Master's Thesis

Hands-on Training for Mitigating Web Application Vulnerabilities

Ngo Van Quyen

Supervisor: Assoc. Prof. Razvan Beuran

Graduate School of Information Science
Japan Advanced Institute of Science and Technology
(Information Science)

August 2023

Abstract

Web applications are becoming increasingly complex and interconnected, making them more vulnerable to attack. In 2022, web application attacks accounted for about 56% of all data breaches. This is due to the fact that web applications are often developed using frameworks that contain security vulnerabilities that are known to hackers.

One of the most popular frameworks to build web applications is the Yii2 PHP Framework. Yii2 is a free and open-source framework that is used by millions of developers worldwide. It is also a very secure framework, and it has been penetration tested by a number of security experts. However, there are still some vulnerabilities, such as those caused accidentally or unknowingly by developers, that need to be mitigated.

Mitigating web application vulnerabilities can be approached in several ways. One approach is to use automated tools to scan for vulnerabilities. However, these tools can only find known vulnerabilities, and they often miss new or zero-day vulnerabilities. Another approach is to use manual security testing. This involves having a security expert manually test the application for vulnerabilities. However, manual security testing is a time-consuming and expensive process. A third approach is to use hands-on training. This involves training developers on how to identify and fix vulnerabilities in web applications. Hands-on training should be used as a complement to automated tools or manual security testing, as it teaches developers secure development practices with an attacker perspective.

CyPROM is a scenario progression management system for cybersecurity training that allows instructors to define training scenarios and provide target information. The management module of CyPROM uses a set of processes to drive the execution of those scenarios in the training environment. This enables conducting hands-on training in which trainees can actively engage in simulated cyberattacks, forensic investigations, and defensive measures. Thus, they can actively participate in realistic cyber exercises, gaining practical experience in dealing with cybersecurity challenges. By simulating attack scenarios, trainees can develop skills in identifying vulnerabilities, detecting and responding to threats, and implementing defensive measures.

The research presented in this thesis is an endeavor focused on bolstering the security characteristics of web applications by conducting a meticulous analysis of the Yii2 framework while drawing upon the reputable OWASP Top Ten as a fundamental reference. The OWASP Top Ten, developed and

maintained by the Open Web Application Security Project (OWASP) Foundation, plays a crucial role in promoting awareness about the most critical security risks faced by web applications. By attentively examining the 2021 updates and trends outlined in the OWASP Top Ten, our research ensured a comprehensive approach to addressing the most pressing threats to web application security.

In particular, vulnerabilities within the Yii2 framework were identified and carefully evaluated for their potential impact on web applications, leveraging insights from the OWASP Top Ten. A significant contribution of our research lies in providing a thorough assessment of web applications built on Yii2, aligning the results with industry-accepted security standards, and offering effective strategies to enhance web application security. This was achieved by extending the functionality of CyPROM via a custom module specifically designed for analyzing web applications. Extending CyPROM required a careful understanding of the intricacies of the Yii2 framework and its underlying architecture. We conducted a thorough analysis of the framework's source code, libraries, and dependencies to identify potential areas vulnerable to security threats. The process required reverse engineering and static code analysis to gain comprehensive insights into the framework's security posture.

After gaining a comprehensive understanding of the Yii2 framework, the CyPROM extension development commenced, entailing an in-depth investigation that surpassed mere vulnerability identification. The focus extended to crafting a specialized hands-on training program using CyPROM, tailored explicitly to address the identified vulnerabilities, accompanied by a strategic approach to tackling each security concern. The training program provided a set of systematically implemented actions and scenarios, empowering web developers with practical knowledge to proficiently secure their web applications. This included rules and heuristics inspired by the OWASP Top Ten, targeting prevalent security issues commonly encountered in web applications. Through the incorporation of this extension, CyPROM facilitated automated security assessments, bolstering the security not only of Yii2-based web applications but also laying the groundwork for enhancing the security posture of other frameworks.

The extension of CyPROM was assessed comprehensively via functionality evaluation, comparative analysis of implemented actions and scenarios, and user evaluation. Overall the enhanced CyPROM demonstrated significant coverage in addressing specific vulnerabilities in web applications, with notable strengths in dealing with Broken Access Control (3.36 out of 5), Identification and Authentication Failures (3.27 out of 5), and Cryptographic Failures (3.18 out of 5). The evaluation result, determined based on

trainees' feedback obtained during training sessions, estimates the capability of the enhanced CyPROM to handle these critical categories. In addition, the assessment also revealed areas that need further improvement to increase overall effectiveness in addressing other vulnerabilities. Even so, participants appreciated the clarity and conciseness of the training, which enabled them to apply their newly acquired knowledge in a real-life environment. The positive feedback from users underscored the practical value and real-world relevance of our research findings. Efforts to increase coverage in areas where CyPROM is currently less effective will help improve its capabilities.

This research serves as a valuable resource for developers and security professionals, aiding them in fortifying the integrity of web applications by highlighting vulnerabilities, benchmarking against the OWASP Top Ten, and conducting a comprehensive evaluation of Yii2-based projects.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

In recent years, the increasing reliance on web applications has brought about a heightened need for robust security measures. The rise in cyber threats and attacks targeting web applications has highlighted the importance of addressing vulnerabilities to ensure the confidentiality, integrity, and availability of sensitive data. With the growing accessibility of the internet, more and more users engaging in online activities makes it crucial to prioritize the protection and security of web applications. As a result, numerous possibilities and breakthroughs have been created for both individuals and organizations.

Organizations in all industries face serious problems as a result of the frequency and sophistication of cyberattacks. In fact, human error is responsible for about 82% of all data breaches [1]. The main reason is that employees are the first line of defense against cyberattacks, and their activities directly impact the security posture of the organization. Without adequate cybersecurity training, the staff could unknowingly participate in unsafe actions like clicking on harmful links or downloading unverified files, leaving firms open to a variety of cyber threats. Besides, web applications may unexpectedly or unaware be developed due to the lack of cybersecurity background. Therefore, investing in comprehensive cybersecurity training programs is crucial to enhance the cybersecurity awareness and readiness of employees.

This research focuses on enhancing cybersecurity training scenario progression management in CyPROM [2]. These improvements will be derived from analyzing the Yii2 PHP framework [3] by using OWASP Top Ten [4], a well-established set of security vulnerabilities. The analysis involves assessing the framework's implementation against the OWASP Top Ten vulnerabilities, making it possible to identify and address any security issues present in

the framework. Furthermore, the analysis will provide insights into potential weaknesses and areas of improvement within the framework, ultimately contributing to the enhancement of cybersecurity training scenario progression management in CyPROM.

By leveraging the findings from the analysis, specific improvements can be proposed for CyPROM. These include incorporating security measures and best practices into the training scenarios, designing effective defense mechanisms, and improving the overall security posture of the system. In other words, the system's security can be bolstered and a robust cybersecurity training experience will be ensured.

1.2   Background

CyPROM   CyPROM [5] is an integrated module developed by Cyber Range Organization and Design (CROND) that uses scenario progression management to create dynamic training environments for cyber security. CyPROM aims to facilitate advanced cybersecurity training activities in dynamic environments that involve attack, forensics, and defense aspects of training. CyPROM not only allows instructors to provide training scenarios and target information that simulate realistic cybersecurity situations, but it also allows trainees to gain hands-on experience in dealing with dynamic cybersecurity challenges. As a result, CyPROM enhances the effectiveness of cybersecurity training activities.

Currently, CyPROM version 0.1 is the the latest release of the tool. It includes a user guide that provides instructions on using the tool, and source code archives are also available for further exploration and customization. This release marks an important milestone in the development of CyPROM, making it available for cybersecurity training purposes.

Yii2 PHP Framework   In the realm of web development, choosing the right framework is crucial for building robust, scalable, and secure applications. One such framework that has gained significant popularity and recognition in recent years is Yii2. Developed as the successor to Yii (which stands for "Yes, it is!" and is pronounced as Yee), Yii2 is a high-performance, component-based PHP framework that provides developers with a powerful toolkit for rapid application development.

Yii2 embraces the concept of "Don't Repeat Yourself" (DRY) and follows the principles of object-oriented programming (OOP), making it an ideal choice for building complex web applications. The framework offers a comprehensive set of features, robust security mechanisms, and extensive

documentation, empowering developers to create efficient and maintainable code.

OWASP Top Ten OWASP, an acronym for the Open Web Application Security Project, is a globally recognized non-profit organization dedicated to enhancing the security of web applications. One of the most influential contributions of OWASP is the OWASP Top Ten Project [4], which provides a comprehensive list of the ten most critical web application security risks. These risks are identified and updated periodically by experts in the field, reflecting the current landscape of cyber threats and vulnerabilities.

As technology continues to advance, so do the threats and vulnerabilities that organizations and individuals face. One crucial aspect of cybersecurity is understanding and mitigating the most common risks and vulnerabilities that exist in web applications. The OWASP Top Ten serves as a valuable resource for developers, security professionals, and researchers, guiding them in understanding and mitigating common security weaknesses in web applications. With the OWASP Top Ten, web developers can gain insights into the prevalent risks and apply appropriate measures to address them effectively.

1.3 Objectives

This research aims to enhance web application security through hands-on training for mitigating web application vulnerabilities. Through a comprehensive analysis of Yii2's implementation using the OWASP Top Ten, valuable insights will be gained into potential vulnerabilities and areas where usability can be enhanced to bolster CyPROM's overall effectiveness as a cybersecurity training scenario progression management system. As mentioned, The OWASP Top Ten is a globally recognized standard awareness document that identifies the most critical security risks to web applications. The research seeks to ensure that CyPROM minimizes the identified risks and aligns with the best practices outlined in the OWASP Top Ten, which is considered the most effective first step toward producing secure code, thereby enhancing the overall security posture of web applications built on this framework.

The objectives of this thesis can be summarized as follows:

1. **Analyze Yii2 vulnerabilities for improving its security**

   The first objective of this research is to conduct a thorough analysis of the Yii2 framework, with a focus on identifying potential vulnerabilities that could pose risks to web applications. By adopting a comprehensive approach, specific vulnerabilities were scrutinized, particularly

those falling within the OWASP Top Ten categories. The purpose is to gauge the potential impact of these vulnerabilities on web application security and establish a robust foundation for practical training. Armed with the knowledge gained from this investigation, web developers can reinforce their applications' defenses effectively. The ultimate goal is to provide a cohesive and comprehensive analysis that enhances the protection of web-based systems, making them more resilient to potential attacks.

2. **Enhance the training capabilities of CyPROM**

   Once the vulnerabilities have been identified through the analysis of the Yii2 framework, the next step is to implement the corresponding training capabilities and scenarios. This process will take into account the specific vulnerabilities uncovered during the investigation. The goal is to improve CyPROM and strengthen the resilience of the Yii2 framework against potential vulnerabilities in web applications. Implementing the necessary actions and scenarios based on the analysis results will strengthen the existing security measures and make the framework more adept at mitigating potential threats.

3. **Validate CyPROM enhancements using OWASP Top Ten**

   Last but not least, an important aspect of this research is a comprehensive testing and evaluation methodology to validate the efficacy of the mitigation strategies applied to address the OWASP Top Ten web application vulnerabilities. Real-world scenarios and simulated attacks will be employed to thoroughly assess the enhanced CyPROM's robustness and effectiveness. By understanding the significance of each OWASP Top Ten category within the context of web application security, this research aligns these categories with the analysis of not only Yii2 but also other commonly used frameworks for web application development. This alignment facilitates a comprehensive evaluation of the security measures implemented, enabling the identification of potential areas for further enhancement and contributing to overall web application security.

1.4 Significance

In this research, the Yii2 PHP framework is analyzed comprehensively based on the OWASP Top Ten to improve CyPROM, a cybersecurity training platform. These findings and analyses can provide valuable insights into im-

provements not only for CyPROM but also for other cybersecurity training platforms. By enhancing such platforms, we can make a significant contribution to improving cybersecurity preparedness and mitigating cyber threats. As cyber threats increase, including those perpetrated by individual actors and those by highly sophisticated entities utilizing artificial intelligence and machine learning, cybersecurity has become more and more important.

Furthermore, global connectivity and the use of cloud services have led to an increase in both inherent and residual risks, making organizations more susceptible to data breach campaigns. By addressing web application vulnerabilities through the analysis of Yii2 and the utilization of the OWASP Top Ten, we contribute to the broader goal of protecting sensitive data, intellectual property, and governmental and industry information systems. Ultimately, this research aims to play a significant role in enhancing cybersecurity measures, not only for CyPROM but for the wider realm of web application security.

1.5   Contributions

The main contributions of this thesis are as follows:

❖ **Investigated Yii2 security vulnerabilities**

This study made an important contribution by performing a comprehensive analysis of the Yii2 framework, focusing on identifying potential vulnerabilities that align with the OWASP Top Ten categories. By carefully examining specific vulnerabilities and thoroughly assessing their potential impact on web application security, this study provided valuable insight into the risks facing web-based systems. What set this study apart from others was its practical approach, which provided web developers with foundational training to strengthen web application defenses across all frameworks, transcending the scope of Yii2 alone. By emphasizing hands-on training for trainees, this study created a solid foundation for holistically strengthening web application defenses. The findings not only benefit Yii2 but are also relevant to other web application development frameworks, making them more resilient to potential cyberattacks. Besides, aligning these findings with the OWASP Top Ten criteria ensures effective mitigation of vulnerabilities and strengthens the overall security posture of Yii2-based projects and beyond.

❖ **Improved the security training capabilities of CyPROM**

Building on the vulnerability assessment results, this research took a proactive approach to improving the resilience of web application development frameworks beyond Yii2. Tailored and strategically implemented remediation efforts were used to directly address identified vulnerabilities, resulting in significant strengthening across multiple frameworks. Incorporating these enhancements ensured that industry best practices were adopted and the latest security standards were met, effectively hardening web applications against potential threats. With this comprehensive approach, CyPROM not only strengthened its security but also improved the overall security posture of web application development frameworks so that they can be effectively protected against a variety of cyber threats.

❖ **Conducted a comprehensive evaluation of the enhanced functionality of CyPROM**

The final critical contribution of this research was to conduct a comprehensive evaluation of CyPROM's effectiveness, focusing on the OWASP Top Ten criteria. By exposing the framework to real-world scenarios and simulated attacks, the robustness and effectiveness of the implemented remediation strategies were rigorously evaluated. Aligning the results with the OWASP top ten categories facilitated the evaluation of not only Yii2 but also other widely used web application development frameworks. This in-depth evaluation also enabled the identification of potential areas of improvement, leading to an improvement in web application security practices. As a result, the research findings are highly significant as they encourage the adoption of more stringent security measures not only within Yii2-based projects but also beyond to improve the overall security of web applications.

1.6 Thesis Structure

The remainder of the thesis is structured as follows:

❖ The thesis begins with an introduction that provided an overview of the research topic, its background, and the objectives pursued. It also highlights the significance of the study and outlines the contributions made to the field.

❖ Following the introduction, Chapter 2 provides an extensive exploration of related work, examining well-known platforms like WebGoat, Damn

Vulnerable Web Application, HackThisSite, eLearnSecurity, OverTheWire, and Juice Shop. The chapter then discusses the limitations, approaches, and challenges associated with the research.

❖ Chapter 3 presents the methodology used in the research, describing the components involved, including CyPROM, the Yii2 PHP Framework, and the OWASP Top Ten. The methodology procedures are then explained, covering initial planning, data collection, analysis, and the utilization of CyPROM and OWASP Top Ten to evaluate the Yii2 PHP Framework. The significance of the approach is also discussed, focusing on adherence to cybersecurity practices and the justification for its strengths.

❖ Chapter 4 focuses on the practical aspects of the research, focusing on hands-on training using CyPROM. It covers an overview of the training, strategies to mitigate vulnerabilities, and the implementation of training objectives, actions, and scenarios. The chapter also discusses the process of conducting the hands-on training and evaluating its effectiveness.

❖ In Chapter 5, the evaluation is presented, including a functionality evaluation and a comparative analysis. The implemented actions and scenarios were assessed within the context of the overall scope of the research. Additionally, user evaluations were conducted to gather feedback on the developed system.

❖ Chapter 6 presents the conclusion, summarizing the key findings and insights from the research. It also discusses potential future work in the field and offers a conclusive statement based on the research findings and outcomes.

Chapter 2

Related Work

The field of web application security has gained significant attention due to the increasing prevalence of vulnerabilities and the potential risks they pose to organizations and their users. Before CyPROM was developed, several cybersecurity education and training programs have been created in Japan, such as enPiT-Security (SecCap) [6], SECCON (SECurity CONtest) [7], CYDER [8], or the Hardening Project [9], etc. that focus on practical activities for participants with different skill levels, such as university students or IT professionals.

In addition, there are a number of other hands-on training programs for web application security. Programs cater to trainees' levels, offering various features and benefits. Students prefer comprehensive programs covering various topics and the latest threats, while professionals prefer experienced programs with challenging exercises.

2.1 WebGoat

WebGoat is an intentionally insecure application that allows you to train the vulnerabilities of web applications that use open-source components commonly found in the Java community [10]. The motto "Learn the hack - Stop the attack" sums up the primary goal: to create a safe and controlled environment for individuals to learn about various security vulnerabilities and the appropriate countermeasures. WebGoat was developed by the Open Web Application Security Project (OWASP) and includes lessons for nearly all of the OWASP Top Ten vulnerabilities, allowing users to explore and understand common vulnerabilities and the potential risks associated with them.

The interactive nature of the platform allows users to actively engage in

the learning process through hands-on exercises. Participants can practice identifying and exploiting security vulnerabilities in the WebGoat application and gain a deeper understanding of how hackers might attack web applications in the real world.

However, WebGoat aims not only to teach participants how to exploit security vulnerabilities but also to prepare them for real-world attacks by providing them with the necessary knowledge and skills. After completing the challenges, users receive guidance and explanations on how to secure their applications. WebGoat demonstrates both the offensive and defensive aspects of web application security, enabling participants to proactively protect their web applications.

WebGoat also encourages community input to continuously improve the platform and keep up with new threats and defensive techniques. In addition, the project's accessibility ensures that knowledge about web application security is widely available, benefiting aspiring developers, security professionals, and enthusiasts alike.

2.2 Damn Vulnerable Web Application

The Damn Vulnerable Web Application (DVWA) is a virtual machine with a web server built with PHP and MySQL that is intentionally vulnerable [11]. It provides a legal environment for security experts to test their skills and tools. Web developers can gain a better understanding of web application security processes, while students can quench their thirst for knowledge in a controlled classroom environment.

DVWA has several categories, each divided into three levels from basic to advanced so that users can progress gradually and deepen their knowledge. Vulnerabilities covered by DVWA include SQL injection, cross-site scripting (XSS), command injection, insecure file uploads, and more [12].

