnerabilities. These sites provide essential data, including severity ratings, disclosure dates, known exploitability (e.g., remote code execution via buffer overflows), and information about whether the vulnerability has already been patched.

The students also learn why this type of research is important when assessing a device's security posture. For example, some vulnerabilities affect multiple devices using the same code base or shared third-party libraries (e.g., OpenSSL). In other cases, outdated communication protocols or rebranded hardware can make it difficult to track and evaluate known issues properly.

To put these aspects into context, the course uses a sample device and guides students through the configuration and setup process. The goal is not only to understand technical details, but also to critically evaluate where and how vulnerabilities can emerge.

Exercise of This Chapter: Writing an Essay

To support the content of this and the following chapters, students start with a written security-focused paper on a selected IoT device. The goal is not only to try to perform technical attacks, it is also to critically examine the device's setup and configuration process and identify potential security issues based on available information (e.g. from the former described websites).

Students work with a test device, such as the HP OfficeJet Pro 6970 or a similar IoT product. For this chapter, students document relevant aspects of the setup process, including the use of local or cloud accounts, default passwords, enabled services and interfaces (e.g., Wi-Fi Direct-Use), password policies, and whether users can increase or reduce security through the available options. They are also asked to research known vulnerabilities and briefly assess whether related issues exist for the device or similar models from the same manufacturer.

Course Chapter 4: Nmap, Telnet and SSH

Chapter 4 introduces students to core network analysis and communication tools that are essential when working with IoT devices. The focus is on three key technologies: Nmap for network and port scanning, Telnet for interacting with services over plaintext protocols, and SSH for secure remote access.

Students learn how to perform different types of scans using Nmap, such as host discovery within a subnet and service detection on specific devices. They also analyze scan results to understand which ports are open, what services are running, and how this information could be relevant in a security context. As a practical example, the HP OfficeJet Pro 6970 is used to demonstrate Nmap scans, which reveal open ports such as 80 (HTTP), 443 (HTTPS), and 9220 (Telnet).

Telnet is introduced as an outdated but still occasionally used protocol, especially in older or misconfigured devices. Students are taught how Telnet can be used to interact with exposed services, and why its unencrypted nature presents a major security risk. The HP GGW service on port 9220 serves as an example for hands-on interaction and exploration.

Finally, SSH is presented as a secure alternative to Telnet. Students learn how SSH encrypts communication, supports key-based authentication, and is commonly used for secure remote access to IoT devices or backend systems.

This chapter continues the device analysis started in Chapter 3, now focusing on exposed ports and communication protocols. The results contribute to the overall security evaluation of the selected IoT device.

Exercise of This Chapter: Investigation of an IoT Smart Plug

This exercise has previously been evaluated in the context of the first paper on the SEPP project, resulting in very positive outcomes. Therefore, it was decided to adopt and integrate the exercise without implementing any significant modifications.

Students start by identifying the IP address of the EIGHTREE smart plug within the local network and scanning it for open TCP and UDP ports using nmap. This introduces key concepts like IP ranges, MAC address identification, and basic service enumeration.

In the second task, a SYN-flooding DoS attack is simulated using a partially completed Python script and the Scapy library. Students fill in the missing target parameters and observe how the attack affects device availability. Later parts of this exercise like the investigation of the network traffic using Wireshark are optional for the students since these techniques were not introduced yet.

Course Chapter 5: WiFi

In about half of the course, students are introduced to the topic of WiFi. First, they learn basic facts such as the IEEE 802.11 standard and the different frequency bands used in wireless communication (2.4 GHz, 5 GHz and 60 GHz).

Afterwards, the chapter focuses on the various encryption methods used in WiFi communication. Students start with WEP, which is completely outdated and can be cracked within minutes due to weak initialization vectors and static keys. Tools like Aircrack-ng are discussed as part of the theoretical background.

Then, the course moves on to WPA, which has been considered broken since 2008. The transition to WPA2 added AES encryption and improved reliability, but also introduced new attack vectors like KRACK. Finally, WPA3 is introduced as the current standard, offering forward secrecy and stronger key exchange mechanisms, but with limitations such as downgrade attacks (e.g. Dragonblood).

