SEPP: Security Education and Penetration-Testing Platform for IoT

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

The European Conference on Education 2025
Official Conference Proceedings

Abstract

The Internet of Things (IoT) is becoming a major part of our everyday lives, offering convenience and smarter solutions, but also bringing significant security challenges. While theoretical knowledge in IoT security is essential, studies have shown that practical content can be an essential part of internalizing understanding. To address this, we developed the Security Education and Penetration-Testing Platform (SEPP) as the practical component of an existing IoT security course at the Ostbayerische Technische Hochschule (OTH) Regensburg. SEPP uses real IoT devices like smart locks, cameras, and plugs, simulating a smart home environment to make learning interactive and engaging. Students can explore vulnerabilities, conduct penetration tests, and document their findings through structured exercises. By working on tasks like network scanning, analyzing data traffic, and simulating attacks, students gain a deeper understanding of IoT security risks. Initial tests show that this approach helps students apply their theoretical knowledge and significantly improve their practical skills. This paper explains how SEPP was built, the exercises it offers, and why it's an important step forward in teaching IoT security effectively. Furthermore, we aim to share the findings and tasks from this paper with other universities, providing them with a solid foundation to teach practical IoT security knowledge in their own courses.

Keywords: IoT, cyber security, hacking, education, SEPP

iafor
The International Academic Forum
www.iafor.org

Introduction

The Internet of Things (IoT) has transformed industries and daily life by connecting devices to enable smarter homes, businesses, and infrastructure. However, this rapid adoption has also introduced critical security risks, as IoT devices often lack of defense mechanisms against cyberattacks. Addressing these vulnerabilities is essential, particularly as IoT systems increasingly handle sensitive data like cameras or control critical processes.

Teaching IoT security to computer science students presents unique challenges. While theoretical instruction provides a foundational understanding, it often falls short of giving the students real-world examples in a practical way. This practice would enable the students to earn first experience in finding, analyzing and using vulnerabilities of IoT devices.

In response to this need, the Security Education and Penetration-Testing Platform (SEPP) was developed. This platform is equipped with various IoT devices, such as smart locks, cameras, plugs and light bulbs, offering students the opportunity to apply their theoretical knowledge. By discovering, documenting and testing device vulnerabilities, the SEPP project equips students with the necessary information to apply their theoretical knowledge and turn it into a practical skill. This paper deals with the establishment of the practical part (SEPP) in the IoT security course and describes the procedure in detail.

Background and Related Work

The dynamic and interconnected nature of IoT systems presents unique cybersecurity challenges. Addressing these requires educational approaches that go beyond theory, providing students with the opportunity to engage directly with the technologies these devices deliver like Bluetooth Low Energy, WLAN and USB.

Hands-on learning is especially effective in helping students grasp the complexities of IoT security. Research highlights that practical exercises focusing on vulnerabilities such as insecure communication and user tracking provide valuable insights into real-world risks (Kolias et al., 2016). These exercises allow students to see firsthand how security flaws manifest and how they can be mitigated.

Interactive and gamified approaches further enhance learning outcomes. Capture-The-Flag (CTF) competitions have been used to immerse students in cybersecurity scenarios, enabling them to analyze and exploit vulnerabilities in IoT systems (Legg et al., 2021). Such activities not only make learning engaging but also point out the importance of critical thinking.

Integrating IoT modules into technical courses has also proven effective in bridging the gap between theoretical and practical learning. For instance, incorporating IoT topics into a smart building curriculum gave students hands-on experience with real devices, significantly improving their understanding of IoT architectures and security principles (Sahrani et al., 2024).

Other studies have shown that practical engagement with commercial IoT devices provides a deeper understanding of real-world challenges. By conducting penetration tests on these devices, students not only identify vulnerabilities but also learn to document and communicate their findings effectively (Chothia & de Ruiter, 2016). This approach ensures that students are better prepared to address the evolving landscape of cybersecurity threats.

