Improving Student Learning and Engagement in Cybersecurity Through Designing and Building Secure Internet of Things (IoT) Systems

Dr. Saeed Al-Haj, Ohio Northern University

Dr. Saeed Al-Haj, PhD., is an Assistant Professor of Computer Science at Ohio Northern University, Ada, Ohio. He completed his Ph.D. in Computing and Informatics from the University of North Carolina Charlotte. His expertise and general interests include: Computer and Network Security; Security Analytics; Firewalls and Access Control Configuration Analytics; Computer Science Education and Cybersecurity Education. His teaching experiences include teaching Computer Science courses and labs, utilizing technology to maximize student learning process, developing curriculum and labs, and supervising undergraduate students projects.

Improving Student Learning and Engagement in Cybersecurity Through Designing and Building Secure Internet of Things (IoT) Systems

Cybersecurity education aims to bring the awareness of the importance of security and privacy issues to students. This will help students change how they think when they develop and implement computer applications to consider security problems while they design and test their products.

As our life these days depends heavily on Internet and the usage of computer and mobile applications, the need for cybersecurity professional and experts will continue to expand. Therefore, graduating students who have proper cybersecurity instruction becomes a necessity. This can be achieved by incorporating modern security analysis tools and engaging students in building secure systems throughout the undergraduate curriculum. The primary goals are: 1) to have more systems and products with fewer exploits and vulnerabilities, and 2) to increase the number of professional individuals who are interested in cybersecurity careers and have the proper cybersecurity knowledge and training.

One key challenge in implementing and designing cybersecurity exercises in classrooms is having the proper infrastructure that allows conducting experiments in an isolated environment without jeopardizing security. To overcome this challenge, this paper presents "Secure-It-Yourself" kit. The kit utilizes Raspberry Pi unit as a development environment for hands-on activities and exercises. The advantage of using Raspberry Pi units is creating a configurable isolated sandbox for security hands-on exercises without affecting the current lab infrastructure.

The corner stone in cybersecurity education is offering a set of engaging projects and exercises by which the students get hands-on experience to reinforce concepts covered in the classroom. This paper focuses on assembling a "Secure-It-Yourself" kit for students that can be utilized to bring the awareness of secure systems while enjoying building "Do-It-Yourself" projects. The kit includes all needed components and instructions to build small Internet of Things (IoT) systems.

The "Secure-It-Yourself" kit gives instructors the needed flexibility to design activities that explain how security concepts and secure design principles should be used to build and implement secure systems. Also, this kit helps students to acquire the recommended National Initiative Cybersecurity Education (NICE) [1] security Knowledge Skills and Abilities (KSAs) relevant to the scope of this paper.

The kit can be used in undergraduate Security courses and in summer camps curriculums designed for high school students. The following components are included in the kit: Raspberry Pi unit, motion detection sensor, camera, breadboard, bush button switches, LEDs, jumper wires, and buzzers. The instructor's kit includes additional components such as: RFID readers and electric locks that support fail-safe and fail-secure features. The paper discusses how the kit is used in two security courses to develop projects and exercises that enhance students' security knowledge and skills.

Introduction

Computer security courses teaches several security concepts and terms such as: confidentiality, integrity, availability, authentication, authorization, encryption, hashing, digital signature, etc. Teaching these concepts and terms requires providing hands-on exercises and labs. The main challenge in offering security courses is having the right platform to conduct security exercises and labs in a contained environment that does not jeopardize the security of university's infrastructure. One solution to this problem is using cloud based solutions such as Ohio Cyber Range [2]. Cloud based solutions provide unlimited isolated resources that can replaced traditional security labs. However, cloud based environments do not help in designing and implementing IoT systems such as smart home systems. This paper introduces "Secure-It-Yourself", a kit for students and instructors that can be utilized to teach security concepts and secure design principles while enjoying building "Do-It-Yourself" IoT projects.