It is important to note that while DVWA is a valuable tool for learning web application security, it should only be used in a legal and ethical manner. Users should obtain proper authorization before performing security assessments or penetration testing on real-world applications.

By actively engaging with DVWA, users can gain hands-on experience identifying and mitigating common web application vulnerabilities, and developing the necessary skills to effectively secure web applications.

2.3 HackThisSite

HackThisSite (HTS) [13] is an online platform and community dedicated to promoting and educating individuals about ethical hacking, computer security, and cybersecurity. It was created to provide a safe and legal environment for users to learn and practice their hacking skills without violating the boundaries of legality and ethics. The site offers a variety of challenges and missions, including CTFs [14] and wargames [15], designed to evaluate and improve users' knowledge of various security-related concepts [13]. These challenges cover topics such as cryptography, web application security, steganography, programming, and more.

Participants can join HackThisSite for free and complete the challenges at their own pace. As users progress through the challenges, they earn more points and can showcase their expertise to the community. HackThisSite encourages users to share their knowledge, collaborate, and learn from each other to create a supportive and educational environment [13].

2.4 eLearnSecurity

The organization eLearnSecurity is one of the largest providers of practical cybersecurity training and certification [16]. They provide a diverse range of courses, for both individuals [17] and organizations [18], aimed to equip participants with the capabilities and grasp needed in the complex world of security. With the rising demand for cybersecurity experts, eLearnSecurity has emerged as a trusted and respected source for comprehensive and hands-on training in this critical field.

With a commitment to practicality and real-world relevance, eLearnSecurity stands out in the crowded marketplace of cybersecurity training providers. Their courses are developed and delivered by experienced industry professionals who possess a deep understanding of the evolving threats and challenges faced by cybersecurity practitioners. This ensures that participants not only gain theoretical knowledge but also acquire the practical skills required to tackle actual security issues in today's digital environments.

eLearnSecurity's course catalog covers a wide spectrum of cybersecurity domains, catering to both beginners and seasoned professionals. From entry-level programs such as Certified Secure Computer User (CSCU) and Certified Professional Penetration Tester (eCPPT) to more advanced offerings like Certified Red Team Professional (eCRT) and Advanced Reverse Engineering of Software (ARES), eLearnSecurity offers a progression of courses that allow individuals to grow and specialize in their cybersecurity careers [17].

In addition to its comprehensive training programs, eLearnSecurity also provides industry-recognized certifications that validate the skills and expertise gained through its courses. These certifications, such as eJPT (eLearnSecurity Junior Penetration Tester), eCPPTv2 (eLearnSecurity Certified Professional Penetration Tester (v2)), eWPT (eLearnSecurity Web application Penetration Tester), or eWPTXv2 (eLearnSecurity Web application Penetration Tester eXtreme), and eCTHPv2 (eLearnSecurity Certified Threat Hunting Professional v2), and so on are highly regarded within the cybersecurity community and can significantly enhance one's professional credibility and employability.

In summary, eLearnSecurity stands at the forefront of cybersecurity education, providing practical and hands-on training coupled with industry-recognized certifications. With its diverse course offerings, emphasis on real-world relevance, and commitment to equipping individuals with the skills needed to address modern cybersecurity challenges, eLearnSecurity is a trusted partner for anyone seeking to develop or advance their career in this rapidly evolving field.

2.5   OverTheWire

OverTheWire is an online platform that hosts a series of "wargames" [19] which are essentially security-related challenges presented in a safe and controlled environment. Each wargame focuses on a specific aspect of cybersecurity, presenting users with tasks that require critical thinking and technical expertise. By engaging in these challenges, participants can enhance their problem-solving abilities, gain practical experience, and develop a profound understanding of various security concepts.

One of the main advantages of OverTheWire is its emphasis on hands-on learning. Traditional cybersecurity education often lacks practical elements, leading to knowledge gaps and an inability to apply theoretical concepts in real-world scenarios. OverTheWire bridges this gap by providing users with opportunities to interact with various security challenges, enhancing their problem-solving and analytical skills.

In other words, OverTheWire offers a unique and practical approach to cybersecurity education, enabling individuals to enhance their skills and knowledge in a hands-on manner. When utilizing OverTheWire or any external source, learners should prioritize avoiding plagiarism by properly attributing and referencing the information they use. By doing so, they uphold academic integrity and contribute to the growth of the cybersecurity community by acknowledging the original creators and enabling others to verify

the information. With a commitment to ethical conduct and responsible use of resources, individuals can maximize their learning potential through OverTheWire and similar platforms.

2.6 Juice Shop

OWASP Juice Shop is a well-known open-source web application created by Bjoern Kimminich [20] and maintained by OWASP, specifically designed as an interactive platform to train developers, security professionals, and enthusiasts about the security vulnerabilities prevalent in web applications. The primary goal of the Juice Shop is to provide a safe environment for trainees to comprehend and mitigate various web application security risks. This feature-rich application, which acts as a realistic online e-commerce store, intentionally includes a wide range of security risks from the OWASP Top Ten list, OWASP ASVS [21], the OWASP Automated Threat Handbook [22], the OWASP API Security Top Ten [23], and MITRE's Common Weakness Enumeration [24]. These risks are categorized into different classes, such as SQL injection, cross-site scripting (XSS), faulty authentication, etc., with over 100 challenges. By embedding these vulnerabilities in Juice Shop's codebase, individuals can practice identifying and fixing them under controlled conditions. As they navigate through the application, participants encounter diverse challenge categories, each presenting a unique opportunity to understand the significance of secure coding practices and the consequences of leaving vulnerabilities unaddressed.

OWASP Juice Shop caters to individuals of all skill levels, from novices to seasoned security professionals. Its carefully crafted challenges cover a broad spectrum of web application vulnerabilities, enabling individuals to gain practical experience in securing web applications effectively. Notably, the platform fosters a thriving community of security enthusiasts, researchers, and contributors who actively collaborate to enhance the application continuously [20]. This community-driven approach ensures that Juice Shop remains updated with the latest security threats and best practices, guaranteeing its relevance and reliability as a training ground for users seeking web application security proficiency.

One standout feature of OWASP Juice Shop is its Capture The Flag (CTF) functionality, which gamifies the process of identifying vulnerabilities. This feature encourages users to apply their knowledge in a competitive environment, spurring them to explore the intricacies of the application and develop effective problem-solving skills. The CTF aspect adds excitement and engagement to the training process, making OWASP Juice Shop

an invaluable resource for anyone aiming to fortify web applications against potential threats.

In conclusion, OWASP Juice Shop is a platform for hands-on training in mitigating web application vulnerabilities. Its emulation of real-world e-commerce scenarios, incorporation of OWASP Top Ten flaws, diverse challenge categories, collaborative ecosystem, and gamified CTF feature make it an ideal choice for developers, security practitioners, and individuals interested in bolstering web application security. By providing a safe and practical learning environment, Juice Shop empowers users to enhance their knowledge and expertise, contributing to a more secure online landscape.

2.7 Discussion

While several hands-on training programs and resources exist to enhance web application security skills, there are still limitations that necessitate further improvements. This section discusses the limitations observed in the existing programs, which serve as the driving factors behind the need for the research presented in this thesis. By addressing these limitations, we aim to enhance the training capabilities for mitigating web application vulnerabilities effectively.

2.7.1 Limitations

Limited Coverage and Relevance The hands-on training programs presented have been instrumental in providing practical exercises for web application security. However, they focus on specific vulnerability types or outdated frameworks, limiting their coverage and relevance in today's web application development landscape. To address the ever-evolving nature of web application vulnerabilities, there is a need for a comprehensive and up-to-date training approach.

As web development frameworks continue to evolve, it is crucial for hands-on training programs to keep pace with these advancements. Many existing platforms focus on older or less prevalent frameworks, limiting their relevance to current industry practices. Consequently, trainees may not acquire the necessary skills to secure modern web applications built on popular frameworks. By analyzing a widely used framework like Yii2, trainees can deeply understand its vulnerabilities and develop tailored exercises that align with contemporary web development practices, thus bridging the gap between theoretical knowledge and practical application.

13

Lack of Integration with Development Processes One common limitation of existing hands-on training programs is their detachment from the software development process. Many developers receive security training as a separate activity, disconnected from their day-to-day development tasks.

This disjointed approach hampers the practical application of learned security practices and can lead to a gap between theory and implementation. By focusing on improving CyPROM, a web application vulnerability mitigation tool, this research aims to bridge the divide by integrating hands-on training directly into the development workflow, enabling developers to apply security measures seamlessly.

2.7.2 Approach

By addressing the above limitations and gaps in the current landscape of hands-on training for mitigating web application vulnerabilities, this research aims to contribute to the development of a more comprehensive and effective approach. The proposed improvements to CyPROM, informed by the analysis of Yii2 using OWASP Top Ten, seek to provide learners with realistic training scenarios, comprehensive coverage of critical vulnerabilities, integration with development processes, customization and adaptability, robust evaluation and feedback mechanisms, and opportunities for collaboration and knowledge sharing.

2.7.3 Challenges

Our approach requires considering several key challenges:

- ❖ Which vulnerabilities are appropriate for inclusion in the training program?

- ❖ How to identify existing security issues in the frameworks for developing web applications that are relevant from a training perspective?

- ❖ How to analyze the potential impact of the security issues and to provide recommendations for mitigation?

- ❖ How to consider the goals of the training program and conduct a needs assessment or pre-test to identify areas that require improvement?

By conquering these challenges, this research aims to overcome obstacles and achieve the objectives specified in the Introduction.

Chapter 3

Methodology

This thesis focuses on improving CyPROM through practical exercises to mitigate vulnerabilities in web applications. This section describes the step-by-step process followed to achieve the objectives. The methodology includes several stages, including initial planning, data collection, analysis, and implementation of improvements. By following this systematic approach, researchers and web developers can identify and effectively address web application vulnerabilities. This methodology also serves as a roadmap for web application projects and guides the participants to achieve the desired results.

3.1 Component Overview

Before delving into the thesis methodology, it is essential to provide a thorough understanding of the key components involved in this thesis. These elements serve as the foundation on which the methodology is built. By deeply understanding these components, readers can better grasp the objectives and scope of the thesis. Additionally, understanding these factors enables a more precise analysis of the findings.

3.1.1 CyPROM

CyPROM is a training-support system developed by the Cyber Range Organization and Design (CROND) at JAIST [2] utilizing scenario progression management to create dynamic cybersecurity training environments. CyPROM aims to enable advanced training activities in attack, forensics, and defense aspects in dynamic environments. The system incorporates realistic simulations of cyber attacks and allows trainees to practice their skills in a controlled and immersive environment. By providing a hands-on train-

ing experience, CyPROM enhances the effectiveness of cybersecurity training programs and prepares individuals for real-world cyber threats.

Overview of CyPROM

Figure 3.1 shows an overview of CyPROM. The CyPROM architecture is a complex system that consists of multiple interconnected components. It provides a comprehensive framework for managing and optimizing various aspects of cybersecurity. Figure 3.1 visually represents the different modules and their relationships within the architecture, giving a clear understanding of how they work together to enhance overall security measures.
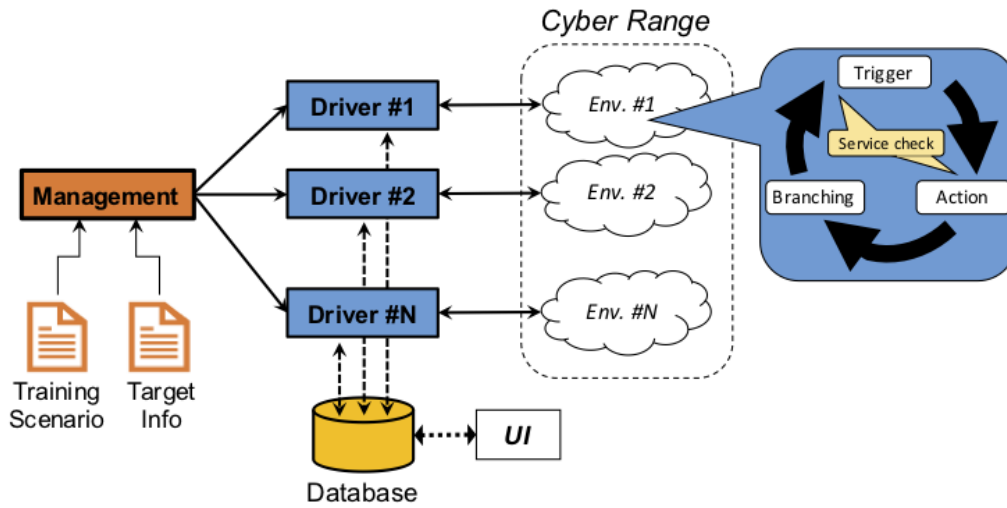
Figure 3.1: CyPROM scenario progression management modules [5].

As shown in Figure 3.1, the management module initiates processes for executing training scenarios in the training environment based on training scenarios and target information provided by the instructors. This module ensures that each trainee's actions trigger specific actions and branching paths within the scenario, allowing for a dynamic and personalized training experience. Additionally, it constantly monitors and adapts the scenario progression based on real-time feedback and performance metrics to optimize the learning outcomes for each trainee. Each scenario driver employs a "Trigger-Action-Branching" mechanism for independent progression based on individual actions. This mechanism ensures that trainees have control over their own learning experience and can explore different paths and outcomes based on their choices. It also promotes active engagement and critical thinking as trainees navigate through the scenario. The "Trigger-Action-Branching"

mechanism enhances the overall effectiveness of the training program by tailoring it to each trainee's unique needs and learning style.

As the training proceeds, the data from the training sessions, including the details of each action performed and its outcome, is stored in a database so that the progress of the scenario can be tracked in real-time and training reports can be generated for each trainee. This data can be accessed through a user interface or generated at the end of the session. The user interface also allows trainees to review their performance and identify areas for improvement. Besides, the generated training reports provide valuable insights for instructors to assess the overall progress of trainees and make informed decisions regarding further training strategies.

Key Components of CyPROM

This section explains the elements in Figure 3.1, highlighting their role in the functioning of CyPROM.

1. **Management Module**

   The management module manages CyPROM's basic features, including input file validation and database initialization. Its primary function is to start a scenario driver module for each training participant in parallel, ensuring independent scenario progression based on trainee actions and environment.

2. **Training Scenarios**

   Training scenarios are composed of a set of steps, each step containing information about the action, target machine, parameters, and so on. These scenarios are designed to simulate real-world situations and provide hands-on experience for trainees. They allow trainees to practice executing different actions and improve individual skills in a controlled environment.

   The scenarios also include branching information for the driver to decide which step to execute next. This branching information adds an element of decision-making and adaptability to the training scenarios. It allows trainees to learn how to make informed choices based on the given circumstances, enhancing their problem-solving abilities in real-world situations. Furthermore, the management module ensures that trainees receive comprehensive training by covering a wide range of actions and target machines, enabling them to develop a versatile skill set.

The training scenario description is provided as a YAML file, a text-based format suitable for instructors to specify using a regular text editor. YAML's text-based format allows instructors to easily define and customize various aspects of the training scenarios, such as the target machine, action details, and so on. In addition, Its human and machine readability ensures that instructors can easily share and collaborate on scenario descriptions with other team members or developers.

A training scenario file begins with the keyword 'scenario' followed by a list of step blocks. Each step block contains a label, a target, a trigger block, an action block, and branching elements. The label is used to identify the step, while the target specifies the machine on which the action is performed. The trigger block contains additional information about when the action should be triggered. The action block defines the specific details of the action to be executed. Finally, the Success and Failure branching elements determine which steps should be executed depending on whether the action is successful or not. Figure 3.2 shows the scenario description required for CyPROM representation.

```
1   scenario:
2   - step: A01
3       target: server1
4       action:
5        module: scan_open_ports
6        host: 192.168.xxx.xxx
7       success: A02
8       failure: COMPLETE
9   - step: A02
10      target: server1
11      action:
12       module: flood_attack
13       host: 192.168.xxx.xxx
14      success: COMPLETE
15      failure: COMPLETE
```

Figure 3.2: Basic CyPROM scenario with two actions.

Next, step definition key elements are going to be presented with their roles. For more specific details regarding CyPROM scenario components, please consult reference "CyPROM User Guide" [5].

❖ **step** - Identifier for step in branching logic, used to indicate step execution for success or failure outcome of an action in a scenario.

❖ **target** - Name of the target for the action included in this step.

❖ **action** - A scenario step contains an action definition, representing the core activity to be carried out at a specific point in the

scenario. Each action has a generic field and module-specific fields, possibly optional.

❖ **success** - The step ID is used to execute a successful action, while if not provided, the next step in the file will be executed.

❖ **failure** - The step ID is used to execute a failed action, while if not provided, the next step in the file will be executed.

It should be noted that complex circumstances require more operations than merely determining the next step. As a result, an alternative branching logic representation that makes use of a dictionary to allow for the identification of more choices is required. Figure 3.3 demonstrates the various possible options available in this case, both for successful and unsuccessful outcomes.

Step id	Meaning
REPEAT	End the current scenario, but keep it so that it can be executed again during the same training session
FINISH	End the current scenario, and do not keep it for repeated execution in the future
COMPLETE	End the entire training session, even if scenario files that were not executed yet still exist

Figure 3.3: Predefined step ids with their meanings.

Following the brief samples for the scenario representation syntax, this part provides a more realistic scenario example based on actual training activities conducted.

The first step in drafting a CyPROM scenario is to consider it from a logical perspective. Visualize training scenarios as a flowchart, with actions executed and branching decisions made based on the outcome. The flowchart for the scenario discussed in Figure 3.4 is an example.

The example scenario includes two actions: (i) scan_open_ports – identify which ports are open, which services are running on those ports, and potentially identify vulnerabilities that can be exploited to gain unauthorized access to the system; (ii) flood_attack - target servers by flooding them with a high volume of traffic. As soon as the first action succeeds, the second action will be executed. If any action fails, the attack will be terminated. As a result, training will continue until all actions fail - which indicates that trainees have successfully defended.
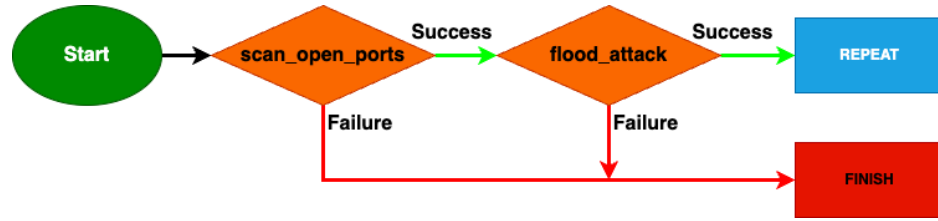
Figure 3.4: Execution flowchart of training scenario example.