Hands-on oriented approaches are highly effective in IoT security education. By focusing on real-world scenarios, such as identifying hardware vulnerabilities, analyzing communication protocols, and addressing web application security flaws, students gain direct, practical experience with IoT systems. Such methods have been shown to significantly enhance understanding and technical skill, with studies reporting that 96% of participants improved their grasp of IoT cybersecurity concepts through structured, practical exercises (Oliveira, Jr. et al., 2022).

These findings underscore the critical role of hands-on, interactive learning in IoT security education. Practical experiences not only solidify theoretical knowledge but also prepare students to tackle the unique challenges posed by IoT ecosystems. Building on these insights, the SEPP platform integrates these proven strategies to deliver a comprehensive and immersive learning environment, ensuring students are equipped to meet the demands of modern cybersecurity.

Simulating a Smart Home With Physical Devices

Although research has shown that there are already various practical approaches to IoT security courses, most of them do not use a wider range of physical devices, which are continuously being developed to teach practical skills. To bridge this gap, the SEPP platform uses realistic devices that are already in use in many households.

Devices like light bulbs, plugs, smart locks and cameras are the top-sellers at big online stores like Amazon. For this reason, these devices are already integrated into the SEPP platform.

Handling these devices is fairly easy and most of the students do not even need any explanation to use them. But to make the platform even more intuitive, the hacking exercises will be designed similar to the use of a smart home.

To implement this, the devices get separated by two criteria: the technology they use (for example Bluetooth Low Energy) and the difficulty to hack them. The latter in particular makes it possible to design exercises that are beginner-friendly but become increasingly difficult as the course progresses while using different technologies.

In the later progress of the project, the platform is planned to be designed like an actual house. This approach aims to help students to become more deeply engaged in the given task by visualizing it. In addition, this also allows the project to implement a gamified approach.

For example, the exercises of the course could be designed to firstly challenge the students to determine if the WLAN of the house is secured and identify devices within the network. The following exercise might involve shutting off a smart plug to disable all devices connected to it.

The simulation of real-world examples within a smart home environment should allow the students to interact with the given material in a more interactive and practical way.

Preparation of the Platform

To realize the described platform in such a way that it functions smoothly and is immediately ready for use by students, a lot of preparation is required.

As this platform focuses on IoT devices, these had to be selected first. As almost no manufacturer discloses the existing vulnerabilities of a device and there is little information on the Internet about the specific devices, we carried out the security tests ourselves.

We decided to implement the following hardware to the platform for operating reasons:

- Ubiquiti Edge Router X. This router is used to pass Internet to the Raspberry Pi and to maintain various security settings for the university's main network.
- Raspberry Pi 4 Model B with Raspberry Pi OS (formerly Raspbian). This Raspberry acts as a router using RaspAP. This step is necessary to create your own network for the platform. This makes it easier to identify the relevant IoT devices and to avoid problems with networks that are actually in use.
- GreenNet Trendnet Switch TEG-S50g. The switch is used between the Ubiquiti Edge Router and the Raspberry Pi to connect these devices and to offer the possibility of integrating further LAN devices into the network.
- Microsoft Surface. This laptop is integrated into the platform to have easy access to the devices, configurations and the setup. In addition to Microsoft Windows 10 as the main operating system, a virtual machine for Kali Linux and Android are also installed.

In addition, we have currently integrated the following IoT devices into the platform to perform tasks or carry out further vulnerability checks:

- EIGHTREE Wifi Smart Plug 16A
- Nuki Smart Lock 3.0 Pro
- Wiko Smartphone with Android 6
- LurcarLE Cloud Wifi HD Video Camera
- Luminea WLAN Water Detector
- Pamalar Wifi Door / Window Sensor
- Yeelight (Xiaomi) S1 Smart Bulb
- Lepro LE A60 Smart Bulb
- TP-Link Tapo L510E Smart Bulb
- Inkbird Smart Air Quality Monitor
- Itius IR Motion Detector

Most of the devices can be updated, but since the users mostly do not want to pay for updates, the companies will not deliver free updates for a long period of time which results in various vulnerabilities. However, if updates are available, we do not recommend carrying them out in order to keep any vulnerabilities open within the framework of the platform.