Authentication methods are also discussed. Students learn the difference between Pre-Shared Key (PSK) and Extensible Authentication Protocol (EAP), including advantages such as better user management. WiFi Protected Setup (WPS) is introduced as a convenience feature that can create security problems due to design flaws and brute-force vulnerability. The chapter concludes with a short demonstration of the WiFi hacking tool aircrack-ng, which students may also explore as part of their ongoing essay work.

Exercise 1 of This Chapter: Brute-Force WPS Hacking

Two exercises have been created for this chapter of the course, of which only the first relates directly to the WPS hacking content. The second exercise deals with the topic of Intrusion Detection Software, which requires both, initiative and the transfer of previously learned methods to new content. It is not yet clear whether both will be used within a semester, as they still need to be tested for practical suitability and duration.

In the first exercise, students perform a practical attack on a WiFi network protected by WPS (Wi-Fi Protected Setup). Using the tool reaver, they brute-force the WPS PIN to gain access to the network without knowing the actual password. The setup includes the preconfigured Raspberry Pi network with multiple IoT devices including an Android smartphone of the SEPP project. Before launching the attack, students activate monitor mode on their wireless adapter and identify relevant network parameters such as BSSID and channel. The goal is to demonstrate how insecure PIN secured WPS implementations can be exploited in practice and why disabling WPS is a crucial security measure.

Exercise 2 of This Chapter: WiFi Port Analysis

In this advanced exercise, students are encouraged to put their acquired knowledge into practice themselves. For this purpose, the students should familiarize themselves with the topic of Intrusion Detection Systems (IDS) and how they can be used to detect suspicious activity in IoT environments. A custom IDS script (IDS.py) is provided to run in parallel with a simple server. The server listens on a predefined port, and students initiate network communication via telnet and controlled nmap scans. The IDS identifies specific scan types based on

predefined rules and raises alerts in real-time. All captured traffic is logged and saved as a .pcap file for further analysis.

In the second part, students may optionally use Wireshark to analyze the recorded network traffic with the .pcap file. Because Wireshark is not yet introduced in the course, this task is only optional. Interested students are encouraged to explore it on their own by attempting to identify communication patterns, scan behavior, and how specific packets triggered IDS alerts. This optional step provides additional insight into traffic-level detection and supports a deeper understanding of how IDS mechanisms operate.

Course Chapter 6: Bluetooth

This chapter introduces Bluetooth as a key technology in many IoT devices and outlines its basic functionality, including operating frequencies, connection models, and protocol structure. Special focus is placed on Bluetooth Low Energy (BLE), which is widely used due to its low power consumption. Students learn about Bluetooth profiles and GATT (Generic Attribute Profile), which define how services and data are structured on BLE devices. The chapter also introduces basic security concepts such as pairing, bonding, authentication, and encryption—starting from legacy PIN-based pairing to modern Secure Simple Pairing (SSP) with elliptic curve cryptography.

A major part of the chapter is dedicated to practical Bluetooth hacking techniques. Students are introduced to tools such as btlejack, gatttool, hcitool, bluemaho, sdptool, and redfang. These tools allow for device scanning, brute-force pairing attempts, GATT-level interactions, jamming, hijacking, and spoofing of Bluetooth traffic. The functionality and limitations of each tool are briefly discussed, along with typical use cases in a lab setting.

To illustrate real-world impact, several Bluetooth vulnerabilities are covered in this chapter, including BlueBorne, BIAS, KNOB, Braktooth, and BLUFFS. Each of these demonstrates how flaws in Bluetooth can be exploited, even in newer versions of the protocol. Students are asked to review their assigned devices for Bluetooth vulnerabilities as well as performing a GATT scan for their essay.

Three different exercises were created for this chapter, all of which are still under evaluation and not yet fully tested in class. Exercise 3 is still in development and may be included in future iterations of the course.

Exercise 1 of This Chapter: BlueDucky Attack on Android Devices

This exercise demonstrates a real-world Bluetooth vulnerability on Android smartphones using the BlueDucky framework. The goal is to exploit outdated security patches and inject keystrokes via a Bluetooth HID profile. Students begin by identifying their device's patch level and checking for unpatched CVEs. On a compatible attack system of the SEPP project, the BlueDucky.py script is run to simulate keystroke injection. Predefined payloads are used to send SMS messages or download APKs to the victim device. Students are also encouraged to develop their own payloads and reflect on potential mitigation strategies.