3. Target Information

To provide a generic representation of a training scenario, CyPROM must provide network information for action targets, represented by labels. This is achieved by using a separate training session-dependent file with a section for each trainee or team, using square bracket syntax. Each participant section contains a sequence of target machine labels and the actual IP address of the machines in the training environment. An example of such a target information file is shown in Figure 3.5, where two targets are defined per participant, such as two web servers.

```
[Team A]
server1 = 127.0.0.1
server2 = 127.0.0.1

[Team B]
server1 = 127.0.0.1
server2 = 127.0.0.1

[Team C]
server1 = 127.0.0.1
server2 = 127.0.0.1
```

Figure 3.5: CyPROM target information file.

The management module verifies training scenario and target information files, creating scenario driver instances with the number of participants specified. Each driver receives the training scenario file name and target network details.

4. Scenario Driver

The core functionality is provided by the scenario driver module, which uses the Trigger-Action-Branching mechanism.

Figure 3.6: The Trigger-Action-Branching mechanism.

This mechanism determines when an action is executed and which scenario step should be executed next. The scenario driver checks the state of services on target machines, only running triggers and actions when available, to avoid undesired situations like attacks when the service is not running. If the service is found not to be running, execution is paused and the check is repeated periodically.

As shown in Figure 3.6, the scenario driver's workflow starts with the top-most step in the description file, which is executed by a trigger. If a trigger is present, the module waits for it to complete before executing the action. The next step is selected from other steps via the branching mechanism, and the processing flow continues. The workflow consists of three main elements.

1. **Trigger**

    According to training needs and specificities, CyPROM uses triggers to delay or interrupt scenario progression. There are two trigger modules available:

    - **timer** - delays action execution by a predefined amount.

- **signal** - prevents action execution until a notification is received.

Timers enable time-driven training, while signals enable event-driven training, allowing realistic activities in various conditions. The signal trigger is currently used for trainees to control scenario execution through buttons in a UI, but it can also be used to unblock scenario progression using external tools.

2. **Action**

   Each scenario step contains an action, and instructors have to specify the module to execute. These modules form an "action library" as a base for various scenarios. There are 9 actions available:

   - **message, hint, question** - UI modules for sending messages, hints, and requesting trainee input.
   - **cmd_injection** - Perform command injection tasks on a given target web server.
   - **metasploit** - Simplifies Metasploit's interface [25], allowing instructors to execute the framework using CyPROM syntax.
   - **php_auth_bypass** - Exploits a vulnerability in WordPress v4.7.0, altering the top web page using a predefined pattern.
   - **ssh_cmd_exec** - Perform remote command execution using login information.
   - **ssh_dict_attack** - A lightweight SSH dictionary attack module for checking login information stored in files or obtained through other means.
   - **test** - A testing tool for Trigger-Action-Branching mechanism, without requiring a cyber range. It simulates execution time using the `sleep()` function in Python, without any specific parameters. This module is similar to the `none` trigger module.

   For each action, instructors can specify some arguments belonging to each action. The arguments' description can be found in the CyPROM User Guide [5]. Modules can also return specific data that is available for the next command in addition to the exit status. This allows instructors to customize the actions and provide specific instructions for each trainee. The availability of specific data for the next command enhances the training experience and allows for more realistic scenarios to be created.

22

3. Branching

The scenario driver determines the next step to execute based on the success or failure of the current action, enabling flowcharts to design scenarios with two possible outcomes. This binary decision is evident in defense training, where success indicates the trainees' defense tactic was incorrect, allowing the system to proceed with the next steps. Failure, however, indicates the presence of defense mechanisms and requires a different attack sequence, or scenario execution can end. In addition to defense training, this binary decision-making process is also commonly used in various other fields such as software development and problem-solving [26, 27, 28]. By incorporating a scenario driver, developers can create more dynamic and adaptable systems that can respond differently based on the outcomes of previous actions.

Each scenario step indicates the next step to be executed if the action was successful or failed by using the keywords `success` and `failure`. If the corresponding keyword is missing, the subsequent step is executed. There are three special step labels, explained in Figure 3.3, that can be used to control scenario execution.

5. Database

CyPROM uses a database to store various data for several purposes, including UI interaction information, configuration settings, logs of actions, current step information, and service check information. The database, implemented by the SQLite relational database management system (RDMS) [29], is initialized by the management module and updated by scenario driver instances and the API module. The use of SQLite allows for efficient storage and retrieval of data, ensuring the smooth operation of the system. This database plays a crucial role in maintaining the integrity and consistency of the system's information across different modules and interactions.

In conclusion, CyPROM is a scenario progression management system specifically designed for advanced cybersecurity training. Its "Trigger-Action-Branching" mechanism enables individual participants to progress through scenarios based on their actions, creating a personalized and realistic training experience. With its focus on cybersecurity training, CyPROM contributes to enhancing the overall security posture of both individuals and organizations. It equips trainees with the necessary skills and experience to effectively identify, mitigate, and respond to cyber threats. By simulating real-world

scenarios and integrating multiple training aspects, CyPROM prepares individuals to handle various cybersecurity challenges, including attacks, forensic investigations, and defensive strategies. This hands-on approach to training helps bridge the gap between theoretical knowledge and practical application, ensuring that cybersecurity professionals are well-prepared to protect systems and data against evolving threats.

3.1.2 Yii2 PHP Framework

Yii2 is a powerful PHP framework that facilitates the development of robust and high-performance web applications. It is the successor to the original Yii framework and has been one of the best frameworks, suggested by both developers and organizations, due to its simplicity, extensibility, and efficiency [30].

Overview of Yii2

Yii2 follows the Model-View-Controller (MVC) architectural pattern, promoting a clear separation of concerns and enhancing code organization. One of the notable features of Yii2 is its excellent performance. The framework is designed to be highly optimized and offers various caching mechanisms, such as page caching, fragment caching, and data caching, which significantly improve the response time of applications [31]. Also, Yii2 leverages lazy loading and dynamic class loading techniques, ensuring that resources are loaded only when needed, thus minimizing the overall memory footprint. It emphasizes the concept of "convention over configuration" [32], which means that it provides sensible defaults for various components, reducing the amount of configuration required. This allows developers to quickly set up a project and start building applications without spending excessive time on boilerplate code.

In terms of extensibility, Yii2 provides a modular architecture that encourages the use of reusable and interchangeable components. The framework offers a rich ecosystem of extensions and plugins, allowing developers to effortlessly enhance their applications with additional functionality. These extensions cover a broad range of areas, including authentication, caching, RESTful APIs, and so on.

One important thing is that Yii2 has an active and supportive community of developers who contribute to its growth and share their knowledge through forums, tutorials, and documentation. The official Yii website [3] provides comprehensive documentation, code samples, and a cookbook to help developers learn and leverage the framework effectively. Krajee (Kartik-v) pro-

vides a number of extensions and modules along with their visual demonstration [33].

In conclusion, Yii2 is a feature-rich PHP framework that offers a solid foundation for building scalable and efficient web applications. Its performance optimizations, extensibility, and security features make it a popular choice among developers. By leveraging Yii2's capabilities, developers can streamline their development process and deliver high-quality applications efficiently.

Key Features of Yii2

Some key features of Yii2 include:

1. **MVC Architecture**

    Yii2 follows the Model-View-Controller (MVC) architectural pattern, which promotes the separation of concerns and modular development. This helps in maintaining clean and organized code.

2. **High Performance**

    Yii2 is built with performance in mind. It utilizes lazy loading, efficient caching mechanisms, and other optimization techniques to ensure fast response times and reduced server load.

3. **Database Access**

    Yii2 supports multiple database systems, including MySQL, PostgreSQL, SQLite, and more. It provides an easy-to-use database abstraction layer, also known as the Object-Relational Mapping (ORM) feature, that simplifies database interactions.

4. **Form Handling and Validation**

    Yii2 includes a powerful form-handling mechanism that simplifies the creation, validation, and processing of forms. It provides built-in validation rules and supports client-side and server-side validation.

5. **Security**

    Yii2 incorporates various security measures to protect against common web vulnerabilities. It includes features like input validation, cross-site scripting (XSS) prevention, cross-site request forgery (CSRF) protection, and encrypted data transmission.

6. **RESTful API Development**

   Yii2 has excellent support for building RESTful APIs. It provides tools for generating API documentation, handling authentication and authorization, and designing clean and scalable API endpoints.

7. **Internationalization (I18N) and Localization (L10N)**

   Yii2 simplifies the process of internationalizing and localizing web applications. It provides features for translating messages, formatting dates, numbers, and currencies according to different locales.

8. **Caching and Session Handling**

   Yii2 offers caching mechanisms to boost performance by storing frequently accessed data in memory or other storage systems. It also provides efficient session management tools for handling user sessions.

9. **Testing Support**

   Yii2 has built-in tools for unit testing and functional testing of applications. It makes it easier to write test cases and perform automated testing, improving the overall quality and reliability of the code.

10. **Extension Ecosystem**

    Yii2 has a vast and active developer community that has contributed numerous extensions and plugins to the framework. These extensions provide additional functionality that can be easily integrated into Yii2 applications.

Yii2 Architecture

Yii uses the Model-View-Controller (MVC) design pattern, which separates logic from user interface considerations [31]. MVC allows developers to easily change parts without affecting the others. The model represents data and rules, while the view contains user interface elements like text and form inputs. The controller manages communication between the model and the view as explained next:

❖ **Model**

   Models in Yii2 represent data, business logic, and rules. They encapsulate the application's data and define the operations that can be performed on that data. Models can be created by extending the `yii\base\Model` class or its child classes. They support features such

as attributes, attribute labels, massive assignments, validation rules, and data exporting [31].

❖ **View**

Views in Yii2 are responsible for presenting data to end-users. They are the output representation of models. Views are created as view templates: PHP script files containing HTML and presentational PHP code. The view application component manages views and provides view composition and rendering methods.

❖ **Controller**

Controllers in Yii2 take user input and convert it into commands for models and views. They handle user requests and orchestrate the interaction between models and views. Controllers contain action methods that define the available actions that users can perform. These methods receive user input, interact with models to perform the necessary operations, and pass the data to views for presentation.

Yii2 offers two versions: the basic template and the advanced template. The choice between these two templates depends on the requirements and complexity of your project.

Feature	Basic	Advanced
Project structure	✓	✓
Site controller	✓	✓
User login/logout	✓	✓
Forms	✓	✓
DB connection	✓	✓
Console command	✓	✓
Asset bundle	✓	✓
Codeception tests	✓	✓
Twitter Bootstrap	✓	✓
Front- and back-end apps		✓
Ready to use User model		✓
User signup and password restore		✓

Figure 3.7: Comparison of Yii2 templates [34].

27

As shown in Figure 3.7, the basic template provides a simple structure for building web applications with Yii2. It includes essential features such as the project structure, site controller, user login/logout, forms, DB connection, console commands, asset bundles, Codeception tests, and Twitter Bootstrap integration. This template is suitable for smaller projects or projects that do not require a separate "backend/frontend" division.

On the other hand, the advanced template offers a more comprehensive and feature-rich architecture. It includes all the features of the basic template and adds additional features like front-end and back-end applications, a ready-to-use `User` model, and user signup and password restore functionality. The advanced template provides a separation between the frontend and backend applications, allowing for more modular development and better code organization. The frontend tier in the advanced template is responsible for handling the user interface and client-side interactions, while the backend tier focuses on business logic and data management. This separation enhances code reusability and scalability, making it suitable for larger projects or projects that require a more extensive application structure. Figure 3.8 shows the Yii2 Advance template structure.

The architecture of Yii2 Advanced is designed to cater to the needs of both small and large-scale applications. As shown in Figure 3.8, it consists of two main application tiers: the frontend and the backend. Each tier represents a separate application with its own set of features and can be developed and deployed independently.

To facilitate the development process, Yii2 Advanced incorporates a range of features and components. These include the Yii2's powerful ActiveRecord ORM (Object-Relational Mapping) for seamless database operations, a robust routing system for handling URL mappings, a flexible and extensible widget toolkit for creating reusable UI elements, and a built-in authentication and authorization system for managing user access control.

One notable aspect of Yii2 Advanced's architecture is its emphasis on configuration. The framework provides a comprehensive configuration mechanism that allows developers to fine-tune various aspects of the application. This includes specifying database connections, configuring URL rules, defining module hierarchies, and setting up caching and error-handling components. The flexibility offered by the configuration system enables developers to tailor the framework to meet specific project requirements.

In conclusion, Yii2 is a powerful PHP framework for developing robust, scalable web applications, following the MVC architectural pattern. Its well-organized directory structure promotes modularity and code separation, enabling developers to manage and maintain their codebase. With an extensive set of components and services, Yii2's ecosystem of extensions makes it an

Name ^	Date modified	Type	Size
.idea	5/4/2023 3:21 AM	File folder	
backend	4/26/2023 6:22 PM	File folder	
common	4/26/2023 6:22 PM	File folder	
console	4/26/2023 6:22 PM	File folder	
environments	4/26/2023 6:22 PM	File folder	
frontend	4/26/2023 6:22 PM	File folder	
resources	4/26/2023 6:22 PM	File folder	
vagrant	4/26/2023 6:22 PM	File folder	
vendor	4/26/2023 6:42 PM	File folder	
.bowerrc	4/26/2023 6:22 PM	Bower RC Source ...	1 KB
.gitignore	4/26/2023 6:22 PM	Git Ignore Source ...	1 KB
codeception.yml	4/26/2023 6:22 PM	Yaml Source File	1 KB
composer.json	4/26/2023 6:22 PM	JSON Source File	2 KB
composer.lock	4/26/2023 6:22 PM	LOCK File	230 KB
docker-compose.yml	4/26/2023 6:22 PM	Yaml Source File	1 KB
init	4/26/2023 6:22 PM	File	11 KB
init.bat	4/26/2023 6:22 PM	Windows Batch File	1 KB
LICENSE.md	4/26/2023 6:22 PM	Markdown Source...	2 KB
README.md	4/26/2023 6:22 PM	Markdown Source...	4 KB
requirements.php	4/26/2023 6:22 PM	PhpStorm2022.3	6 KB
Vagrantfile	4/26/2023 6:22 PM	File	3 KB
yii	4/26/2023 6:22 PM	File	1 KB
yii.bat	4/26/2023 6:22 PM	Windows Batch File	1 KB
yii_test	4/26/2023 6:22 PM	File	1 KB
yii_test.bat	4/26/2023 6:22 PM	Windows Batch File	1 KB

Figure 3.8: File structure in Yii2 PHP Framework advanced template.

ideal choice for web application development.

Key Components of Yii2

The Yii2 architecture comprises key components for a robust, scalable web application development framework. These components are essential for understanding the organization and functionality of Yii applications. They play key roles in implementing the MVC architecture, separating concerns, and facilitating the development of scalable and maintainable web applications.
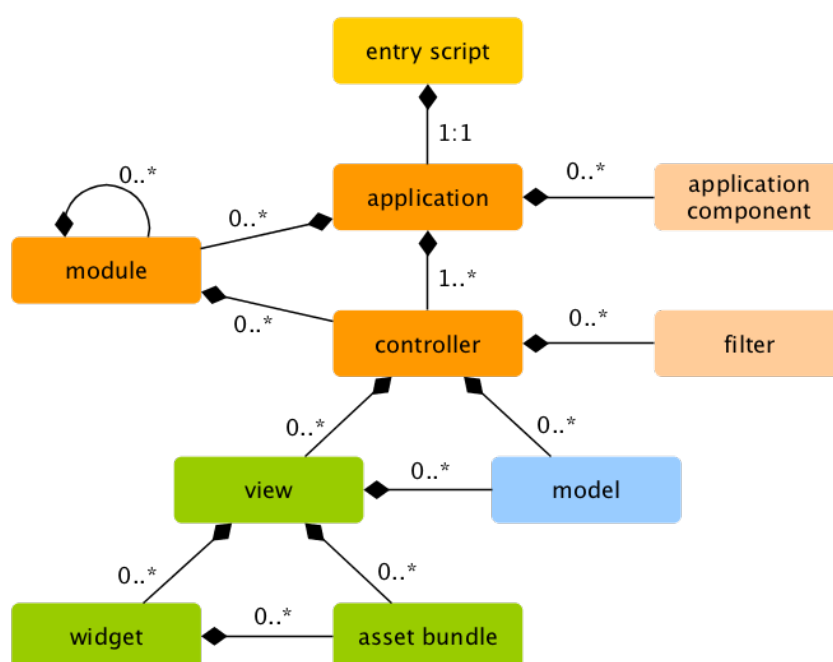
Figure 3.9: The static structure of Yii2 application [31].

As shown in Figure 3.9, besides MVC, Yii2 applications also have some other entities as detailed next:

❖ **Application Components**

Application components are objects registered with Yii applications that provide various services for fulfilling requests. They are responsible for tasks such as routing web requests, managing database-related services, caching, and more. Each application component has a unique ID and can be accessed globally through the expression "**\Yii::$app→componentID**". Application components are configured in the "**yii\ba-**

se\Application:$components" property of the application configuration.

❖ **Modules**

Modules in Yii2 are self-contained software units that contain models, views, controllers, and other supporting components. They are often viewed as mini-applications within the main application. Modules provide a way to organize and encapsulate related functionality. They cannot be deployed alone and must reside within applications. Modules have their own directory structure, similar to applications, with sub-directories for controllers, models, views, and so on [31].

❖ **Assets**

Assets in Yii2 are files, such as CSS, JavaScript, images, and videos, that are referenced on a web page. They are located in web-accessible directories and are directly served by web servers. Yii provides powerful asset management capabilities, allowing assets to be managed programmatically through asset bundles. An asset bundle is a collection of assets located in a directory and can be used to manage the inclusion and versioning of assets in the application [31].

❖ **Extensions and Widgets**

Extensions in Yii2 are additional libraries or packages that extend the functionality of the framework. They can be integrated into applications to provide extra features and capabilities. Widgets are a type of extension in Yii2 that can be embedded in views. They encapsulate controller logic and can be reused across different views. Widgets enhance the reusability and modularity of views in Yii applications [31].

In collaboration with other entities, they implement the MVC architectural pattern, separate concerns, and facilitate robust and maintainable web application development.

3.1.3 OWASP Top Ten

OWASP Top Ten, published by the Open Web Application Security Project (OWASP), is a widely recognized document that focuses on the most critical web application security risks [4]. It serves as an essential resource for developers and organizations in identifying and addressing common vulnerabilities in their web applications. The document is regularly updated to reflect the

evolving threat landscape and provide up-to-date guidance on security best practices.