Configuration

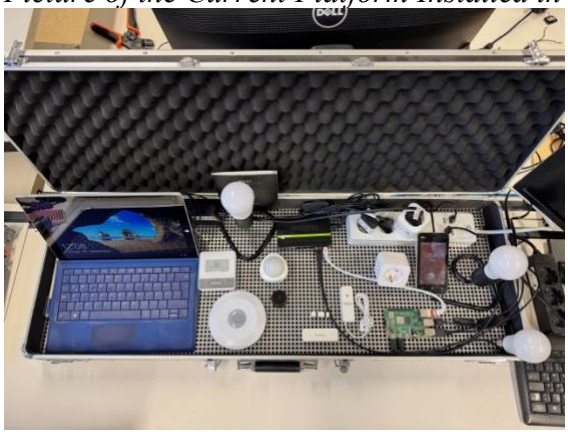
To configure all devices within the platform, we first set up the mentioned network using the Raspberry Pi and RaspAP. Then we connected the mentioned devices to the Raspberry Pi via WLAN or Bluetooth. This should be done step by step and the successful connection should then be checked individually for each device.

After all devices were successfully integrated into the platform, we started to look for vulnerabilities for each individual device to create tasks for the students from the findings.

To ensure a certain portability of the platform and an overview of it, we decided to temporarily install the entire configuration in a case.

Figure 1

Picture of the Current Platform Installed in a Case



Example Exercises

In this section, we provide a set of standalone tasks designed to help students explore and address key IoT security challenges. Rather than prescribing a fixed exercise sequence, these tasks allow users to flexibly combine and adapt them into exercises that best suit their learning objectives. In addition, these tasks cover key aspects of IoT security and enable practical exploration of vulnerabilities and network behavior.

Objective

By successfully attempting all given tasks, the students will be able to:

- Identify IoT devices within a local network
- Perform network scans to discover open TCP and UDP ports
- Analyze exchanged packages within the network
- Simulate a DoS attack by using SYN-flooding on an IoT device
- Controlling certain devices using terminal commands
- Perform Replay attacks using Python

Preparation

To find and make use of the given vulnerabilities, the following tools and devices were used within the tasks:

- Nmap - command line tool to scan a given network
- netcat - command line tool to send data through the network
- Wireshark - application to trace packages in the network traffic
- Scapy - python package to send, receive and analyze network packages
- telnet - command line tool to communicate with devices and computers

Tasks

The following tasks can be flexibly combined and are ideal for creating step-by-step instructions.

Network Scanning Using the EIGHTREE Smart Plug

Students start by finding out the relevant IP address range using either the ifconfig command for Linux and Mac operating systems or ipconfig for Windows. After accessing this necessary information, the students move on by identifying the IP address of the IoT devices and performing network scans using nmap -sn <IP address> range. At this point, either the name resolution of the desired device should be activated or the specific MAC address should be made available to the students. Subsequently, the scan results are used to list all open TCP and UDP ports for the given IP adress of the device (in our case, the EIGHTREE Smart Plug) by using nmap -sT -p <IP adress> for TCP ports and nmap -sU -p <IP adress> for UDP ports. This step provides fundamental knowledge to understand the communication of the device and recognize potential vulnerabilities.

Simulating a DoS Attack With the EIGHTREE Smart Plug

Using a partially completed Python script, students implement a SYN-flood attack on the IoT device. The script uses the Scapy Python library to generate and send a large number of spoofed SYN packets in a simple while True: loop. To do this, the script only needs the IP address of the device to be attacked and the corresponding open port. The relevant functions are the following:

```
Python:
ip = IP(src=RandIP(), dst=target_ip)

tcp = TCP(sport=RandShort(),
    dport=target_port, flags="S")
    #S for SYN package

pack = ip/tcp #Combines the ip and tcp package
send(pack)
```

The result will overwhelm the communication capabilities of the plug. Students observe the impact of the attack in real time, including its effect on the device's responsiveness and functionality using Wireshark. This task demonstrates how a DoS attack can disrupt an IoT device.