Exercise 2 of This Chapter: BLE Traffic Capture and Analysis

In this exercise, students analyze the communication of Bluetooth Low Energy (BLE) devices using standard Linux tools. The focus is on passive monitoring and capturing of BLE traffic with btmon, bluez, and hcitool. Then a connection will established to a designated temperature sensor in the test environment of the SEPP project. For students which already have experience with Wireshark or for interested ones, the traffic gets recorded and exported for inspection with Wireshark. Students document their findings and identify potential privacy or security issues, such as unencrypted data. As an alternative or extension, students may extract Bluetooth logs directly from the Android smartphone in the SEPP suitcase using the HCI snoop function.

Exercise 3 of This Chapter: BLE Information Leak and DoS Attack

This exercise is still under development and will be tested in a future version of the SEPP project.

In this exercise, students analyze the Bluetooth communication of two different Nuki Smart Locks using the BLE protocol. For this purpose, the students should split up in two teams. The goal is to extract device-specific information such as serial number, firmware revision, and model number using Python scripts based on the bleak library. Students first identify the target devices using the nRF Connect for Mobile app and then extend a given script to query readable GATT characteristics.

In the second part, students modify a preconfigured Python script to simulate a DoS attack. The impact of the attack is monitored through the smartphone app, which is connected to the lock. As a final step, students are encouraged to compare their results with the other team. Since the two Locks are from different revisions, only the older one should loose the connection to the App by the DoS attack.

Course Chapter 7: Other Protocols in IoT

This chapter provides a short overview of less commonly discussed protocols that are still relevant in the context of IoT security. Students are introduced to ZigBee, Z-Wave, DECT, LoRaWAN, MQTT and Matter. Each protocol is briefly described with regard to its technical features and security implications. Examples include fallback keys and key distribution issues in ZigBee, weak encryption in early DECT implementations, and (disabled by default) TLS encryption in MQTT. Students are encouraged to document any of these protocols if they appear in their assigned device and to investigate vulnerabilities for their essay.

Exercise of This Chapter: MQTT Service Traffic Analysis

This exercise is still under development and will be tested in a future version of the SEPP project.

Course Chapter 8: Communication and Protocol Security

Chapter 8 focuses on the security of communication protocols used in IoT environments. Students are introduced to TLS and its predecessor SSL, along with key security features such as symmetric encryption, public-key cryptography, and message authentication. The TLS handshake and Public Key Infrastructure (PKI) are explained to show how authenticity is

ensured in encrypted communication. Datagram TLS (DTLS) is also covered as a variant optimized for UDP-based connections, which are common in IoT. Technical considerations such as handshake size and certificate optimization are briefly addressed.

In the second part, students explore how to analyze network traffic using Wireshark. This includes capturing, filtering and interpreting communication between IoT devices in a network. If possible, students are encouraged to perform basic Man-in-the-Middle attacks as well as data traffic analysis with Wireshark as part of their essay. The tool sslstrip is briefly introduced to demonstrate how HTTPS connections can be redirected to HTTP.

Exercise 1 of This Chapter: Replay Attack on an IoT Smart Bulb

This exercise was also already tested in the progress of the first SEPP paper and was also directly integrated into the course, since the results of various tests were consistently positive. The goal is, to simulate a replay attack on a smart light bulb (Yeelight) out of the SEPP suitcase using multiple techniques introduced in previous chapters. Students first perform a network scan using nmap to locate the device and identify an open TCP port. A connection is then established using telnet, through which control commands in JSON format are manually sent to switch the bulb on and off. The associated network traffic is recorded using Wireshark, allowing students to analyze whether the transmitted data is encrypted or readable in plaintext.

In the final part of the exercise, students connect to the platform's Raspberry Pi via ssh and capture live traffic using tcpdump. The resulting .pcap file is transferred to the local system and used as a basis for scripting a replay attack in Python with the help of Scapy. A provided script must be completed by inserting the correct parameters and the replay payload. When executed, the script reproduces the captured control command, successfully manipulating the device without user interaction. The exercise highlights both, the impact of unencrypted local traffic and the importance of authentication at the application layer.

Exercise 2 of This Chapter: SSL Stripping

This exercise is still under development and will be tested in a future version of the SEPP project.