OWASP Top Ten provides valuable insights into the evolving landscape of web application security and helps prioritize security measures to protect against potential threats. It is based on extensive research and analysis by a large community of experts from around the world who contribute their knowledge and expertise to the OWASP project. The OWASP Top Ten is regularly updated to reflect emerging threats and vulnerabilities in web applications, ensuring that developers and security professionals stay informed about the latest risks. By following the recommendations outlined in the OWASP Top Ten, organizations can proactively address security weaknesses and mitigate the potential impact of attacks on their web applications.

Risk Categories

The OWASP Top Ten consists of ten categories, each representing a different type of security risk. This section aims to provide an overview of each category and discuss key aspects, common vulnerabilities, and recommended practices to address them. For a comprehensive understanding, please consult the OWASP website and relevant OWASP publications [4].

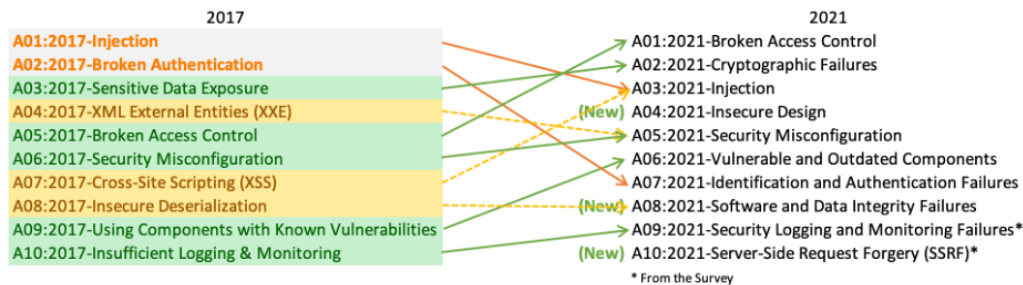| 2017	2021
A01:2017-Injection	A01:2021-Broken Access Control
A02:2017-Broken Authentication	A02:2021-Cryptographic Failures
A03:2017-Sensitive Data Exposure	A03:2021-Injection
A04:2017-XML External Entities (XXE)	(New) A04:2021-Insecure Design
A05:2017-Broken Access Control	A05:2021-Security Misconfiguration
A06:2017-Security Misconfiguration	A06:2021-Vulnerable and Outdated Components
A07:2017-Cross-Site Scripting (XSS)	A07:2021-Identification and Authentication Failures
A08:2017-Insecure Deserialization	(New) A08:2021-Software and Data Integrity Failures
A09:2017-Using Components with Known Vulnerabilities	A09:2021-Security Logging and Monitoring Failures*
A10:2017-Insufficient Logging & Monitoring	(New) A10:2021-Server-Side Request Forgery (SSRF)*
	* From the Survey

Figure 3.10: OWASP Top Ten web application security risks [4].

As shown in Figure 3.10, OWASP Top Ten 2021 categories include three new categories, four changes, and some consolidation.

1. **Broken Access Control**

   Access control is a policy that prevents users from acting outside their intended permissions, preventing unauthorized information disclosure, modification, or destruction. Common vulnerabilities include violating the principle of least privilege, bypassing access control checks, allowing viewing or editing someone else's account, accessing APIs with missing access controls, elating privileges, metadata manipulation, CORS

misconfiguration, and forcing browsing to authenticated or privileged pages [4]. These vulnerabilities can lead to unauthorized access to data, business functions, and user privileges.

2. **Cryptographic Failures**

Cryptographic failures refer to vulnerabilities related to the incorrect or insecure implementation of cryptographic algorithms, protocols, and functions in web applications. These vulnerabilities can result in the compromise of sensitive information, such as passwords, personal data, or encryption keys [4]. This may enable attackers to exploit them to bypass protection, exposing sensitive data, causing breaches, identity theft, and unauthorized access.

3. **Injection**

Injection attacks involve sending untrusted data to an interpreter, tricking it into executing unintended commands, or accessing unauthorized data through a command or query [4]. Injection attacks can target various types of interpreters, such as SQL databases, NoSQL databases, operating systems, or even web browsers. The consequences of successful injection attacks can be severe, including unauthorized data access, data loss, data manipulation, or even complete system compromise.

4. **Insecure Design**

Insecure Design refers to vulnerabilities that stem from poor design choices, inadequate security controls, and flawed architectural decisions during the development of web applications [4]. These design weaknesses can introduce significant security risks and make applications more susceptible to attacks.

One of the key aspects of Insecure Design is the failure to properly implement security controls and mechanisms [35]. This includes issues such as lack of input validation, improper session management, inadequate authentication and authorization mechanisms, and ineffective error handling.

5. **Security Misconfiguration**

Security Misconfiguration refers to the improper configuration of various security controls within a web application [4]. It includes instances where default configurations, insecure configurations, or incomplete configurations are present, leaving the application vulnerable to potential attacks. These misconfigurations can occur in various layers of the

application stack, such as the web server, application server, database, framework, or platform [36]. Security misconfigurations can be introduced during development, deployment, or when applying patches or updates to the application.

6. **Vulnerable and Outdated Components**

Vulnerable and Outdated Components focuses on the risks associated with using third-party software components, libraries, and frameworks that contain known vulnerabilities or have not been updated to include necessary security patches. By using outdated or vulnerable components, applications may become susceptible to various attacks, including remote code execution, data breaches, or compromise of the entire system [4, 37].

7. **Identification and Authentication Failures**

Identification and Authentication Failures refers to the vulnerabilities that allow attackers to bypass or compromise the user identification and authentication processes. Inadequate authentication can result from applications allowing automated attacks, weak or ineffective credential recovery, using plain text, encrypted, or weakly hashed passwords, missing or ineffective multi-factor authentication, exposing session identifiers, reusing session identifiers after successful login, incorrectly invalidating session IDs, and not properly invalidating user sessions or authentication tokens during logout or inactivity [4, 38]. These vulnerabilities can lead to unauthorized access, account takeover, and privilege escalation, exposing sensitive data and compromising the security of the application.

8. **Software and Data Integrity Failures**

Software and Data Integrity Failures occur when malicious actors exploit vulnerabilities in software systems or data stores, allowing them to modify, delete, or compromise the integrity of the software or data. These vulnerabilities can lead to unauthorized access, data breaches, or manipulation of critical systems, resulting in financial losses, reputational damage, and legal consequences [4, 39].

9. **Security Logging and Monitoring Failures**

Security logging and monitoring are crucial for the early detection of cyber threats and data breaches [4, 40]. Without proper systems, businesses can face risks such as recording login attempts, not backing

up or storing logs locally, improperly backed up logs, lacking real-time monitoring systems, missing monitoring and alerting systems, and logs not protected for integrity. Recording login attempts helps verify who logged in, tracks hosts causing unintentional logins, and mitigates breaches. Proper logging levels ensure the most important logs are backed up, while a central system or SIEM (Security Information Event Management) provides additional protection. Monitoring systems in real-time, such as SIEM, help prevent attacks and analyze network infrastructure events. Businesses should ensure that all necessary systems are configured correctly and log to the correct central point. Additionally, logs should be protected for integrity to prevent failing audits and compliance regulations, making them inadmissible as evidence for law enforcement agencies [40].

10. **Server-Side Request Forgery (SSRF)**

Server-Side Request Forgery (SSRF) refers to a vulnerability that allows an attacker to make requests from a web application to another internal or external server. The attacker can abuse this vulnerability to interact with resources that should typically be inaccessible to them, such as internal systems, cloud metadata, or external services [4, 41]. The impact of SSRF can be severe, as it enables attackers to bypass firewalls and access sensitive information, launch attacks against internal systems, or perform reconnaissance to gather valuable data. Some examples of potential SSRF exploits include reading local files, initiating port scans, accessing internal APIs, or making requests to other vulnerable systems [41].

3.2 Methodology Steps

The next goal is to identify specific vulnerabilities within the Yii2 framework that align with the categories outlined in the OWASP Top Ten. This analysis involves several steps such as planning, collecting and analyzing data to identify the flaws, and then utilizing CyPROM and OWASP Top Ten to evaluate Yii2-based projects.

3.2.1 Initial Planning

The first step in the methodology involves establishing a clear plan and defining the scope and objectives. We conducted a comprehensive literature review to understand the existing knowledge in the field of web application

security, including the OWASP Top Ten vulnerabilities and their relevance to the Yii2 framework. This step established a solid foundation and ensured that our work contributes to existing knowledge.

3.2.2   Data Collection and Analysis

Once the initial planning phase was complete, necessary data was collected in the next phase. This stage involved obtaining the necessary resources and data required for the analysis. The information about web applications, including their architecture, components, and security features, comes from publicly available sources such as vulnerability databases [42], security forums, OWASP Top Ten, and so on. Additionally, several real-world examples and case studies of web application vulnerabilities had been collected from web developers' experience and official documents from Yii2 [31]. This diverse range of data ensures a comprehensive understanding of the vulnerabilities present in web applications.

This stage involves multiple steps to ensure a comprehensive and reliable dataset. Firstly, a systematic literature review was conducted to identify existing knowledge of web application vulnerabilities. This review served as a foundation for understanding the current state of the field and identifying any gaps in knowledge that needed to be addressed. Following the literature review, most of the vulnerabilities were collected from the OWASP Top Ten descriptions, vulnerability databases, or any reliable security forums. As shown in Figure 3.11 and Figure 3.12, many issues along with their state had been raised in the community.
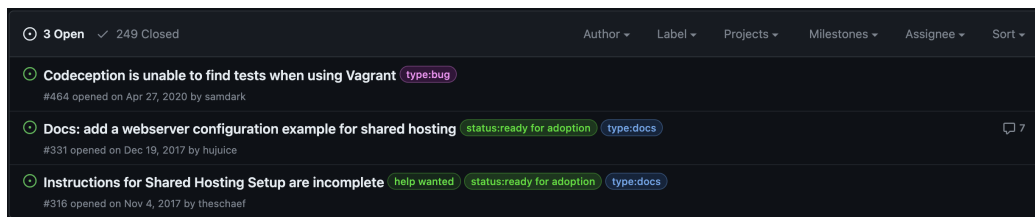

Figure 3.11: Example of Yii2 open issues on GitHub.

These issues might be related to web applications or any other platforms and had not been classified yet. This thesis focuses on frameworks for developing web applications (Yii2 in this case), therefore, the chosen vulnerabilities should satisfy some conditions:

1. **Relevance to web application frameworks:** Ensures the vulnerabilities align with the frameworks.

36

Vuln ID ⚖	Summary ❶	CVSS Severity ⚖
CVE-2021-32494	Radare2 has a division by zero vulnerability in Mach-O parser's rebase_buffer function. This allow attackers to create malicious inputs that can cause denial of service. **Published:** July 07, 2023; 3:15:09 PM -0400	*V3.x:*(not available) *V2.0:*(not available)
CVE-2023-36256	The Online Examination System Project 1.0 version is vulnerable to Cross-Site Request Forgery (CSRF) attacks. An attacker can craft a malicious link that, when clicked by an admin user, will delete a user account from the database without the admin's consent. The email of the user to be deleted is passed as a parameter in the URL, which can be manipulated by the attacker. This could result in a loss of data. **Published:** July 07, 2023; 2:15:09 PM -0400	*V3.x:*(not available) *V2.0:*(not available)
CVE-2021-33798	A null pointer dereference was found in libpano13, version libpano13-2.9.20. The flow allows attackers to cause a denial of service and potential code execute via a crafted file. **Published:** July 07, 2023; 2:15:09 PM -0400	*V3.x:*(not available) *V2.0:*(not available)
CVE-2021-33796	In MuJS before version 1.1.2, a use-after-free flaw in the regexp source property access may cause denial of service. **Published:** July 07, 2023; 2:15:09 PM -0400	*V3.x:*(not available) *V2.0:*(not available)
CVE-2023-3544	A vulnerability was found in GZ Scripts Time Slot Booking Calendar PHP 1.8. It has been declared as problematic. This vulnerability affects unknown code of the file /load.php. The manipulation of the argument first_name/second_name/phone/address_1/country leads to cross site scripting. The attack can be initiated remotely. The identifier of this vulnerability is VDB-233296. NOTE: The vendor was contacted early about this disclosure but did not respond in any way. **Published:** July 07, 2023; 1:15:10 PM -0400	*V3.x:*(not available) *V2.0:*(not available)

Figure 3.12: Example of Common Vulnerabilities and Exposures (CVE) descriptions.

2. **Severity:** Prioritize vulnerabilities with higher severity levels. These vulnerabilities pose a greater risk to the security of web applications and are likely to have a more significant impact.

3. **Exploitability:** Consider vulnerabilities that have known methods of exploitation. This helps to analyze the potential impact and provide recommendations for mitigation.

4. **Frequency:** Focus on vulnerabilities that occur frequently in real-world scenarios. These vulnerabilities are more likely to be encountered during web application development (using Yii2).

5. **Practicality:** Choose vulnerabilities that can be practically addressed or mitigated within the context of the framework. This will enable to provide meaningful recommendations and solutions for developers working with the framework.

After examining each vulnerability and assessing its potential impact on the Yii2 framework, the vulnerabilities identified would be categorized according to the OWASP Top Ten list. Quantitative and qualitative analysis techniques [45] are used in this process.

These techniques helped in determining the severity of each vulnerability and prioritizing them based on their potential impact. The categorization according to the OWASP Top Ten list provided a standardized framework for addressing and mitigating the identified vulnerabilities in the Yii2 framework.

Quantitative techniques were used to summarize, interpret, and identify the most common vulnerabilities within the Yii2 framework according to the OWASP Top Ten list, while qualitative analysis included thematic coding and content analysis of developer interviews and uncovering issues in the Yii community [44] and NVD [42]. These techniques allowed for a comprehensive understanding of the vulnerabilities present in the Yii2 framework and helped prioritize the mitigation efforts. By utilizing both quantitative and qualitative analysis, developers could gain insights into the root causes of vulnerabilities and make informed decisions on how to address them effectively. Additionally, this approach enabled the Yii community to collaborate and share knowledge in order to continuously improve the security of the framework.

The outcomes of these techniques was carefully examined to identify recurring themes, patterns, and suggestions for improving the ability to address security vulnerabilities. These qualitative insights provided valuable context and depth to the findings and ensured a more comprehensive understanding of the research problem.

The thesis used the OWASP Top Ten knowledge base to identify and prioritize significant vulnerabilities in web applications. It allowed trainees to tailor their approach and focus on the most prevalent vulnerabilities within the Yii2 framework. This proposed mitigation strategies to enhance the overall security of Yii2-based web applications. The findings could be valuable resources for developers and security professionals in understanding and mitigating potential risks in web applications.

Table 3.1 shows the possible Yii2 vulnerabilities matched to OWASP Top Ten security risks.

Table 3.1: Mapping the OWASP Top Ten categories to Yii2 issues.

OWASP Top Ten	Yii2 Framework Issues
Broken Access Control	❖ Violation of the principle of least privilege or deny by default, where access should only be granted for particular capabilities, roles, or users, but is available to anyone. ❖ Accessing API with missing access controls for POST, PUT and DELETE. ❖ Elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user.
Cryptographic Failures	❖ Any old or weak cryptographic algorithms or protocols are used either by default or in older code. Currently, Yii2 uses a hash with random encrypted salt to avoid this. ❖ Lack of cryptographic algorithms for pid [43].
Injection	❖ Currently, Yii2 provides various methods to protect against injection attacks (using Active Record), and XSS attacks (using the HtmlPurifier library).

Insecure Design	❖ Status of a record in database. For instance, a user is disabled or deleted but still acts as an active one or appears in order details, and so on. ❖ The E-commerce web application allows group booking discounts and has a maximum of fifteen items before requiring a deposit. Attackers could threat model this flow and test if they could book a hundred items and all available at once in a few requests, causing a massive loss of income.
Security Misconfiguration	❖ The security settings in the application servers, application frameworks, libraries, databases, etc., are not set to secure values. ❖ Unnecessary features, authority are enabled. For instance, every user can edit other user details, anyone can access the login-required page.
Vulnerable and Outdated Components	❖ Most libraries are trusted, maintained, built by Kartik, and compatible with different versions of Yii2, including support for Bootstrap 3, 4, and 5.

Identification and Authentication Failures	❖ Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords. ❖ Permits brute force or other automated attacks. ❖ Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin". ❖ Uses plain text, encrypted, or weakly hashed passwords data stores. Currently, Yii2 uses a hash with random encrypted salt to avoid this.
Software and Data Integrity Failures	❖ The libraries or data are from unexpected sources and/or had not been verified. ❖ Yii2 provides safe methods for passing data, especially when dealing with raw queries or query builders.
Security Logging and Monitoring Failures	❖ Auditable events, such as logins, failed logins, and high-value transactions, are not logged ❖ The application cannot detect, escalate, or alert for active attacks in real-time or near real-time. ❖ Warnings and errors generate no, inadequate, or unclear log messages. ❖ Logs of applications and APIs are not monitored for suspicious activity.

Server-Side Request Forgery	❖ Sensitive data exposure – Attackers can access local files or internal services to gain sensitive information. Currently, Yii2 does not allow access to local files or directories. ❖ Port scan internal servers – If the network architecture is unsegmented, attackers can map out internal networks and determine if ports are open or closed on internal servers from connection results or elapsed time to connect or reject SSRF payload connections. ❖ An attacker can craft a malicious link. When clicked by an admin user, will delete a user account from the database without the admin's consent. The email or ID of the user to be deleted is passed as a parameter in the URL, which can be manipulated by the attacker. This could result in a loss of data. ❖ The attacker can abuse internal services to conduct further attacks such as Remote Code Execution (RCE) or Denial of Service (DoS).

3.2.3 Utilizing CyPROM and OWASP Top Ten to Evaluate the Yii2 PHP Framework

With a clear understanding of CyPROM, Yii2, and the OWASP Top Ten in Section 3.1, and a comprehensive analysis that maps the OWASP Top Ten to the malicious issues in Section 3.2.2, this stage aims to develop a plan to mitigate these flaws identified in the Yii2 framework, thus improving the overall security of web applications developed. The following steps are undertaken to achieve this objective:

1. **Building a Representative Sample of Yii2 Web Applications**

   In order to evaluate the effectiveness of CyPROM in mitigating web application vulnerabilities in the Yii2 framework, it is essential to start by building a representative sample of Yii2 web applications.

   To build the sample, there are various methods such as:

   ❖ Searching for open-source Yii2 applications on platforms like GitHub and GitLab.

   ❖ Collaborating with Yii2 developers and communities to acquire existing projects.

   ❖ Developing custom Yii2 applications that mimic real-world scenarios.