Analyse Network Traffic Using Wireshark

During the SYN-flood attack, students use Wireshark to monitor network traffic. They apply filters for the relevant IP adress by using ip.dst == <IP adress> for the destination IP and ip.src == <IP adress> to focus on packets sent to and from the IoT device, analyze the patterns of SYN and SYN-ACK packets, and observe how the attack exploits the device's communication protocols.

Investigating Packet Encryption

Students test the IoT device's communication security by sending custom UDP packets using netcat. By capturing and analyzing these packets in Wireshark, they determine whether the communication of the device is encrypted or transmitted in plain text. This task highlights potential risks of unencrypted data transmission.

Packet Analysis of App Commands

Using a smartphone app to control the the IoT device (for example, the EIGHTREE Smart Plug), students capture and analyze packets exchanged between the app and the IoT device within Wireshark. The captured traffic gets reviewed for encryption and replay attack vulnerabilities again using Wireshark.

This task allows students to consolidate their theoretical knowledge of network packets and encryption through a practical insight.

Controlling the Yeelight S1 Smart Bulb via Telnet

For this task, the students need to find the device and the open port(s) within the network using the described commands with nmap. After finding this necessary information, the packages send to the device using a device-specific application can be traced using Wireshark. If the packages sent to the devices are unencrypted, the students should be able to extract the commands to control the device. In the case of the Yeelight S1 Smart Bulb, the device can be turned on and off sending the following JSON commands directly through telnet:

```
Python:
telnet <IP adress> <Port> #connects to the device

{"id":0, "method": "set_power", "params":
   ["on", "smooth", 200]} #turns bulb on

{"id":0, "method": "set_power", "params":
   ["off", "smooth", 200]} #turns bulb off
```

Packages sent through this method can also be traced and reviewed using Wireshark. The method shown also provides the basic knowledge required for replay attacks.

Replay Attack Using Python and Scapy on the Yeelight S1 Smart Bulb

A replay attack works by capturing legitimate data packets and sending them again to trick a device into performing unintended actions. Using Python and the Scapy library makes it easy to simulate such attacks, as the library allows creating and transmitting a variety of packages. This provides a hands-on way to understand the mechanics and risks of replay attacks.

In order to carry out such an attack, it is necessary for the students to extract the correct commands from the package traffic beforehand. In the case of the Yeelight S1 Smart Bulb, we can use the JSON commands already shown in conjunction with Python.

The following Python script can perform a replay attack on the bulb:

```
Python:
from scapy.all import *
from scapy.layers.inet import IP, TCP

#IP and port of the lamp
ip = "XXX"
port = XXX

json_command = 'XXX' + '\r\n'
# \r\n required to terminate the message

# 3-Way Handshake: Initiate TCP session
syn = IP(dst=ip) / TCP(dport=port, flags="S")
syn_ack = sr1(syn)

# 3-Way Handshake: ACK for the handshake
ack = IP(dst=ip) / TCP(dport=port,
    sport=syn_ack[TCP].dport, flags="A",
    seq=syn_ack.ack, ack=syn_ack.seq + 1)
send(ack)

#Send the JSON command
push_ack = IP(dst=ip) / TCP(dport=port,
        sport=syn_ack[TCP].dport,
        flags="PA", seq=syn_ack.ack,
        ack=syn_ack.seq + 1) / json_command
send(push_ack)

#Close the TCP session
fin = IP(dst=ip) / TCP(dport=port,
    sport=syn_ack[TCP].dport, flags="FA",
    seq=syn_ack.ack + len(json_command),
    ack=syn_ack.seq + 1)
fin_ack = sr1(fin)

ack = IP(dst=ip) / TCP(dport=port,
    sport=syn_ack[TCP].dport, flags="A",
    seq=fin_ack.ack, ack=fin_ack.seq + 1)
```

After successfully performing the replay attack shown, students should have a deeper understanding of the relevance of vulnerabilities and encryption as well as Python in conjunction with the Scapy package.

Evaluation

To evaluate the tasks shown, we bundled them into two exercises and carried each out with a test group of nine students. For both exercises, rough instructions were created, which contained gap scripts for Python and instructions on when to use nmap, for example. The

students came from an IoT security course, for which the practical content is planned for the future. Accordingly, the students have already gained theoretical knowledge on the topics shown.