Security must be designed and not attached. To emphasize this concept, students need to work on designing systems from scratch to understand the importance of building secure designs instead of attaching security features later after building the system. In the absence of security design and implementation techniques that systematically exclude security flaws, it is useful to have a set of secure design principles that can guide developing secure systems [3].

Secure design principles are essential security principles that are taught in computer security courses. Most computer security books cover these principles, and they provide scenarios explaining how these principles are applied. The secure design principles as discussed in [3] and [4] are listed along with definitions as follows:

- *Economy of mechanism*: The design of security measures embodied in both hardware and software should be as simple and small as possible.
- *Fail-safe defaults*: Access decisions should be based on permission rather than exclusion.
- *Complete mediation*: Every access must be checked against the access control mechanism.
- *Open design*: The design of a security mechanism should be open rather than secret.
- *Separation of privilege*: A practice in which multiple privilege attributes are required to achieve access to a restricted resource.
- *Least privilege*: Every process and every user of the system should operate using the least set of privileges necessary to perform the task.
- *Least common mechanism*: The design should minimize the functions shared by different users, providing mutual security
- *Psychological Acceptability*: The security mechanisms should not interfere unduly with the work of users, while at the same time meeting the needs of those who authorize access. If a security mechanism hinders the usability of resources, then users may opt to turn off this mechanism.
- *Least astonishment*: A program or user interface should always respond in the way that is least likely to astonish the user.

Most computer security books cover these principles, but they lack to connect these principles to a specific design project in which the students have the opportunity understand the effect of every principle on building a secure system. Allowing students to incorporate secure design principle in a project of their choice provides a unique unforgettable experience that will last for future implementations.

To ensure proper cybersecurity coverage of the developed materials, the following CS2013 [5] Information Assurance and Security (IAS) knowledge units are be covered:

- Foundational concepts in security: Confidentiality, Integrity, and Availability
- Principles of Secure Design
- Defensive Programming
- Threats and Attacks
- Network Security
- Cryptography

Covering these knowledge units enhances cybersecurity pedagogy by directly impacting the way security courses and labs are taught.

Related Work

The work in this paper focuses on providing flexible environment to conduct security related projects while focusing on teaching security concepts and secure design principles.

The work in [6] discusses using IoT in cybersecurity education. The course: "Resilience of Internet of Things and Cyber-Physical Systems" integrates education, research and regional development together. It shows how industry-university collaboration can be organized to enhance cybersecurity education. The work in [7] examines the NIST SP 800-160 systems security strategies and design principles, and offers a mapping of conceptual strategies to concrete security principles that can be more effectively designed-for, built-in, and tested. The work in [8] describes a course in Secure Systems that uses the Flaw Hypothesis Methodology for penetration testing as a vehicle for motivating and teaching students fundamental principles of security engineering.

Our work is different by using IoT kits as a vehicle to build secure systems while emphasizing on teaching the secure design principles.

Approach

This section provides details on how the kit is used to achieve the learning outcomes and objectives related to designing and building secure IoT systems.

"Secure-It-Yourself" Kit:

At the beginning of the semester, "Secure-It-Yourself" were distributed to the students so they can familiarize themselves with its components. At the end of the semester, students are required to turn in the kits. Instructors need to have extra backup kits and extra components.

The students' kit includes the following:

- Raspberry Pi unit, 3B+ model or 4B model
- PIR motion detection sensor
- Raspberry Pi camera module
- Magnetic door switches
- GPIO matrix keypad
- Breadboard, bush button switches, variety of LEDs, resistors, jumper wires, and buzzers.

Other sensors and components are given to students based on their need in their projects such as temperature and humidity sensors, digital displays, etc.

The instructor's kit includes additional components such as:

- RFID readers and tokens
- Multichannel relay switches
- USB Battery pack for Raspberry Pi
- Electric locks that support fail-safe and fail-secure features

The kit is used in two undergraduate computer security courses, one introductory course and one advanced course. It is also used for in-class activities in lab setting classes and used for individual/group students project that were built outside the classroom. The kit can be expanded to include more components depending on the curriculum and what the instructor wants from his students to develop and implement in their projects.