   In this thesis, a custom web application is developed with the collaboration of Yii2 developers. The objective is to build a sample Yii2 Advanced project that serves as a representation of typical projects encountered in the industry (shown in Figure 3.13).
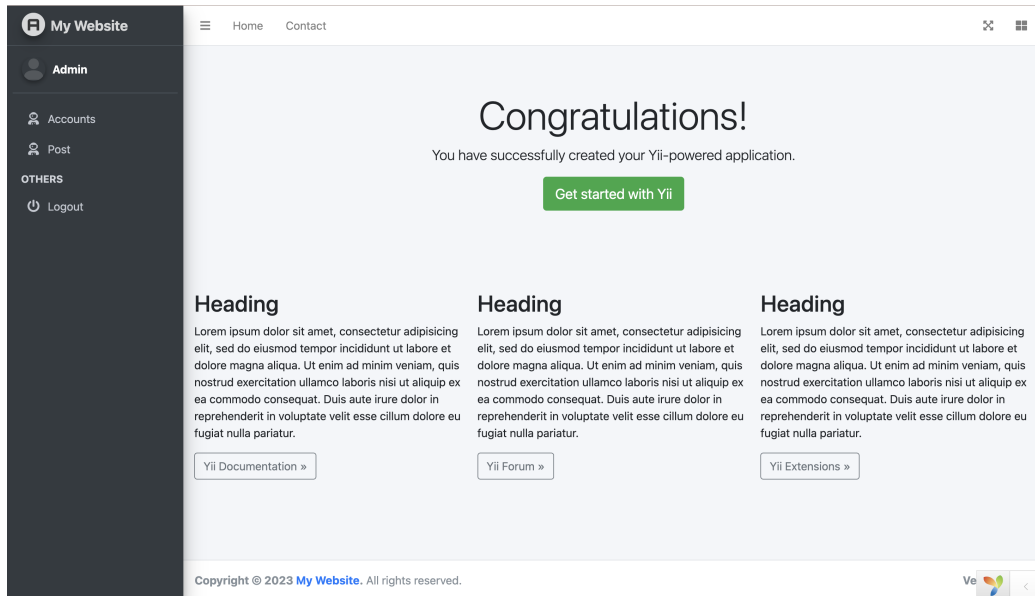

Figure 3.13: Sample of the Yii2 Advanced template project.

Similar to many projects initiated by inexperienced or novice developers, this sample project lacks essential configurations and fails to implement robust security measures to counter vulnerabilities. The primary motivation behind this omission is the prevailing mindset among such

developers, who prioritize expeditious project completion in order to quickly generate revenue or meet other personal objectives.

This thesis serves as a comprehensive exploration of the implications and consequences that arise from prioritizing speed over quality in software development projects. By focusing on the shortcomings of this sample Yii2 project, it highlights the criticality of adopting best practices, such as implementing proper configurations and fortifying defenses against potential vulnerabilities.

Through an in-depth analysis of this project's limitations, the research aims to shed light on the potential risks and pitfalls associated with neglecting crucial aspects of development. It underscores the importance of developer education and experience in cultivating a thorough understanding of secure coding practices, emphasizing the need to strike a balance between efficiency and software integrity.

Ultimately, by examining the challenges faced by this sample Yii2 project, this thesis provides valuable insights for both aspiring developers and industry professionals, serving as a reminder of the indispensable role that expertise and conscientiousness play in the successful execution of software development endeavors.

2. Selecting Vulnerabilities that are Implementable as CyPROM Actions

Once a representative sample of Yii2 web applications is established, the next step is to identify vulnerabilities that can be implemented as CyPROM actions. This process involves carefully examining the identified vulnerabilities and determining their suitability for CyPROM implementation. In order to choose vulnerabilities for CyPROM actions, several factors should be considered.

Firstly, the vulnerabilities in web applications should represent common security weaknesses exploited by attackers and developers in real-world scenarios. They should reflect the types of security weaknesses commonly encountered by developers and attackers when working with Yii2 web applications. By focusing on vulnerabilities that are frequently exploited or encountered, trainees can enhance their understanding of the security risks associated with Yii2 and develop relevant mitigation strategies.

Besides, these flaws should align with the capabilities and features of CyPROM. CyPROM is designed to provide practical cybersecurity training, and therefore, the selected vulnerabilities should be suit-

able for implementing interactive training actions and scenarios. This requires evaluating the practicality of simulating the vulnerabilities within the training environment and ensuring that they can be effectively demonstrated and addressed.

Furthermore, the chosen vulnerabilities should cover a range of security issues to provide comprehensive training. By addressing a diverse set of security weaknesses, trainees can broaden their knowledge and skills in securing Yii2 applications effectively.

By considering the mapping of the OWASP Top Ten to Yii2 issues and selecting vulnerabilities that satisfy the requirements mentioned above, the implementation of CyPROM actions and scenarios can effectively address common security weaknesses, simulate real-world attack scenarios, and provide developers with relevant training specific to Yii2 framework.

3. Performing CyPROM Assessment on the Yii2 Framework

The next step is to perform a CyPROM assessment on the Yii2 framework based on a representative sample of Yii2 web applications and the selected vulnerabilities. CyPROM assessment involves applying the defined CyPROM actions to the sample applications to identify vulnerabilities and assess their severity. This will help to identify the common security flaws and weak points in the Yii2 framework and allow for the development of strategies to mitigate them. The assessment will also provide insights into how to improve the security of existing and future Yii2 applications.

4. Analyzing the Evaluation Results

Once the CyPROM assessment is complete, the evaluation results need to be analyzed to gain insights into the vulnerabilities present in the Yii2 framework and their impact on the sample applications. This analysis enables a deeper understanding of the security weaknesses and guides the subsequent mitigation planning.

5. Proposing a Mitigation Plan for the Yii2 Framework

As a result of the assessment, a mitigation plan can be recommended in order to address the identified vulnerabilities in a cost-effective manner. Based on the prioritized vulnerabilities and the associated risks, trainees should identify suitable mitigation measures for each vulnerability. These measures may include code changes, configuration adjustments, security controls implementation, and best practices adoption.

For each mitigation measure, detailed implementation guidelines can be found in User Guide or security forums, Yii2 official documents, and so on. These guidelines will outline the specific steps required to implement the mitigation measure effectively. It is also crucial to provide trainees with practical instructions to ensure proper implementation and adherence to security best practices.

Last but not least, the mitigation plan will be documented, including all measures implemented, guidelines, and results of testing. Moreover, training materials and training sessions are developed to educate developers and stakeholders about the importance of web application security and specific mitigation measures implemented in the Yii2 framework.

3.3 Approach Significance

3.3.1 Adherence to Cybersecurity Practices

In the realm of enhancing web application security, unwavering commitment to cybersecurity practices holds the utmost significance. Throughout the methodology, the implementation of well-established cybersecurity principles would be diligently upheld. These principles revolved around preserving the fundamental tenets of confidentiality, integrity, and availability concerning all collected data.

Essentially, our approach is about protecting sensitive information from unauthorized access or disclosure, ensuring its accuracy and trustworthiness, and making it reliably accessible when needed. In addition, the approach includes best practices for trainees to interact with sensitive data, minimizing the risk of data breaches or insider threats.

3.3.2 Justification and Strengths

The chosen methodology, involving the analysis of Yii2 using CyPROM and the OWASP Top Ten, was strongly justified based on its inherent strengths and suitability for effectively addressing the research question at hand. This approach combined the specialized capabilities of CyPROM with the industry-standard OWASP Top Ten, thus establishing a robust foundation for evaluating and assessing the security posture of the Yii2 PHP Framework. By leveraging these powerful tools, the methodology enabled a systematic and comprehensive analysis of the framework's vulnerabilities and identifies specific areas for improvement, ensuring an optimized approach to enhance

CyPROM's efficacy as a hands-on training in mitigating web application vulnerabilities.

One of the most important justifications for this methodology lies in its systematic nature. By leveraging the capabilities of CyPROM, this work could perform a thorough analysis of Yii2's vulnerabilities and potential areas for improvement. CyPROM's extensive capabilities allowed for a comprehensive assessment that ensured both common and less obvious security vulnerabilities were identified. By incorporating the OWASP Top Ten and NVD, which identify the most common web application security risks, the assessment was aligned with industry best practices and focused on the most critical areas.

Moreover, the methodological approach demonstrates a clear focus on not only identifying vulnerabilities but also on finding actionable solutions to improve CyPROM's effectiveness in mitigating web application vulnerabilities. The emphasis on data collection methods, correctness assessment criteria, adherence to cybersecurity practices, and attributes for effective improvements underpinned the practical applicability of the thesis objectives. This robustness in the methodology ensured that the findings and recommendations derived from this thesis are reliable and could be readily implemented in real-world scenarios to bolster the security of Yii2 and similar web application frameworks.

Chapter 4

CyPROM Enhancement

This chapter presents our comprehensive approach to implementing hands-on training for mitigating web application vulnerabilities. This will concentrate on the process of scenario progression management, which involves creating realistic scenarios that enable learners to develop their skills progressively. By following the key steps outlined, instructors can ensure that their cybersecurity training is efficient, effective, and results-driven.

4.1 Hands-on Training Methodology

In the ever-evolving landscape of cybersecurity, web application vulnerabilities pose significant risks to organizations worldwide. Malicious actors constantly exploit security gaps in web applications, leading to data breaches, financial losses, and reputational damage. As a result, there is an increasing demand for skilled cybersecurity professionals who can effectively mitigate web application vulnerabilities and protect sensitive information.

4.1.1 Key Benefits

Hands-on training is a vital component of cybersecurity education, as it equips participants with practical skills and real-world experience. When it comes to addressing web application vulnerabilities, a structured hands-on training program is crucial for empowering learners to identify, understand, and remediate potential security risks effectively. This training not only enhances technical skills but also fosters a proactive and security-conscious mindset among cybersecurity practitioners.

On the other hand, hands-on training offers numerous positive effects for organizations looking to improve their cybersecurity posture. Some of the

key benefits include:

❖ **Improved Technical Skills**

Through immersive experiences involving real-life scenarios, trainees enhance their practical technical skills, enabling them to effectively address challenges in the real world. This comprehensive approach encompasses tasks such as recognizing vulnerabilities, deploying effective mitigation strategies, and adeptly resolving issues through efficient troubleshooting. By actively engaging in these activities, individuals develop a strong foundation of hands-on expertise that empowers them to excel in their technical pursuits.

❖ **Enhanced Critical Thinking and Problem-solving Abilities**

Trainees are challenged to think both critically and creatively as they meticulously analyze each scenario, identifying potential vulnerabilities that may arise. This process pushes them to generate innovative and effective mitigation strategies, thus refining their problem-solving skills. By actively engaging in these exercises, individuals strengthen their ability to approach complex situations with a well-rounded and strategic mindset.

❖ **Better Preparedness for Real-world Threats**

Hands-on training equips trainees with invaluable practical experience to counter real-world threats effectively. Organizations that invest in such training empower their teams to safeguard critical assets and proactively mitigate the risk of cyber attacks. By simulating realistic scenarios, trainees gain the confidence and expertise needed to respond swiftly and decisively to potential threats. This proactive approach to preparedness not only enhances the organization's overall security posture but also fosters a culture of vigilance and readiness within the workforce.

4.1.2   Identifying Training Objectives

The primary objective of this hands-on training program is to equip trainees with the necessary knowledge and skills to effectively identify and mitigate web application vulnerabilities. The training aims to enhance trainees' understanding of secure coding practices and enable them to proactively address potential security risks during the development process. By the end of the training, trainees should be able to:

1. Demonstrate a comprehensive understanding of the OWASP Top Ten vulnerabilities relevant to web application frameworks such as Yii2, and so on.

2. Identify and assess potential security threats in web application frameworks.

3. Apply secure coding practices to prevent and mitigate common web application vulnerabilities.

4. Utilize relevant security tools and frameworks effectively.

5. Foster a security-first mindset and culture among trainees, making security an integral part of the development lifecycle.

4.1.3   Conducting Hands-on Training

Developing effective web application security training demands a comprehensive blend of theoretical lectures and immersive, hands-on sessions. By employing this well-rounded approach, trainees could delve into web application security principles with a deeper understanding and practical experience, elevating the overall learning experience for the development team.

Theoretical lectures form a fundamental component of web application security training. Through these sessions, trainees gain an essential theoretical foundation, learning about the various vulnerabilities, attack vectors, and best practices for securing web applications. Instructors delve into concepts such as cross-site scripting (XSS), SQL injection, session management, and secure coding principles. Theoretical lectures equip individuals with the necessary knowledge to comprehend the underlying principles and frameworks of web application security.

However, theoretical knowledge alone is insufficient to truly grasp the intricacies of web application security. Practical hands-on sessions serve as an invaluable complement to theoretical lectures, enabling trainees to put their knowledge into action. During these sessions, individuals actively engage with realistic scenarios, applying their theoretical understanding to identify vulnerabilities, performing penetration testing, and implementing appropriate security measures. By immersing themselves in hands-on activities, trainees would gain practical experience and develop critical thinking skills necessary for effectively securing web applications.

For the purpose of providing efficient hands-on training, a variety of tactics and approaches might be used. A few of these include:

❖ **Interactive Vulnerability Scanning Exercises**

Vulnerability scanning exercises offer trainees hands-on experience in identifying weaknesses and loopholes in applications. Using CyPROM scenarios, trainees can actively scan for vulnerabilities and assess the potential risks they pose. This practical approach enhances their skills in recognizing specific security weaknesses and helps develop an eye for detail, crucial in any cybersecurity role.

❖ **Design Practical Exercises Simulating Real-world Scenarios**

Maximizing the effectiveness of cybersecurity training requires engaging in hands-on exercises that cover a variety of pertinent tasks. These exercises are carefully designed to simulate intentional cybersecurity challenges, ensuring that participants are well-prepared to face certain real-world scenarios. Trainees can identify potential vulnerabilities and apply appropriate mitigation techniques by actively identifying and addressing these scenarios. For example, trainees may be given a simulated brute-force authentication attack and asked to analyze the factors or configurations causing it. By practicing in a controlled environment, trainees can learn to recognize common brute-force tactics and develop the skills needed to respond effectively. This hands-on approach allows for immediate feedback and fosters a deeper understanding of cybersecurity best practices.
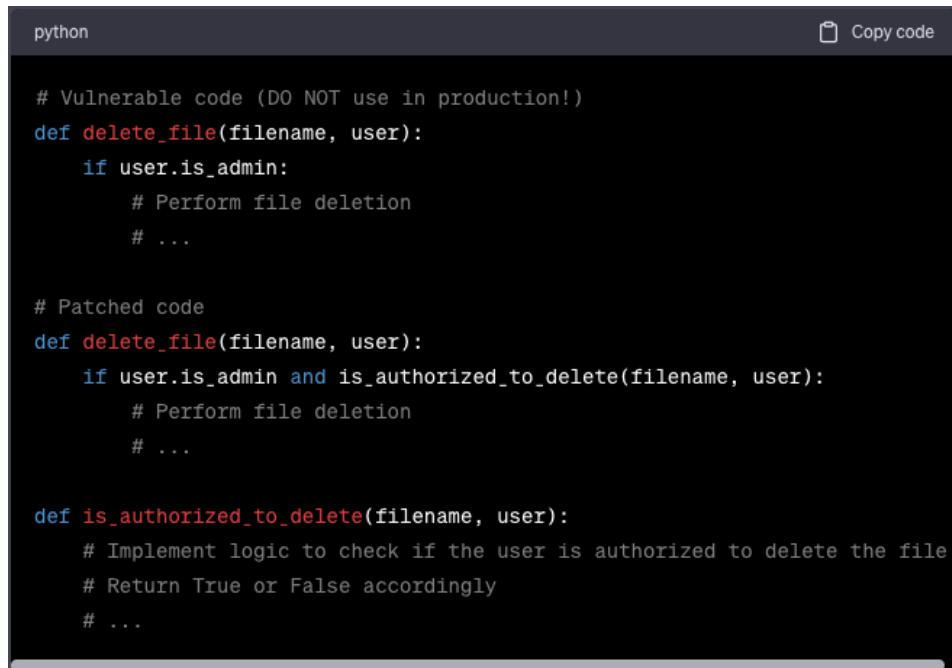
❖ **Collaborative Learning Environments**

Trainees in this hands-on training program are able to engage in collaborative exercises to address intricate cybersecurity challenges, fostering a shared learning experience and promoting effective problem-solving strategies. These exercises emulate real-world scenarios, mirroring the teamwork that cybersecurity professionals frequently employ to combat cyber threats successfully. Also, individuals can gain communication and cooperation skills while exchanging information and insights when working in groups. This collaborative environment promotes mutual learning, creating a thorough educational experience that gives trainees the knowledge and skills necessary to negotiate the steadily changing cybersecurity landscape.

❖ **Patching Exercises**

Trainees should be presented with scenarios where they are required to apply patches and updates to different applications. This process mirrors real-world situations where timely patching is crucial to safeguarding against known vulnerabilities. Through these exercises, trainees

learn the importance of prompt and effective patch management while gaining confidence in their ability to apply fixes to secure systems. An example of a patching exercise is shown in Figure 4.1.

```python
# Vulnerable code (DO NOT use in production!)
def delete_file(filename, user):
    if user.is_admin:
        # Perform file deletion
        # ...

# Patched code
def delete_file(filename, user):
    if user.is_admin and is_authorized_to_delete(filename, user):
        # Perform file deletion
        # ...

def is_authorized_to_delete(filename, user):
    # Implement logic to check if the user is authorized to delete the file
    # Return True or False accordingly
    # ...
```

Figure 4.1: Sample exercise: Access Control patching.

❖ **Feedback and Assessment**

Regular feedback and assessments are vital to gauge trainees' progress and understanding. Constructive feedback helps individuals identify areas for improvement and reinforces their learning. Assessments can take the form of quizzes, practical assessments, or even simulated red-team vs. blue-team exercises.

In conclusion, a holistic web application security training approach that integrates theoretical lectures and practical hands-on sessions is essential for cultivating a skilled and security-conscious development team. This blended learning experience empowered trainees to apply their knowledge effectively, equipping them to build robust and secure web applications that safeguard against potential threats.

4.1.4 Measuring Training Effectiveness

The final step in implementing hands-on training for mitigating web application vulnerabilities is to measure the success of the training program. This involved evaluating the effectiveness of the program in achieving the defined training objectives and assessing the impact on trainees' knowledge, skills, and behaviors. To measure success, there are various evaluation methods, including:

❖ **Knowledge Assessments**

Conducting pre-training and post-training assessments to measure participants' knowledge levels before and after the hands-on training. This provides quantitative data on the knowledge gained through the program. Conducting pre-training and post-training assessments to measure participants' knowledge levels before and after the hands-on training. This provides quantitative data on the knowledge gained through the program.