In the first exercise, the students had to use nmap to find the IP address and the open port of the EIGHTREE Smart Plug. The data traffic was then observed and analyzed using Wireshark. Finally, the Python commands were used to simulate a DoS attack and the data traffic was observed again using Wireshark.

For the second exercise, the described task was chosen to control via telnet and perform the replay using Python attack on the Yeelight S1 Smart Bulb. Here, too, the relevant information such as IP address, port and commands had to be determined first.

The results of the two test groups showed that all tasks were successfully completed with the instructions. As a result, the students were able to apply their theoretical knowledge in practice and significantly deepen it.

The bundling of some tasks into exercise sheets also made it possible to cover some relevant areas such as nmap, netcat, telnet, Scapy/Python and Wireshark at the same time.

Unfortunately, we noticed that there were problems with the EIGHTREE Smart Plug from the first exercise. When too many participants tried to access it at the same time, an unintentional DoS occurred and the plug refused to work. This could also occur with other IoT devices and must be observed in subsequent exercises. One solution would be, for example, to limit the number of participants per device and set up additional platforms.

Perspectives of Intrusion and Response

In the training of security experts, it is becoming increasingly important to explain defense strategies on the basis of attacks. This is especially true as attackers and defenders are in a constant race in practice. Attackers exploit vulnerabilities in systems and processes to compromise data or impair services and much more. In contrast, IT security experts in the defense sector, also known as the blue team, pursue the goal of ensuring the integrity, confidentiality and availability of information. Due to the ever-growing complexity of modern IT infrastructures, coordinated and holistic approaches to security are becoming increasingly important. An essential component of this is monitoring, detection and damage limitation in the event of an attack. Understanding the cause and effect of attacks from the perspective of system protectors and using the knowledge gained to integrate and evaluate prevention strategies and measures offers an efficient way of combining correlated topics.

Conclusion and Outlooks

In conclusion, this evaluation underlines the relevance of practical exercises to theoretical content, especially in the field of IoT security. However, such practical exercises must be well prepared and continuously improved to ensure a seamless process for the students.

Further improvements are necessary to resolve problems such as unintentional DoS attacks. One way of solving this could be to build additional platforms, for example. This attempt would allow us to limit the number of participants per platform and also allow students to work in groups, which could increase the learning effect.

In addition, other devices have already been purchased, such as programmable USB sticks, Arduinos and some Micro:Bit's. In the future, we want to use these devices to demonstrate even more ways in which vulnerabilities can be exploited (e.g. bad USB). The Aurduino and Micro:Bit devices could also be used to create an interface between IoT devices for control or to simulate IoT devices.

Devices like the smart lock, a camera and a variety of sensors (air, light, moisture) are already integrated into the platform, which are currently being checked for security gaps to create further tasks.

Testing and adding additional devices should also cover more relevant areas and could offer tasks like:

- Bluetooth / Bluetooth Low-Energy hacking
- Man in the Middle attacks
- Bad USB
- Brute Force
- Injections
- Insecure Webserver

Through the ongoing integration of further devices and tasks, the existing platform should provide an even more comprehensive insight into existing vulnerabilities. The aim is to cover as many technologies as possible, especially as more modern devices often have more than one interface and the security risk is therefore constantly increasing.

The future inclusion of defense measures to monitor, detect and prevent self-perpetrated attacks holds great potential in the training of CS experts. By gamifying the techniques of both the attacker and the defender, cause and effect as well as advanced attack and prevention techniques can be taught in an understandable and hands-on way.

Due to the rapid and continuous development of new IoT devices, it will become increasingly important to keep a critical eye on the security vulnerabilities of these very devices in the future. With this work-in-progress paper, we will try to keep up to date with these developments and include or investigate both old and new vulnerabilities.

References

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

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

Oliveira Jr., A., Funchal, G., Queiroz, J., Loureiro, J., Pedrosa, T., Parra, J., & Leitao, 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

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

Contact email: d.hauser93@icloud.com