Multi-Staged IoT Project:

A multi-staged IoT project that requires students to implement a secure system that can be used for home automation is utilized to teach security concepts and the secure design principles. The project is built incrementally over several stages allowing students to rethink their designs and focus on applying secure design principles.

The project helps CS students to get hands-on experience in building a small system using sensors, LEDs, resistors, transistors, etc. CS students usually lake hardware skills when dealing with small electric components; this project helps them to make a solid connection between hardware and software implementations.

The project is divided into several stages as follows:

- *Stage 0*: Opportunity recognition and exploration

  In this stage, students will explore home automation and smart homes systems looking for possible project ideas that can be implemented to convert typical homes into smart homes. Students are required to analyze their ideas and assess the security level of the implemented project and what possible threats and vulnerabilities could be introduced to the home system.

  Security Concepts learned in this stage: Confidentiality, Integrity, Availability, Privacy, Threats and Attacks.

Secure Design Principles learned in this stage: Students in this stage explore different ideas to be implemented. There are two major secure designed principles that can be learned here: 1) economy of mechanism and 2) open design. Economy of mechanism principle focuses on simplicity, the design should be as simple and small as possible. The rational behind this principle is that small and simple systems are easier to test and verify. Complex systems have more opportunities for weaknesses and vulnerabilities from components interactions. Open design principle encourages reusing open system that can be tested and verified by anyone instead of building systems from scratch without being tested by many experts. A good example is using known and tested encryption algorithms that are publicly available to everyone instead of creating their own secret encryption algorithm.

- *Stage 1*: Building the physical system

  In this stage, students will use Raspberry Pi to build a small physical device that can be used in home automation and smart homes. For example, they can control a garage door using motion detection sensors and push buttons. After building the system, students are asked the following questions: Can you open the door remotely using your smartphone? What do you need to add this feature?

  Security Concepts learned in this stage: Physical Security.

  Secure Design Principles learned in this stage: In addition to economy of mechanism and open design principles that can be applied in this stage as well, two more secure design principles can be discussed in this stage: 1) fail-safe defaults and 2) least astonishments. Fail-safe default principle in this stage could be explained by asking the following question: What should the system do if the Raspberry Pi unit is not running (i.e. electricity outage)? The default scenario is a closed door not an opened door. The least astonishments principles means that the system should work as expected with no surprises. If there are two colored buttons to control the door, green button is expected to open the door and the red button to close it; otherwise, users will be confused.

- *Stage 2*: Providing remote access to the system

  In this stage, students will control the system built in stage 1 by accessing a webpage. Students will design and create a webpage, add all needed controls to the webpage, and convert the Raspberry Pi into a light web server. Now the garage door can be opened physically and remotely. After finishing this stage, students are asked the following questions: Who can access this webpage? Is it secure? How can you fix the problem?

  Security Concepts learned in this stage: Availability, Network Security, Multi-Factor Authentication.

  Secure Design Principles learned in this stage: All secure design principles discussed in the previous stages apply in this stage. Also, least privilege principle can be discussed when building the webpage that allows remote access. The webpage should have the needed resources to open the door, it should not provide means to control the Raspberry Pi unit. Having more privileges than needed increases the attack surface.

- *Stage 3*: Securing the web entry to the system

  In this stage, students will add a security layer to the system by requiring a password to access the webpage created in Stage 2. Students will implement a way to store the password in the system. After finishing this stage, students are asked the following questions: Is your system secure against brute-force attack? Can one find the stored password in the Raspberry Pi? What do you need to secure the system?

  Security Concepts learned in this stage: Access Control, Confidentiality, Integrity, Availability, Authentication, Authorization.