❖ **Skills Assessments**

Administering practical assessments to evaluate participants' ability to identify and mitigate web application vulnerabilities. These assessments can simulate real-world scenarios and require learners to apply their skills in a controlled environment.

❖ **Trainees' Reflection**

Collecting reflections from trainees through surveys to gather qualitative data on their perceptions of the training program. Reflection can include questions about the training's relevance, effectiveness, session summary, and overall satisfaction.

❖ **Performance Metrics**

Tracking relevant performance metrics, such as the number of vulnerabilities identified and mitigated, the time taken to resolve vulnerabilities, and the impact on the overall security posture of web applications. These metrics provide tangible evidence of the training program's effectiveness.

❖ **Incident Response Drills**

Organize incident response drills that allow learners to demonstrate their ability to handle web application security incidents effectively.

These drills should simulate real-life scenarios and involve various stake-holders, such as IT teams, developers, and management. By providing hands-on experience in a controlled environment, learners can practice identifying vulnerabilities, implementing mitigation strategies, and communicating effectively during an incident. This practical approach helps build confidence and ensures that the necessary skills are honed to respond efficiently to web application security incidents in a real-world setting.

❖ **Certification Exams**

Offer certifications upon successful completion of the training program. These certifications validate participants' skills and can be valuable for their professional growth.

In summary, measuring the success of hands-on training for web application vulnerability mitigation is a critical step in evaluating the training program's effectiveness and the participants' progress. By employing various metrics and data collection methods, trainers could gain valuable insights into the strengths and weaknesses of the program and make data-driven decisions to improve the overall training experience. Measuring success not only benefits the participants by enhancing their skills and knowledge but also ensures that organizations and institutions receive a valuable return on their investment in cybersecurity training.

4.2 Action and Scenario Implementation

To achieve the training objectives effectively, a series of actions and scenarios were implemented. This involved various hands-on exercises and real-world simulations to provide practical experience in mitigating web application vulnerabilities. As a part of this thesis, a comprehensive set of actions and scenarios were put into practice.

4.2.1 Actions

The mapping of the OWASP Top Ten categories to the newly implemented CyPROM actions is shown in Table 4.1.

Table 4.1: Mapping the OWASP Top Ten categories to CyPROM actions.

OWASP Top Ten Categories	New CyPROM Actions
Broken Access Control	❖ crawl_content ❖ brute_force_auth ❖ crawl_content_auth
Cryptographic Failures	❖ crawl_content ❖ crawl_content_auth
Injection	❖ login_sql_injection
Insecure Design	N/A
Security Misconfiguration	❖ crawl_content
Vulnerable and Outdated Components	N/A
Identification and Authentication Failures	❖ brute_force_auth
Software and Data Integrity Failures	N/A
Security Logging and Monitoring Failures	N/A

Server-Side Request Forgery	❖ scan_open_ports ❖ flood_attack

As shown in Table 4.1, it is observed that some of the OWASP Top Ten categories do not have corresponding actions implemented in CyPROM. The reason behind this absence lies in the complexity of their implementation, making it challenging to devise standardized and actionable steps within the existing CyPROM. These vulnerabilities might call for context-specific or intricate solutions that cannot be easily generalized. To effectively address these issues, a more nuanced approach is required, potentially involving a combination of multiple actions or specialized techniques beyond the current scope of CyPROM.

Below we discuss in detail each of the new CyPROM actions:

1. **crawl_content**

   - **Description:** A web scraping process that involves retrieving information from various web pages using the provided URLs.

   - **OWASP Top Ten Categories:** `Broken Access Control`, `Cryptographic Failures`, `Security Misconfiguration`.

   - **Purpose:** Evaluate Yii2 application issues:

     1. Violation of the principle of least privilege or denied by default, where access should only be granted for particular capabilities, roles, or users, but is available to anyone.

     2. Elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user.

     3. Lack of cryptographic algorithms for pid. Anyone can retrieve information from various web pages by modifying pid.

     4. Unnecessary authority is enabled. For instance, everyone can access the login-required page.

2. **brute_force_auth**

   - **Description:** A technique that involves repeatedly attempting every possible set of login information until the right one is found in order to gain unauthorized access to user accounts, passwords, or sensitive data.

- **OWASP Top Ten Categories:** `Broken Access Control`, `Identification and Authentication Failures`.

- **Purpose:** Evaluate Yii2 application issues:

  1. Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords.
  2. Permits brute force or other automated attacks.
  3. Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin".

3. **crawl_content_auth**

   - **Description:** A web scraping process that involves retrieving information from various web pages that require authentication or authorization before granting access to the provided URLs.

   - **OWASP Top Ten Categories:** `Broken Access Control`, `Cryptographic Failures`, `Server-Side Request Forgery`.

   - **Purpose:** Evaluate Yii2 application issues:

     1. Elevation of privilege. Acting as an admin when logged in as a user.
     2. Unnecessary features, and authority are enabled. For instance, every user can crawl other user details.

4. **login_sql_injection**

   - **Description:** exploits SQL Injection vulnerabilities in login systems, allowing unauthorized access to systems or applications by manipulating SQL queries and bypassing authentication mechanisms.

   - **OWASP Top Ten Categories:** `Injection`.

   - **Purpose:** Evaluate Yii2 application issues:

     1. Login SQL injection attack.

5. **scan_open_ports**

   - **Description:** The process of using various tools and techniques to scan for open ports on a network or a specific IP address.

   - **OWASP Top Ten Categories:** `Server-Side Request Forgery`.

   - **Purpose:** Evaluate Yii2 application issues:

1. Port scan internal servers – If the network architecture is unsegmented, attackers can map out internal networks and determine if ports are open or closed on internal servers from connection results or elapsed time to connect or reject SSRF payload connections.

6. **flood_attack**

   - **Description:** Overwhelm a target system or network by flooding it with an excessive amount of traffic, rendering it unable to respond to legitimate requests.

   - **OWASP Top Ten Categories:** `Server-Side Request Forgery`.

   - **Purpose:** Evaluate Yii2 application issues:

     1. The attacker can abuse internal services to conduct further attacks such as Remote Code Execution (RCE) or Denial of Service (DoS).

In addition, there are some extra actions that were employed to perform some specific tasks, such as:

❖ **cmd_exec**: Perform a command on a target machine.

❖ **find_ip_address**: Returns the IP address of a target host using the Python 3 `socket` library.

4.2.2   Scenarios

The implemented actions were used to create specific scenarios based on the OWASP Top Ten categories. There are seven new scenarios implemented:

1. **basic_cmd**

   - **Description:** Perform a command execution on a target machine without authentification.

   - **Actions:** `cmd_exec`
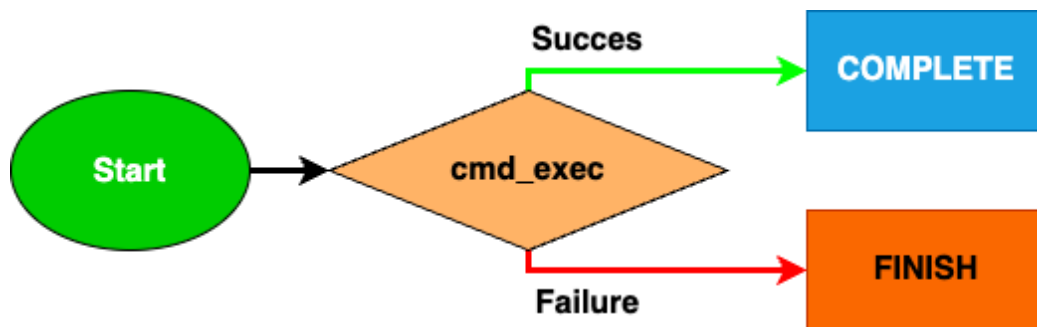
   - **OWASP Top Ten Category:** `N/A`.

Figure 4.2: Flowchart of `basic_cmd` scenario.

2. authen_failures

- **Description:** Identify and counter automated attacks, such as credential stuffing, brute force attempts, and the use of default, weak, or well-known passwords.

- **Actions:** `brute_force_auth`

- **OWASP Top Ten Category:** `Identification and Authentication Failures`.
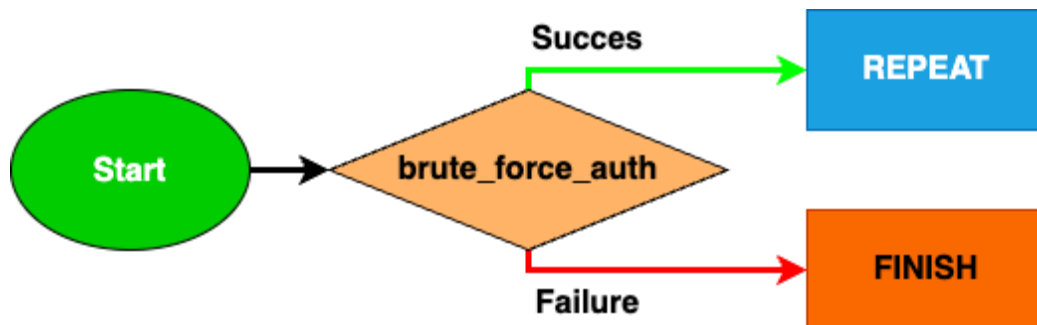

Figure 4.3: Flowchart of `authen_failure` scenario.

3. sql_injection

- **Description:** Exploit vulnerabilities in the login system by inserting malicious SQL code into the input fields, potentially gaining unauthorized access to the database or the application.

- **Actions:** `login_sql_injection`
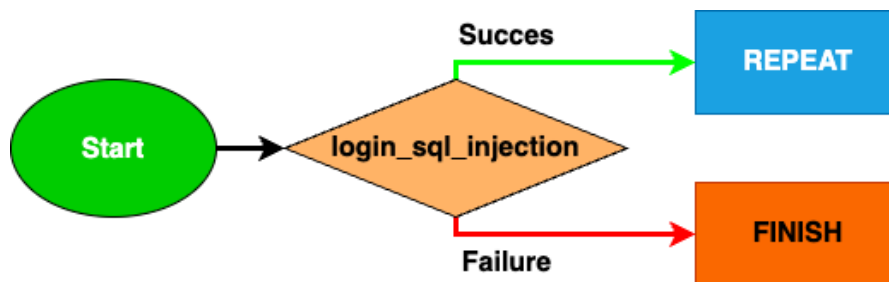
- **OWASP Top Ten Category:** `Injection`.

Figure 4.4: Flowchart of `sql_injection` scenario.

4. broken_access

- **Description:** Identify unauthorized access to restricted resources or features due to inadequate or faulty access controls in the system.

- **Actions:** `brute_force_auth`, `crawl_content`, `crawl_content_auth`

- **OWASP Top Ten Category:** `Broken Access Control`.
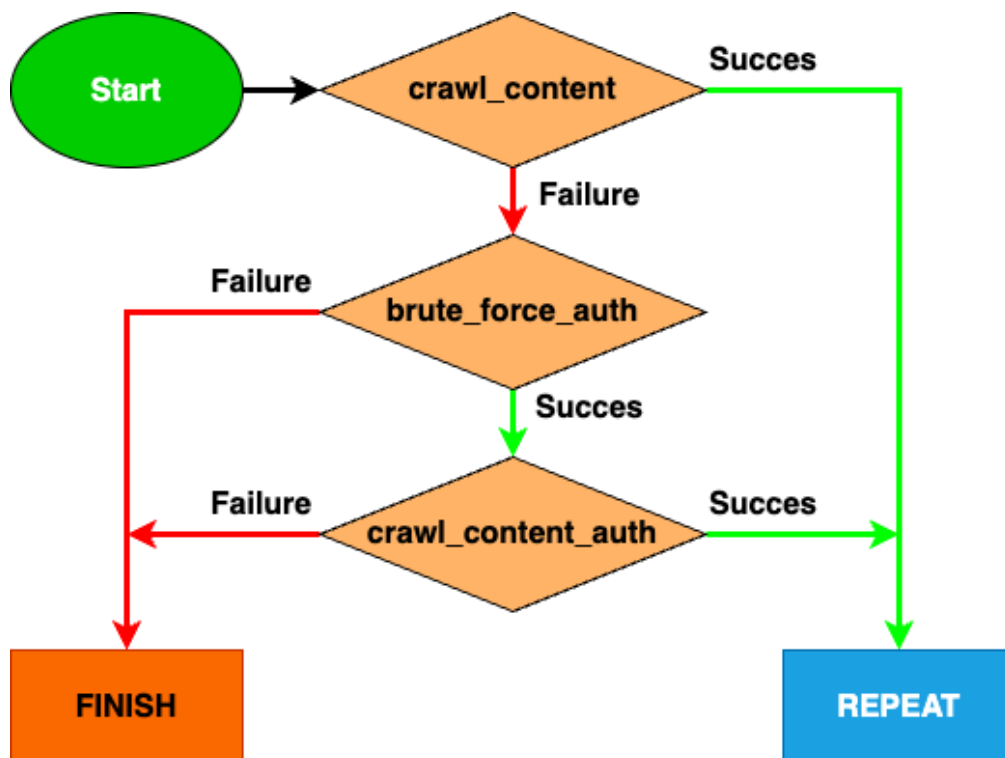

Figure 4.5: Flowchart of `broken_access` scenario.

5. crypto_failures

- **Description:** Identifying issues includes the use of outdated or weak cryptographic algorithms or protocols, either as default settings or within older code, as well as the absence of cryptographic algorithms.

- **Actions:** `brute_force_auth`, `crawl_content_auth`
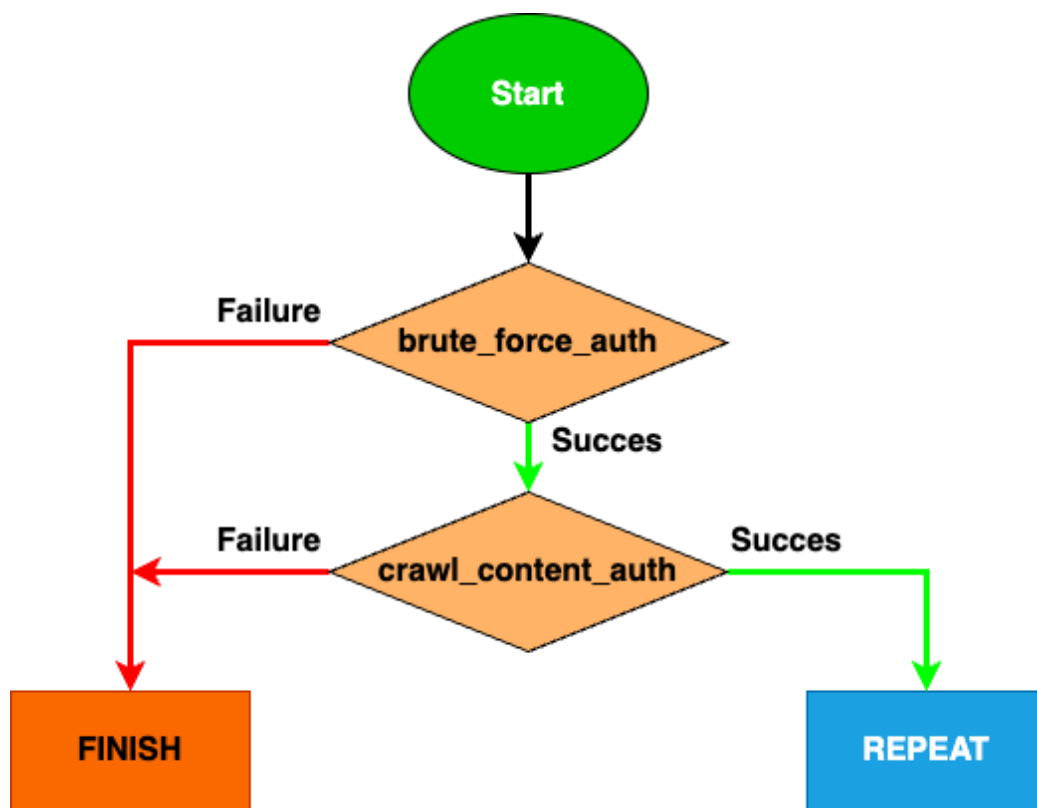
- **OWASP Top Ten Category:** `Cryptographic Failures`.

Figure 4.6: Flowchart of `crypto_failure` scenario.

6. flood_attack

- **Description:** Send a large volume of malicious data packets to a target system, overwhelming its resources and causing disruption or denial of service to legitimate users.

- **Actions:** scan_open_ports, flood_attack

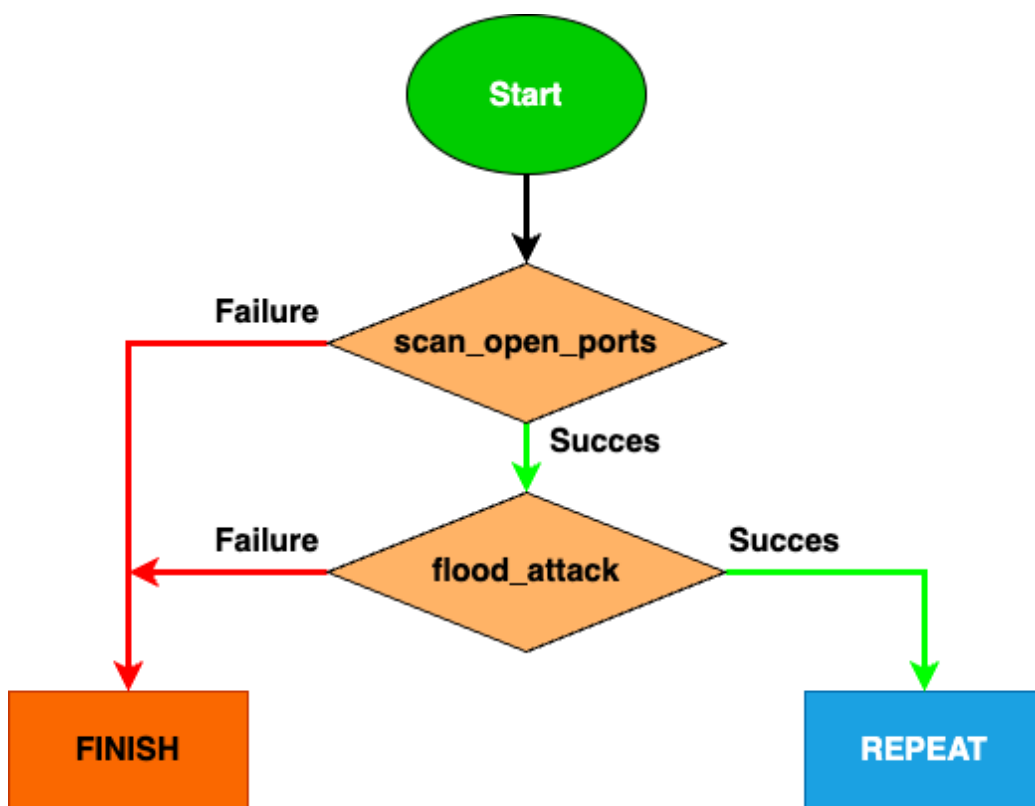- **OWASP Top Ten Category:** Server-Side Request Forgery (SSRF).