Course Chapter 9: Web Interfaces and Firmware

For this chapter, students are introduced to the OWASP Zed Attack Proxy (ZAP), an open-source tool for analyzing and testing web applications. ZAP is used to detect common web vulnerabilities by analyzing and manipulating HTTP traffic between client and server. The course demonstrates how web-interfaces or configuration interfaces can be analyzed for weaknesses using ZAP. Afterwards, students are encouraged to check whether their assigned device offers a web-interface and to document any relevant findings in their essay.

The second part of the chapter deals with firmware security and reverse engineering. Students learn about common update methods (e.g. manual vs. automatic), the importance of signature validation, and the risks of fake or downgraded firmware. If the firmware is publicly accessible, it can be downloaded and analyzed using Ghidra, an open-source reverse engineering tool developed by the NSA. The course emphasizes that most firmware is not open source, making transparency and verification more difficult. Students are asked to evaluate how firmware

updates are handled on their assigned device and to identify potential weak points in the update process.

Exercise of This Chapter: Siemens Logo!

Web Interface Hacking. This exercise is still under development and will be tested in a future version of the SEPP project.

Conclusion and Outlook

The positive results from the initial SEPP evaluation, as documented in the first paper (Hauser et al., 2025), proved to be an important basis for the further development of the platform. Feedback from students as well as investigations while testing the first exercises helped identify which tools and techniques should also be integrated into the SEPP project. These insights helped to design new exercises, which were mostly covered in this paper. Also, most of the new exercises could be integrated into the according course to the project with only minor adjustments. The course structure provided a natural framework for aligning these tasks with specific learning objectives.

While some exercises have already been tested with students, others are still in development or awaiting practical validation. All exercises introduced in this paper, regardless of their current status, are scheduled to be evaluated in the upcoming semester. In this context, the focus will not only be on technical practicability and learning effectiveness but also on how the individual tasks fit into the overall concept of a modular, scalable teaching platform.

As a further step, selected exercises will also be tested by an AI system to explore whether automated methods can solve or support parts of the hands-on challenges. This approach is intended to provide insights into the potential of artificial intelligence in practical cybersecurity education as well as in hacking approaches. The results and will be documented in future iterations of the SEPP project.

References

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., & Zhou, Y. (2017). Understanding the Mirai botnet. In *Proceedings of the 26th USENIX Security Symposium* (pp. 1093–1110).

Chothia, T., & de Ruiter, J. (2016). Learning from others' mistakes: Penetration testing IoT devices in the classroom. In *ASE Workshop at the USENIX Security Symposium*. https://api.semanticscholar.org/CorpusID:217195674

European Commission. (2025). *Cyber resilience act*. https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act

Hauser, D., Graf, J., Fischer, S., & Hackenberg, R. (2025). SEPP: Security education and penetration-testing platform for IoT. In T*he European Conference on Education 2025: Official Conference Proceedings* (pp. 443–453). https://doi.org/10.22492/issn.2188-1162.2025.36

Junior, A. O., Funchal, G., Queiroz, J., Loureiro, J., Pedrosa, T., Parra, J., & Leitão, P. (2022). Learning cybersecurity in IoT-based applications through a capture the flag competition. In *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)* (pp. 560–565). https://doi.org/10.1109/INDIN51773.2022.9976079

Kolias, C., Stavrou, A., Voas, J., Bojanova, I., & Kuhn, R. (2016). Learning Internet-of-Things security hands-on. *IEEE Security & Privacy, 14*(1), 37–46. https://doi.org/10.1109/MSP.2016.4

Legg, P., Higgs, T., Spruhan, P., White, J., & Johnson, I. (2021). Hacking an IoT home: New opportunities for cyber security education combining remote learning with cyber-physical systems. In *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)* (pp. 1–4). https://doi.org/10.1109/CyberSA52016.2021.9478251

OWASP. (n.d.). https://owasp.org

Sahrani, S., Saad, M. H. M., Mutalib, A. A., & Zaidel, D. N. A. (2024). Incorporating the Internet of Things (IoT) learning module into the smart building course. *Jurnal Kejuruteraan*. https://api.semanticscholar.org/CorpusID:269323964

Shodan. (n.d.). https://www.shodan.io

Zed Attack Proxy. (n.d.). https://www.zaproxy.org

Contact email: d.hauser93@icloud.com