  Secure Design Principles learned in this stage: This stage provides a great opportunity to introduce two new secure design principles: 1) complete mediation and 2) psychological acceptability. Complete mediation principle means that the user must enter the password every time he accesses the webpage. The password must not be cashed or stored in the webpage. The rational behind this principle is simple, what happens if a user is no longer allowed to enter the home? If the user had accessed the webpage recently before revoking the privileges, he still can open the door because of the cased password and he is not required to provide it. Psychological acceptability principle fits nicely here, passwords need to selected by users not the system. If the system generates the passwords it will be hard for users to remember them, this will force them to write down their passwords to be able to use later. Users need to change their passwords frequently, if the system forces users to change their passwords very often (every week for example); users may opt out to use the security mechanism.
  Fail-safe defaults principle fits perfectly when talking about passwords, what should the system do if unauthorized user accessed the system?

- *Stage 4*: Securing the system against brute-force attack

  In this stage, students will revise their implementation from the previous stage and store hashed passwords instead of plain-text passwords. Also, they will implement a lockout mechanism that limits how many wrong passwords are allowed. When locked out, a security code is needed to activate the system, this security code should be sent to the user's mobile phone as a text message and should expire after a period of time.

  Security Concepts learned in this stage: Encryption, Hashing, Network Security, Defensive Programming, Multi-Factor Authentication.

  Secure Design Principles learned in this stage: In this stage, two secure design principles can be discussed in addition to the principles applied in the previous stages. The two principles are: 1) separation of privilege and 2) least common mechanism. When a user is locked out after exceeding the limit of possible passwords, he can get a temporary password using the registered email or the phone number stored in user's profile. If a user wants to change the password, she must answer some challenge questions first. These questions were answered previously when the profile is created, challenges response must match the stored answers before giving the permission to change the password. These scenarios are examples of applying separation of privilege principle. When multiple users are interacting

with system, the least common mechanism principle requires minimizing the shared resources among different users.

In summary, this multi-staged project forced students to rethink their designs many times considering the security level of the system. It showed them the importance of building a secure system at the design stage. IoT and computer security cannot be an afterthought, it must be the foundation of design.
Additional stages can be designed and integrated to the current project depending on the course the knowledge level of students in the class.

Results and Discussion

To use this Raspberry Pi kit in classes, instructors need to be aware of students' knowledge level on Python programming and using GPIOs in Raspberry Pi. It is noticed that CS students lack some hardware skills in general, while engineering students lack some Python programming knowledge.

It is suggested to provide two or three simple tutorials on how to use breadboard and connect multiple components to Raspberry Pi GPIOs. In my introductory computer security class, students had little experience with Raspberry Pi hardware part and the Python programming part. Providing step by step instructions will help students to overcome the barrier of using the Raspberry Pi for the first time.

It is also helpful to provide students with some online tutorials to read and watch before they come to the class. This active learning approach will reduce the time needed to understand the basics such as how LEDs are used and what is a resistor? This also will help in bridging the gap between CS students who lack some hardware knowledge and engineering students who are familiar with the kit components.

Students in the introductory computer security course performed better on tasks that have clear deliverables and expectations. In open-ended design projects, students need more guidelines to help them move from one stage to another stage in the project. Senior students in advanced security course performed better in open-ended design project. This is expected because they have more knowledge about the design process and they are working on their capstone projects.

Students' Projects:

The kit was used in two courses over several semesters, 26 students submitted the project. In one semester, students were asked to form teams and work on ONE given idea; implementing an IoT surveillance system for smart homes. The goal was to see different project with different requirements for the same proposed idea. This helped students to realize that the same idea could have several designs, and every design has its own requirements and limitations. At the end, students discussed the presented designs in terms of the introduced threats and vulnerabilities, ramifications of the selected designs, and secure design principles that were implemented or neglected and the effect of the design principles on the overall design.

Table 1: Acquired Technical Skills Before and After Completing the Project.