Figure 4.7: Flowchart of flood_attack scenario.

7. yii2_evaluation

A scenario including all actions was implemented to evaluate Yii2-based applications, as represented in Figure 4.8. The implemented scenario aims to assess the resilience and security of Yii2-based applications through a comprehensive set of steps. It simulated real-world usage scenarios to thoroughly test and evaluate application performance, functionality, and usability in a practical environment. The structure of the scenario included several steps, each representing a specific security challenge and corresponding action. It is noteworthy that the action "crawl_content_auth" occurred twice in the scenario. This action occurred in two separate steps, "Broken_Access_Control" and "Crypto_Failure". Although both steps involved the same action, their context, and outcomes differed as they addressed different security issues. In the context of "Broken_Access_Control", the "crawl_content_auth" action aimed to assess vulnerabilities related to non-functioning access controls, while in the context of "Crypto_Failure", the same action focused on assessing cryptographic vulnerabilities. This duplication allowed for a more comprehensive examination of the applications and provided valuable insight into potential security vulnerabilities from multiple perspectives. Throughout the evaluation process, the scenario effectively uncovered vulnerabilities and weaknesses and helped improve the overall security posture of the framework.

These scenarios served as unit tests or specific hands-on exercises. The purpose of these unit tests or hands-on exercises was to assess the effectiveness of the implemented actions in mitigating the vulnerabilities associated with each OWASP Top Ten category.

In summary, by engaging in hands-on activities and immersive simulations, trainees would gain practical skills, honing their ability to identify and mitigate potential threats. The carefully designed scenarios would mimic real-life situations, empowering learners to apply their knowledge in a risk-free environment and develop confidence in handling security challenges.
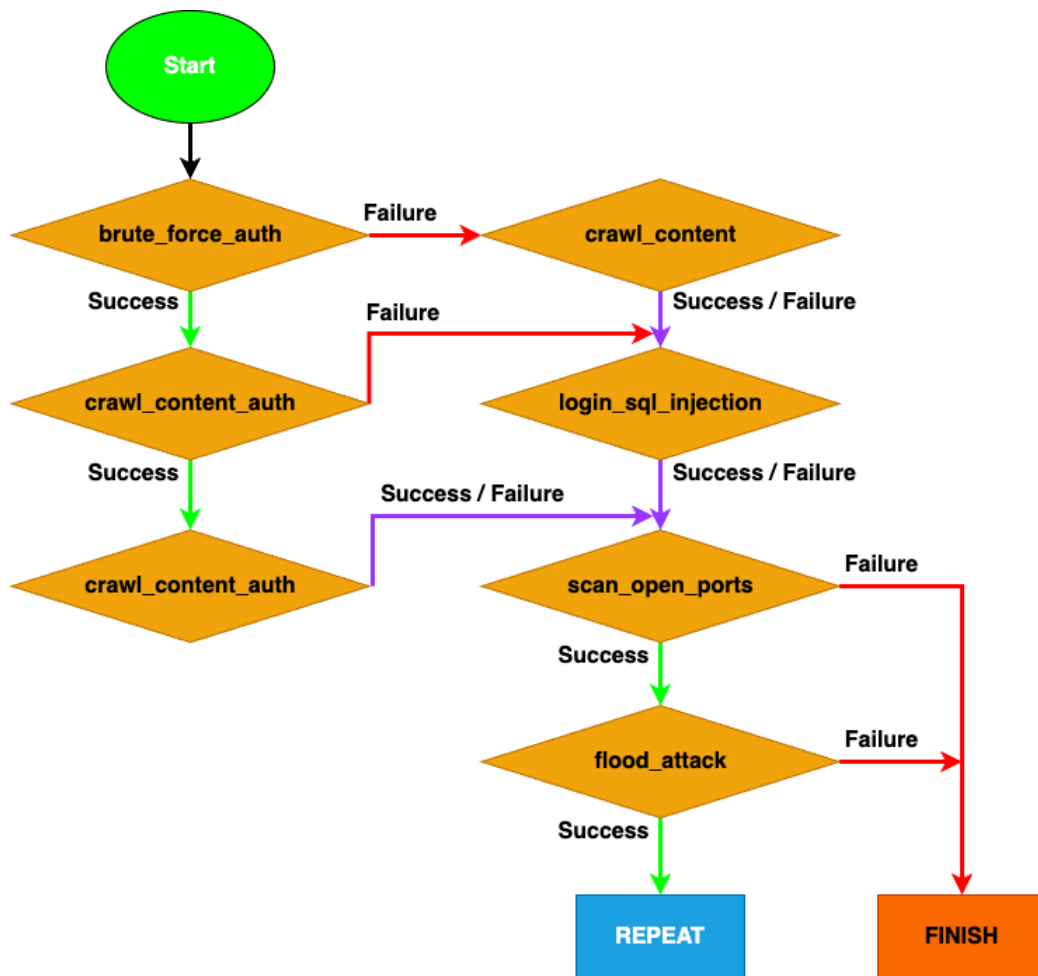
Figure 4.8: Flowchart of yii2_evaluation scenario.

Chapter 5

Evaluation

This research aims to design and implement a hands-on training program to mitigate web application vulnerabilities. And then, several comprehensive evaluations will assess its effectiveness, targeting specific aspects of impact and value.

5.1 Functionality Evaluation

The functionality evaluation aims to comprehensively test the actions and scenarios within the enhanced CyPROM. Also, this evaluation ensures that CyPROM effectively identifies vulnerabilities that are carefully selected to represent real-world threats commonly encountered by web applications. The evaluation process will involve the creation of simulated attack scenarios against web applications, allowing CyPROM to assess its ability to detect and mitigate these threats. In this thesis, the expertise gained from [31] and [46] will be employed to guide the evaluation approach.

The evaluation process of CyPROM involves the strategic design of scenarios aimed at simulating potential attacks, aligning with the OWASP Top Ten vulnerabilities list and Table 3.1. Through this approach, CyPROM can effectively gauge its ability to accurately identify and proactively counter the wide array of risks web applications face. By actively detecting, analyzing, and responding to these simulated threats using carefully crafted actions and scenarios, CyPROM ensures its readiness to safeguard against potential security breaches. The results obtained from these simulations are diligently monitored and recorded in our database, enabling us to comprehensively assess the effectiveness and unwavering reliability of CyPROM's enhanced features throughout this robust evaluation process.

Table 5.1 contains assessment criteria utilized to evaluate the actions and

scenarios implemented in this enhancement. The pass criteria are typically the minimum acceptable conditions that must be met for a certain aspect to be considered satisfactory or successful. The checkmarks (✓) were employed to indicate the pass criteria for each metric. Keep in mind that the list of assessment criteria was prepared using [31, 46] and Table 3.1, and it is consistent with CyPROM and web application capabilities and functionality.

Table 5.1: CyPROM enhancement evaluation for each OWASP Top Ten category.

OWASP Top Ten Categories	Assessment Criteria	Status
Broken Access Control	❖ Build strong access controls using role-based authentication mechanisms ✓ ❖ Except for public resources, deny default access to features ✓ ❖ Rate limit API and controller access ✓ ❖ Maintain lean servers by shutting down unnecessary services, deleting inactive and unnecessary accounts ✓ ❖ Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record ✓	**5/5** **Passed**
Security Logging and Monitoring Failures	N/A	-

Cryptographic Failures	❖ Encrypt all data at rest using secure and robust encryption algorithms, keys, and protocols ✓ ❖ Store passwords using robust, adaptive, and proven hashing functions ✓ ❖ Always use authenticated encryption instead of just encryption ❖ Verify independently the effectiveness of configuration and settings ✓	**3/4** **Passed**
Injection	❖ Use positive server-side input validation ✓ ❖ Use safe APIs to avoid interpreters completely ✓ ❖ Use intrusion detection systems to spot suspicious behavior ❖ Use parameterized queries ✓ ❖ Use LIMIT and other SQL controls within queries, preventing mass disclosure of records ✓ ❖ For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter ✓	**5/6** **Passed**
Insecure Design	N/A	-

Security Mis-configuration	❖ Remove unused features and services and deploy an application with minimal setup ❖ Use preconfigured templates (with different credentials) ✓	**1/2** **Passed**
Vulnerable and Outdated Components	N/A	-
Server-Side Request Forgery	❖ Enforce user-input validation and sanitization ✓ ❖ Remote resource access features, if any, must be isolated in a separate impact ❖ Block unwanted incoming traffic using deny-by-default firewall policies ✓ ❖ Ensure clients don't get raw responses ❖ Build a positive allow list for port, destination, and URL schema ✓ ❖ Disallow HTTP redirections ✓	**4/6** **Passed**

Identification and Authentication Failures	❖ Implement multi-factor authentication to prevent automated credential stuffing, brute force, and stolen credential reuse attacks ❖ Don't use default credentials, especially for admin privileges ✓ ❖ Implement a strong password policy ✓ ❖ Monitor failed login attempts and set limits and delays on the same ✓ ❖ Strengthen registration, credential recovery, and other authentication-related processes ✓ ❖ Implement weak password checks, such as testing new or changed passwords against the top 10,000 worst passwords list ✓ ❖ Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login	**5/7** **Passed**
Software and Data Integrity Failures	N/A	-

5.2 Comparative Analysis

The purpose of this section is to conduct a comprehensive comparative analysis of the improved version of CyPROM and its original equivalent. The thesis intends to gain valuable insights into the effectiveness of the upgraded version's enhancements by evaluating critical aspects such as the number

of actions and scenarios implemented, as well as the overall security posture achieved. Through careful examination and linking these key factors together, a clear understanding of how the improved CyPROM performs in comparison to its predecessor will be attained.

5.2.1 Action and Scenario Coverage

To conduct a comprehensive comparison of the two versions of CyPROM, it is necessary to assess the number of actions and scenarios implemented in each version. This evaluation will help to visualize how the two CyPROM versions have changed over time.

As shown in Table 5.2, there was a noticeable increase in the number of actions from CyPROM v0.1 to the current version. The "N/A" label denotes that a particular action is not available in the respective version of CyPROM. As can be seen, the enhanced CyPROM version has expanded coverage by introducing new actions, such as `brute_force_auth`, `cmd_exec`, `crawl_content`, `crawl_content_auth`, `login_sql_injection`, `scan_open_ports`, and `flood_attack`. These additions increase the tool's capabilities in various areas like content crawling, network scanning, dictionary attacks, exploitation, and denial-of-service attack.

The new actions are also illustrated in the Venn diagram shown in Figure 5.1. This diagram provides a comprehensive overview of the action coverage achieved by the enhanced CyPROM compared to its predecessor, CyPROM v0.1 by emphasizing the common features of the two versions as well as the major improvements made in the extended CyPROM.
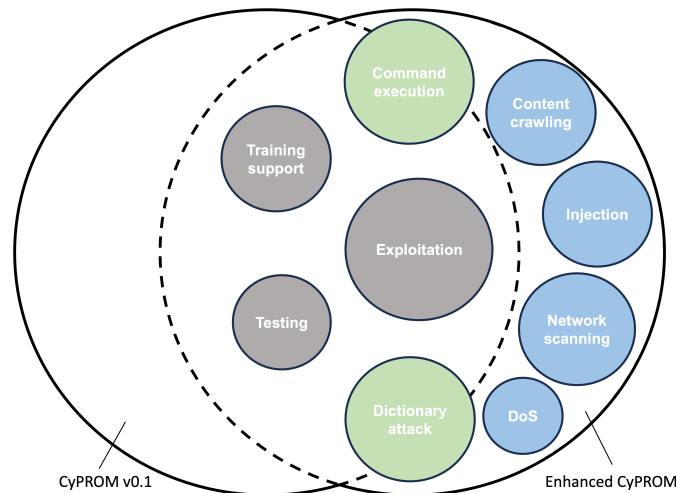

Figure 5.1: Venn diagram of action coverage.

Table 5.2: Comparison of the implemented actions in the enhanced CyPROM versus the original version.

Categories	CyPROM v0.1	Enhanced CyPROM
Training support	hint, message, question	hint, message, question
Command execution	ssh_cmd_exec	ssh_cmd_exec, **cmd_exec**
Dictionary attack	ssh_dict_attack	ssh_dict_attack, **brute_force_auth**
Exploitation	metasploit, php_auth_bypass	metasploit, php_auth_bypass
Content crawling	N/A	**crawl_content**, **crawl_content_auth**
Network scanning	N/A	**find_ip_address**, **scan_open_ports**
Denial of service	N/A	**flood_attack**
Injection	N/A	**login_sql_injection**
Testing	test	test

The first overlapping area showed the core areas that have been carried over from CyPROM v0.1 and retained in the enhanced version, including the "Training Support", "Testing", and "Exploitation" functions that formed the backbone of the original system. These fundamental functions had been carefully retained so that trainees could benefit from the same solid foundation for the training. The second overlapping segment highlighted components originally included in CyPROM v0.1 that have been expanded and refined in the enhanced version. This segment included two categories: "Command execution" and "Dictionary attack" which enabled the enhanced CyPROM to effectively address vulnerabilities in web applications and provide trainees with capabilities to identify and remediate potential security vulnerabilities. The most notable enhancement to the enhanced CyPROM was in the third and exclusive segment, which included newly added features that were not included in the original release. These include "Content crawling", "Injection", "Network scanning", and "Denial of Service (DoS)" capabilities. These additions were carefully designed to address specific web application vulnerabilities found in the Yii2 framework and listed in the OWASP Top Ten. By integrating the findings from the OWASP Top Ten security risks and performing a thorough analysis of Yii2, the enhanced CyPROM provides security professionals and developers with advanced capabilities to effectively address modern vulnerabilities in web applications.

In summary, this section demonstrates that the enhanced CyPROM version achieved a substantial increase in coverage compared to the original version. By incorporating additional actions and scenarios, CyPROM now effectively identifies a broader range of web application vulnerabilities, including those categorized under the OWASP Top Ten. The comprehensive scenario evaluating Yii2 using all actions ensures a holistic assessment of the framework's security.

5.2.2 Overall Scope Assessment

The overall scope of a training system is a crucial indicator of its ability to be used for training regarding how to detect, prevent, and respond to potential security threats effectively. This evaluation focuses on how the enhanced version of CyPROM has significantly broadened its scope, leading to a substantial improvement in its security posture compared to its original counterpart. By analyzing the addition of new actions and the expansion of scenarios, there will be a demonstration of how these improvements collectively reinforce CyPROM's capacity to detect and neutralize potential threats.

Table 5.3: Comparison of the overall scope of the enhanced CyPROM versus the original version.

Aspects	CyPROM v0.1	Enhanced CyPROM
Quantity	❖ Limited to 9 actions and 4 scenarios	❖ Expanded to 16 actions and 11 scenarios
Diversity	❖ Limited scope implies that the system's capabilities are restricted to a smaller set of potential actions and security scenarios	❖ A broader scope enhances the system's adaptability, enabling it to tackle a wider variety of cybersecurity situations
Real-world Threat Simulation	❖ Limited to some specific actions and scenarios	❖ The latest real-world cyber threats outlined by the OWASP Top Ten

As highlighted in Table 5.3, the enhanced version of CyPROM introduces new security actions and scenarios that significantly bolster its overall scope. These measures are strategically designed to swiftly detect and neutralize potential threats, thus greatly reducing the likelihood of successful cyber-attacks. The broader scope allows CyPROM to encompass a wider range of security challenges, providing a more robust and comprehensive defense against evolving cyber threats.

5.3 User Evaluation

This section aims to assess the effectiveness of the hands-on training program for mitigating web application vulnerabilities in Yii2, based on the analysis of the OWASP Top Ten vulnerabilities. The evaluation gathered feedback from participants with different backgrounds and experiences to identify strengths and areas for improvement in the training.

The evaluation was conducted using a survey distributed to 22 participants who attended a short hands-on training program. The survey included questions related to participants' current occupation, web application development experience, Yii2 development experience, familiarity with OWASP Top Ten, perception of the training objectives' clarity, and satisfaction with the enhancements made to CyPROM; the detailed content of the survey is provided in Appendix A.

5.3.1 Participant Analysis

In Figure 5.2 we show the classification of the participants according to several criteria. The majority of participants consisted of software and web developers, constituting 63.7% of the respondents, and students, comprising 27.3% of the cohort (see Figure 5.2a). Interestingly, a significant proportion of participants, approximately 45.5%, had only 0-1 year of web application development experience (see Figure 5.2b). Furthermore, among the respondents, 63.6% have experience in Yii2 development (see Figure 5.2c). Out of this group, half of them are relatively new to Yii2 development, while the remaining are at a higher level of expertise, indicating that the program is also appealing to experienced developers seeking to enhance their skills.

In addition, a remarkable 77.3% of the participants displayed familiarity with OWASP Top Ten vulnerabilities (see Figure 5.2d), indicating a positive trend in their awareness of critical security concerns. This understanding of OWASP Top Ten serves as a foundation for building upon their existing knowledge during the hands-on training program. As a result, it is evident that tailoring the training objectives to align with participants' backgrounds and experience levels can greatly impact the program's success.