Foundational Concepts	Acquired Technical Skills	
	Before	After
Confidentiality	58%	85%
Integrity	53%	92%
Availability	69%	96%

Table 2: Secure Design Principles Implementation Throughout the Project

Secure Design Principles	Project Stages				
	Stage 0	Stage 1	Stage 2	Stage 3	Stage 4
Economy of Mechanism	100%	100%	100%	100%	100%
Open Design	100%	100%	100%	100%	100%
Fail-Safe Defaults	X	88%	96%	100%	100%
Least Astonishment	X	73%	100%	100%	100%
Least Privilege	X	X	84%	100%	100%
Complete Mediation	X	X	X	85%	100%
Psychological Acceptability.	X	X	X	92%	100%
Separation of Privilege	X	X	X	X	92%
Least Common Mechanism	X	X	X	X	96%

In other semesters, students were asked to work individually on their course project. Students were required to propose an IoT project idea related to enhancing the security of smart homes. Some of students proposed projects are listed below:

- Surveillance system that interacts with smartphone
- Surveillance system that provides secure cloud backup
- Smart door bell
- Smart controller for IoT devices connected in the same network
- Firewall to block any suspicious traffic from the inside IoT devices to the outside world

Table 1 shows the percentage of students who acquired technical skills by implementing at least one approach to achieve the foundational security concepts, Confidentiality, Integrity, and Integrity. After completing the project, students show improvement in using encryption and hashing tools in their design.

Table 2 shows the percentage of students who accurately implemented secure design principles in their design throughout different stages of the project. Some design principles were easier to implement than other design principles. The feedback given to students after each stage gave them the opportunity enhance their design and adopt the needed secure design principles in later stages. After finishing the final stage, Stage 4, the majority of students found a way to adopt secure design principles in their design.

This project explained why it is important to adopt secure design principles and gave students the opportunity to see how neglecting secure design principles introduces vulnerabilities that could be exploited by attackers. Students received the following message after finishing project: the security of a product must be considered in the design stage and not attached after implementation.

Overall, students had a positive experience. Many students did comment that they felt the kit helped them to understand how the concepts that had been presented in the class were applied in real systems using secure design principles. Because of the ownership of the project they proposed, students felt the responsibility to build secure systems.

Conclusion

Cybersecurity education requires having a proper platform to teach security concepts and topics related to threats, vulnerability and attacks. In general, this requires a dedicated isolated security lab to run students experiments. Cloud based solutions provide access to resources that can be utilized to run security experiments in contained environments. This paper presents "Secure-It-Yourself" kit that provides a portable platform to build secure IoT systems. Building secure systems requires having deep understanding of secure design principles and how to apply them in the design phase. The paper discusses how "Secure-It-Yourself" kit is used to teach students security concepts and applying secure design principles at the design phase.

The multi-staged project covers CS2013 Information Assurance and Security (IAS) knowledge units and helps students to acquire the recommended National Initiative Cybersecurity Education (NICE) security Knowledge Skills and Abilities (KSAs) relevant to the scope of this paper.

Using the kit helped CS students to acquire additional hardware skills, and helped engineering students to acquire additional programming skills. The kit does not require a specialized security lab, it can be used with any general computer lab. The raspberry Pi units work as standalone sandboxes. The flexibility of the kit allows students to select different possible topics for their projects, this is an essential part in inclusive teaching approach in which the course contents satisfy students' needs.

References

[1] NIST National Initiative for Cybersecurity Education (NICE). https://www.nist.gov/itl/applied-cybersecurity/nice

[2] Ohio Cyber Range. http://ohioc3.org/cyber-range

[3] William Stallings and Lawrie Brown, "Computer Security: Principles and Practice", 4th Edition, Pearson 2018, ISBN-13: 978-0134794105.

[4] Matt Bishop, "Computer Security Art and Science", 2nd Edition, Pearson 2019, ISBN-13: 978-0-321-71233-2.

[5] Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society

[6] J. Rajamäki, "Industry-university collaboration on IoT cyber security education: Academic course: "Resilience of Internet of Things and cyber-physical systems"" 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018.

[7] Mailloux, Logan & Beach, Paul & Span, Martin. "Examination of Security Design Principles from NIST SP 800-160". IEEE International Systems Conference (SYSCON) 2018.

[8] Irvine, C. and Levin, T. "Teaching Security Engineering Principles". Proceedings Second World Conference on Information Security Education, 2001.