Upon analyzing the survey results, valuable insights were gleaned. Many respondents found the training based on CyPROM and Yii2 useful and effective for improving their knowledge about web vulnerabilities (81.82%) and vulnerability mitigation (86.36%).
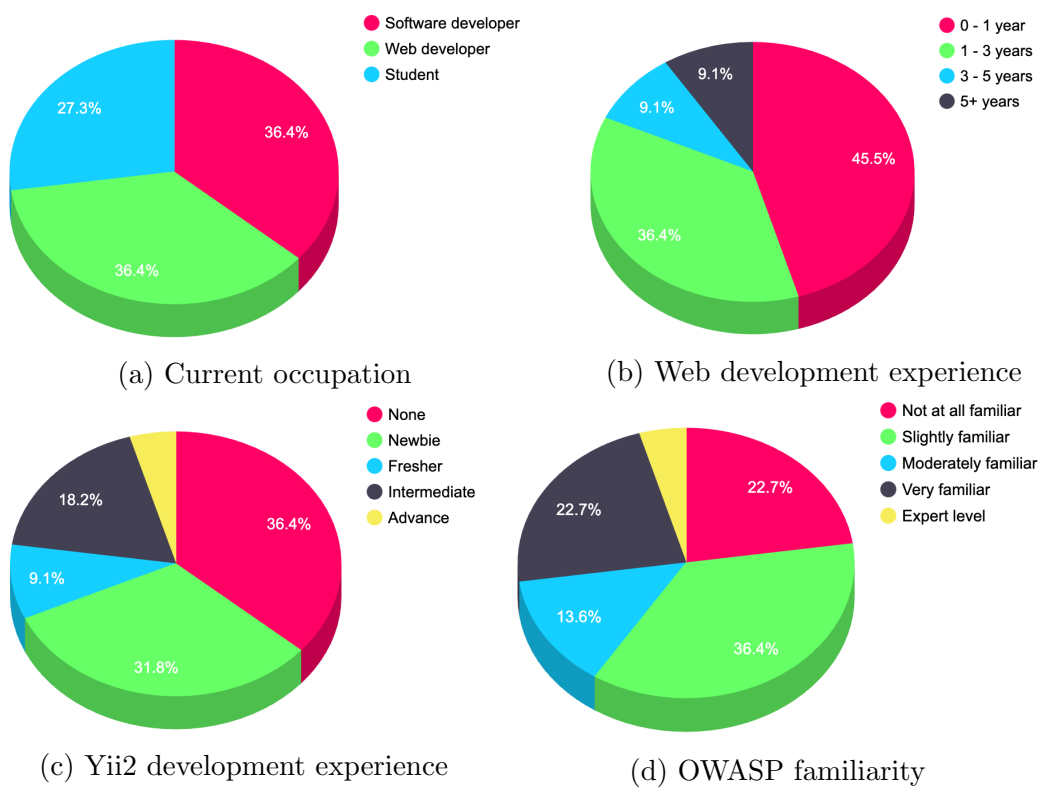
(a) Current occupation

(b) Web development experience

(c) Yii2 development experience

(d) OWASP familiarity

Figure 5.2: Participant classification according to several criteria.

5.3.2 OWASP Top Ten Category Coverage

In terms of vulnerability coverage, the respondents evaluated the CyPROM functionality on a scale from 1 to 5 for each OWASP Top Ten category, where higher scores indicate better coverage and mitigation. The result analysis, presented in Figure 5.3, highlights the CyPROM strengths and areas for improvement. The bar chart displays the results of the voting process for each OWASP Top Ten category, with the average scores allocated to each category.
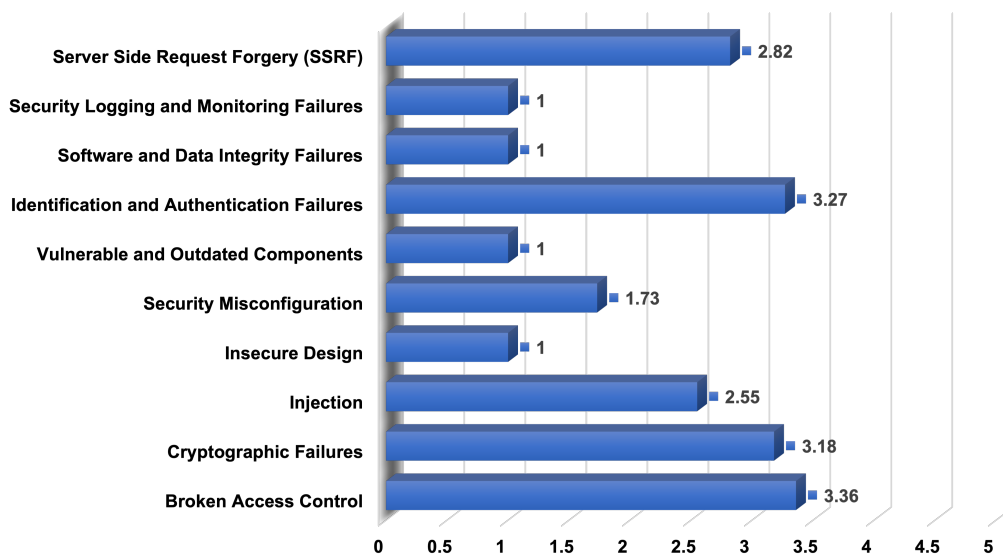

Figure 5.3: Average results for the respondent evaluation of the OWASP Top Ten category coverage (1 = poor, 5 = excellent).

The enhanced CyPROM demonstrated a relatively high coverage of these categories, as follows:

❖ **Broken Access Control:** 3.36 out of 5 - This category received the highest average score, showcasing CyPROM's powerful capabilities in dealing with access control vulnerabilities.

❖ **Identification and Authentication Failures:** 3.27 out of 5 - CyPROM displayed its effectiveness in addressing identification and authentication-related vulnerabilities.

❖ **Cryptographic Failures:** 3.18 out of 5 - Indicates CyPROM's proficiency in handling cryptographic vulnerabilities.

It is worth noting that while CyPROM performed well in the aforementioned categories, it received a low rating in other areas. The low score in these categories suggested that further improvements are required to enhance CyPROM's overall effectiveness. Note that the score of 1 (poor coverage) was expected for the four OWASP Top Ten categories that we did not address in the current implementation.

In summary, the assessment revealed strengths and areas for improvement. The high level of coverage in critical categories underscores the value of the program in addressing specific vulnerabilities in web applications. To improve CyPROM's comprehensive vulnerability remediation capabilities, efforts should be directed at increasing coverage in areas where CyPROM currently performs worse. If these deficiencies are addressed, CyPROM can establish itself as a more robust and reliable tool for addressing vulnerabilities in web applications in several OWASP Top Ten categories.

Chapter 6

Conclusion

6.1 Summary

In this research, a comprehensive journey was undertaken to improve web application security through hands-on exercises. The foundation was laid with an overview of CyPROM, the Yii2 PHP Framework, and the OWASP Top Ten, effectively combining the strengths of these tools and frameworks. The result was a well-thought-out methodology for evaluating and improving the security posture of Yii2 applications.

During the implementation phase, a hands-on training program was developed and delivered, specifically targeting the web application vulnerabilities identified in the OWASP Top Ten list. The vulnerability remediation strategy included several key steps. First, training objectives were established to ensure a targeted approach. Then, realistic actions and scenarios were created to give developers hands-on experience. The hands-on training equipped developers with the skills they needed to effectively address potential security issues. In particular, the importance of measuring the success of the training program to accurately measure its effectiveness was discussed.

The evaluation process included a comprehensive assessment of the functionality of the hands-on training program. This included conducting a comparative analysis to understand the impact on the overall security posture of Yii2 applications. Crucial insights were gained through user evaluations, which provided valuable perspectives on the practicality and usability of the training program from the developers' perspective.

In order to make our work accessible to the broader community and invite collaboration for continued advancement in the field of web application security, the enhanced CyPROM will be published on GitHub[1]. The source

---

[1]https://github.com/crond-jaist/cyprom

code was released under a permissive open-source license, allowing anyone to build upon and modify the code. We also included thorough documentation and instructions, making it easier for everyone to get started with CyPROM.

In summary, the research focused on improving web application security through a well-structured and practical training program. By coherently linking the concepts, from the initial foundation to the evaluation of the program's effectiveness, this research aimed to improve security practices in Yii2 applications.

6.2 Future Work

This thesis represents a notable advancement in addressing web application vulnerabilities through practical training. However, there are still numerous potential avenues for future research to expand upon and further improve our contributions. By capitalizing on these opportunities, we can continue to strengthen the effectiveness of our approach and make even greater strides in enhancing web application security.

Enhanced Training Modules While the current hands-on training approach focuses on mitigating vulnerabilities outlined in the OWASP Top Ten and NVD, there is potential for enriching the training modules to cover a broader range of web application security topics. To achieve this, additional vulnerabilities and attack scenarios beyond the OWASP Top Ten can be included. This expansion could include emerging threats or specific vulnerabilities relevant to any other frameworks.

Training Gamification To enhance engagement and learning retention, transforming the training approach into a gamified learning experience could be explored. Gamification elements, such as challenges, rewards, and leaderboards, can motivate learners to actively participate in the training and promote healthy competition among participants.

Acknowledgement

I would like to express my heartfelt gratitude to Associate Professor Razvan Beuran for his invaluable guidance and support throughout the journey of this thesis. His expertise and encouragement have been instrumental in shaping the direction of my research and enhancing the quality of my work. Additionally, parts of this thesis were written with the assistance of Chat-GPT[2], Grammarly[3], QuillBot[4], and Bard AI[5].

---

[2]https://chat.openai.com/
[3]https://app.grammarly.com/
[4]https://quillbot.com/co-writer
[5]https://bard.google.com/

Bibliography

[1] The DBIR Team. "2022 DBIR Data Breach Investigations Report".
    Available: https://www.verizon.com/business/en-gb/resources/2022-
    data-breach-investigations-report-dbir.pdf

[2] R. Beuran, T. Inoue, Y. Tan, Y. Shinoda. "Realistic Cybersecurity
    Training via Scenario Progression Management". IEEE European Sym-
    posium on Security and Privacy Workshops (EuroSPW 2019), Workshop
    on Cyber Range Applications and Technologies (CACOE'19), Stock-
    holm, Sweden, June 20, 2019

[3] "Yii PHP Framework". Available: https://www.yiiframework.com/

[4] OWASP. "OWASP Top 10 Vulnerabilities". owasp.org.
    https://owasp.org/www-project-top-ten/ (Accessed Jun. 1, 2023).

[5] Cyber Range Organization and Design (CROND) NEC-endowed chair
    at the Japan Advanced Institute of Science and Technology (JAIST).
    (2019). "CyPROM User Guide". Available: https://github.com/crond-
    jaist/CyPROM/releases

[6] enPiT University Consortium. enPiT-Security (SecCap) Training Pro-
    gram (in Japanese). Available: https://www.seccap.jp/

[7] Japan Network Security Association (JNSA). Security Contest (SEC-
    CON) (in Japanese). Available: http://seccon.jp/

[8] CYDER. Ministry of internal affairs and Communications. "Cyber
    defense exercise with recurrence (CYDER)" [in Japanese]. Available:
    https://cyder.nict.go.jp/

[9] Web Application Security Forum, "Hardening Project", (in Japanese),
    [Online]. Available: https://wasforum.jp/hardening-project/

[10] OWASP. "OWASP WebGoat". owasp.org. Available:
    https://owasp.org/www-project-webgoat/

[11] "DAMN vulnerability web application". Available: https://github.com/digininja/DVWA

[12] "DVWA (Damn Vulnerable Web Application)" (in Japanese). Available: https://security-blog-it.com/16317/

[13] "Hack This Site". Available: https://www.hackthissite.org/

[14] "About CTFs". Available: https://ctftime.org/about/

[15] Orange Cyberdefense. "Serial hackers: War Games". Available: https://www.orangecyberdefense.com/be/blog/serial-hackers-war-games

[16] "Cyber Security Certifications". Available: https://elearnsecurity.com/

[17] "INE Courses (INE Individual Plans)". Available: https://ine.com/learning/courses

[18] "Business Solutions". Available: https://ine.com/business-solutions

[19] OverTheWire. "Wargames". Available: https://overthewire.org/wargames/

[20] Bjoern Kimminich. (2021). OWASP Juice Shop. OWASP Juice Shop Project. Available: https://owasp.org/www-project-juice-shop/

[21] OWASP. "OWASP Application Security Verification Standard". Available: https://owasp.org/www-project-application-security-verification-standard/

[22] OWASP. "OWASP Automated Threats to Web Applications". Available: https://owasp.org/www-project-automated-threats-to-web-applications/

[23] OWASP. "OWASP API Security Project". Available: https://owasp.org/www-project-api-security/

[24] MITRE. "Common Weakness Enumeration". Available: https://cwe.mitre.org/

[25] "Metasploit — Penetration Testing Software, Pen Testing Security". Available: https://www.metasploit.com/

[26] Hastie, T., Tibshirani, R., & Friedman, J. (2009). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Stanford, CA: Stanford University". Available: https://web.stanford.edu/ hastie/Papers/ESLII.pdf

[27] Pliego Marugán, A., García Márquez, F.P. and Lorente, J., 2015. "Decision making process via binary decision diagram. International Journal of Management Science and Engineering Management", 10(1), pp.3-8.

[28] Emmert-Streib, F., Moutari, S. and Dehmer, M., 2019. "A comprehensive survey of error measures for evaluating binary decision making in data science". Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(5), p.e1303.

[29] Qin, L., Yu, J.X. and Chang, L., 2009, June. "Keyword search in databases: the power of RDBMS. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data" (pp. 681-694).

[30] "Top 10 PHP Frameworks To Use in 2023". Available: https://www.interviewbit.com/blog/php-frameworks/

[31] "Yii PHP Framework: The Definitive Guide to Yii 2.0". Available: https://www.yiiframework.com/doc/guide/2.0/en/

[32] Gomes, E., Guerra, E., Lima, P. and Meirelles, P., 2023. "An Approach Based on Metadata to Implement Convention over Configuration Decoupled from Framework Logic". Available: https://www.authorea.com/doi/full/10.22541/au.168067455.55373151

[33] "Krajee Yii Extensions — Kartik". Available: https://demos.krajee.com/

[34] "Yii Project Template Comparison". Available: https://www.yiiframework.com/extension/yiisoft/yii2-app-advanced/doc/guide/2.0/en/start-comparison

[35] "Insecure Design". Available: https://crashtest-security.com/insecure-design-vulnerability/

[36] "Security Misconfiguration: Types, Examples & Prevention Tips". Available: https://www.aquasec.com/cloud-native-academy/supply-chain-security/security-misconfigurations/

[37] "OWASP Top 10 – A06 Vulnerable And Outdated Components Explained". Available: https://www.vumetric.com/blog/owasp-top-10-a06-vulnerable-and-outdated-components-explained/

[38] "Prevention guide for the identification and authentication failure". Available: https://info.veracode.com/rs/790-ZKW-291/images/identification-authentication-failure-prevention-guide-en.pdf

[39] "OWASP Top 10 – A08 Software And Data Integrity Failures Explained". Available: https://www.vumetric.com/blog/owasp-top-10-a08-software-and-data-integrity-failures-explained/

[40] "OWASP Top Ten: #9 Security Logging and Monitoring Failures". Available: https://foresite.com/blog/owasp-top-ten-9-security-logging-and-monitoring-failures/

[41] Bright. "SSRF Attack: Impact, Types, and Attack Example". Available: https://brightsec.com/blog/ssrf-attack/

[42] NIST Computer Security Division, Information Technology Laboratory. "National vulnerability database". Available: https://nvd.nist.gov

[43] O. Wannenwetsch and T. A. Majchrzak, "On constructing persistent identifiers with persistent resolution targets", 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), Gdansk, Poland, 2016, pp. 1031-1040.

[44] "Yii2 App Advance: Issues". Available: https://github.com/yiisoft/yii2-app-advanced/issues

[45] Grad Coach. "Quantitative Data Analysis Methods & Techniques 101 - Grad Coach". Available: https://gradcoach.com/quantitative-data-analysis-methods/

[46] Indusface. "OWASP Top 10 Vulnerabilities in 2021: How to Mitigate Them?". Available: https://www.indusface.com/blog/owasp-top-10-vulnerabilities-in-2021-how-to-mitigate-them/

Appendices

Appendix A

Survey Questionnaire: Hands-on Training for Mitigating Web Application Vulnerabilities

Hands-on Training for Mitigating Web Application Vulnerabilities

This evaluation aims to gather insights on your experience with CyPROM, its hands-on training, and how effectively it helped you identify and mitigate web application vulnerabilities. Your feedback will play a crucial role in enhancing the overall performance and usability of CyPROM, making it a more robust tool for the community.

* Indicates required question

Personal Information
Please provide all neccessary information about your self. This will be used as a factor in the evaluation.

What is your name? *

Your answer

What is your email? *

Your answer

What is your current occupation or field of study? *

○ Web developer

○ Software developer

○ Lecturer or Instructor

○ Student

○ Other: _____

Your professional background: *

☐ Software Development

☐ Cybersecurity

☐ Web Application Development

☐ Other: _____

Years of experience in web application development/cybersecurity? *

○ 0 - 1 year

○ 1 - 3 years

○ 3 - 5 years

○ 5+ years

Your Yii2 experience level is: *

○ None

○ Newbie

○ Fresher

○ Intermediate

○ Advance

CyPROM and Security Knowledge
This section focuses on the basic knowledge about CyPROM, OWASP Top Ten and related issues

Have you heard about CyPROM before participating in this evaluation?

○ Yes

○ No

○ Maybe

How familiar are you with the OWASP Top Ten vulnerabilities?

○ Not at all familiar

○ Slightly familiar

○ Moderately familiar

○ Very familiar

○ Expert level

Have you used any other security training platforms or frameworks in the past?

○ Yes

○ No

○ Maybe

Hands-on Training with CyPROM and Yii2

Rate your overall experience with CyPROM

	1	2	3	4	5	
Very poor	○	○	○	○	○	Excellent

Were the training objectives clearly defined and relevant?

○ Strongly disagree

○ Disagree

○ Neutral

○ Agree

○ Strongly agree

Did the hands-on training using CyPROM and Yii2 help you understand web application vulnerabilities better?

○ Yes

○ No

○ Other:

Did you find the hands-on training approach effective in learning about web application vulnerabilities and their mitigation?

◯ Yes

◯ No

◯ Maybe

How would you rate the comprehensiveness of the CyPROM platform in covering web application security topics?

	1	2	3	4	5	
Not comprehensive at all	◯	◯	◯	◯	◯	Extremely comprehensive

Which specific web application vulnerabilities or security concepts did you find most valuable to learn about during the training?

Your answer

Do you believe that the hands-on training has improved your ability to identify and mitigate web application vulnerabilities in real-world scenarios?

◯ Yes

◯ No

◯ Maybe

Overall Assessment

Did the training effectively address the OWASP Top Ten vulnerabilities? Please rate the coverage of each vulnerability:

Broken Access Control

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Cryptographic Failures

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Injection

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Insecure Design

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

Security Misconfiguration

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

Vulnerable and Outdated Components

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

Identification and Authentication Failures

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

Software and Data Integrity Failures

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Security Logging and Monitoring Failures

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Server-Side Request Forgery

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Overall, how satisfied are you with the enhancements made to CyPROM using Yii2 and OWASP Top Ten analysis?

	1	2	3	4	5	
Very Dissatisfied	○	○	○	○	○	Very Satisfied

Considering your experience with CyPROM and the hands-on training, how likely are you to recommend CyPROM to others in the industry?

	1	2	3	4	5	
Not likely at all	○	○	○	○	○	Likely

Please provide any additional comments or suggestions for improving CyPROM or the hands-on training experience.

Your answer

Thank you for participating in this survey! Your feedback is invaluable to our research.