Development of Vulnerable Web Application Based on OWASP API Security Risks

Muhammad Idris
Dept. of Information Technology
Politeknik Negeri Batam
idris@polibatam.ac.id

Iwan Syarif
Dept. of Information and Computer
Politeknik Elektronika Negeri Surabaya
iwanarif@pens.ac.id

Idris Winarno
Dept. of Information and Computer
Politeknik Elektronika Negeri Surabaya
idris@pens.ac.id

2021 International Electronics Symposium (IES) | 978-1-6654-4346-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IES53407.2021.9593934

***Abstract*—APIs are critical for digital transformation as well as the establishment and development of new business models. They are the foundation of application economics which allows for quicker, better, and less expensive development. In security perspective, OWASP released its first API security report in 2019 which finally differentiate the security risk categories between API and web application. In recent years, there have been many incidents of cyber attacks related to API, while the implementation of the API itself is growing in popularity among organizations. Therefore, the need to understand APIs from a security perspective should be taken seriously and considered as an integral part of the software development life cycle. In this research, we proposed an API security learning environment called Vulnerable Academic Information System (VAIS) based on the OWASP API Security Risks with containerization deployment plan and gamification technique to provide a fun yet challenging environment for people in understanding the API security in a legal environment.**

***Keywords*—API, API security, Vulnerable Web Applications, Vulnerability Assesment, Penetration Testing, Gamification**

I. Introduction

The implementations of API in web development in the last few years appears to be beneficial to application development and innovations. By using API, people can share their own data or services to other applications. APIs are rapidly proliferating as a key element to foster reusability, integration, and innovation, enabling new consumption models such as mobile or smart TV applications [1]. However, at the same time, a whole new attack surface of web vulnerabilities is also available for the attackers to exploit and gain access to the system.

APIs companies currently have 73% of enterprises use more than 50 APIs which is tough to manage, especially when 4 out of 5 publish APIs are used for external consumption by partners and clients [2] . Several incidents in the last few years are reported to be related to the web API security. Vulnerabilities (54%) and authentication issues (46%) topped the list, followed by bot/scraping (20%) and denial of service attacks (19%). These vulnerabilities remain until an attacker discovers and exploits them, which can result in data exfiltration, account misuse, or service downtime [3]. Furthermore, the top concern of this security problem is the targets are not only modern web applications but also platforms such as Internet of Things (IoT) devices and Mobile Application that use API server as either main or support data provider for their system. It means, the hackers are technically can exploit the platform without exploiting the platform itself and bypass it to the API server as illustrated in figure 1.

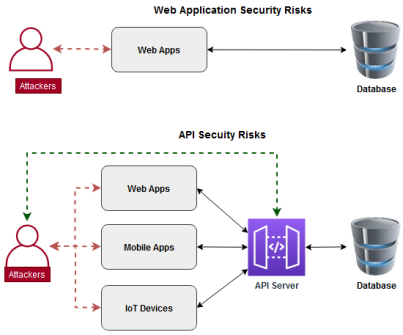

Fig. 1. API vs Web Security Risks

Application security learning can be performed through an experimental practice such as ethical hacking or penetration testing. However, because it technically requires the application as the target, this becomes the biggest problem considering that it is illegal to test security on real-world applications without the consent of the target party. Not to mention that the level of vulnerability on every application has cases with different levels of security so that the potential vulnerabilities that can be obtained from certain application do not guarantee to cover all the security risks commonly found in the real world. Currently, some existing vulnerable applications can be used as learning media or target experiments. Damn Vulnerable Web Application (DVWA) and WebGoat are two of the most popular vulnerable web applications. DVWA is a PHP/MySQL web application that is damn vulnerable which has main goals to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment [4]. WebGoat is a deliberately insecure application that allows interested developers to test vulnerabilities commonly found in Java-based applications that use common and popular open source components [5]. However, both of these popular applications

address security risks or vulnerabilities based on top ten OWASP Web Application Security Risks [6] whereas API has different perspectives in terms of security risks which are reported in the top ten OWASP API Security Risks [7].

Therefore, in this research, we designed a vulnerable web application based on OWASP API security risks to help students, teachers, web developers, and security penetration testers to recognize, learn and practice API security in a safe and legal environment. Furthermore, a study for determining the applicability of gamification to the proposed application is analyzed to create a fun yet challenging learning environment. And finally, we proposed a container-based service that can be scaled horizontally to ensure optimal performance and availability of the system, particularly for a vulnerable web application that will be tested through several web application attacks which consume high resources of the system.

II. Related Works

In this section, we try to evaluate and give some insight on how vulnerable web application can be involved and what they can bring to solve the problems in the web security field. In security learning research, S. Shin et al [8] propose a cybersecurity exercise system in a virtual computer environment called CyExec by using virtual machine and container technology. In the proposed model, WebGoat is implemented as the basic exercise of the environment. P. Chen, M. Zhao, J. Wang and H. Yu [9] proposed a teaching assistant which introduce a multi-purpose defense attack model to organize an experimental training. The model used the DVWA as the experiment project that are focused on web attacks on SQL Injection, XSS, file upload and CSRF. J. Su, M. Cheng, X. Wang and S. Tseng [10] proposed a system called SimTI-WS scheme, an online simulated test items for assessing the learning outcome of students in web security subject by conducting an experiment on Cross Site Request Forgery (CSRF) of WebGoat. N. A. Aziz, S. N. Z. Shamsuddin and N. A. Hassan [11] implemented a secure coding practice for undergraduate students. The goal of the research is to teach student on how hackers can take advantage on vulnerabilities that exist on the web applications and allow students to experience within the WebGoat environment.

In web security vulnerability assessment research, vulnerability scanning and penetration testing are the two common methodologies that are approached by researcher to explore, demonstrate, and analyze the web vulnerabilities. These both methodologies typically involve some automated tools and vulnerable web application as an object experiment. A. K. Priyanka and S. Sai Smruthi [12] demonstrate and analyze SQL and XSS injection attack with both manual and automated penetration testing using Havij, SQLMap and Xenotix to DVWA. With DVWA as target testing, the researchers are able to demonstrate and provide some countermeasures on each vulnerabilities. P. Xiong and L. Peyton [13] proposed a framework which provides a repeatable, systematic and cost-efficient approach that are fully integrated into a Security-Oriented Software Development Life Cycle by testing WebGoat and under development hospital web application to the framework. S. Tyagi and K. Kumar [14] evaluate two source code analysis tools against two vulnerable web applications which are bWAPP and DVWA. In the experiment, the researchers used OWASP WAP and RIPS to analyze the vulnerable source code and successfully find some vulnerabilities on the applications and give their insight on the performance of each tools.

Moreover, in order to improve the existing vulnerable application as an educational system, we investigate the recent studies on effectiveness of gamification implementation in the security field. D. Alami and F. Dalpiaz [15] proposed the use of an interactive gamified tutorial that is embedded within a security requirements modeling tool called TS-Tooltorial. Based on the experiment, the results suggest that the innovative teaching method may be a competitor for both traditional and modern approaches. S. A. Kumar, N. R. Kumar, S. Prakash and K. Sangeetha [16] proposed a simple mini game based CAPTCHA called next generation CAPTCHA using gamification technique. The goal of the gamification implementation in CAPTCHA is to differentiate human being from the automated bots. The study was conducted to test the performance of the game based CAPTCHA in terms of understanding, the level of complexity, time taken, and user experience to 30 participants. The result shows that next generation CAPTCHA is 20% more understandable than the traditional technique such as text based, image or audio based CAPTCHA. Moreover, the time taken by the user in solving the game based CAPTCHA is relatively faster compared to the old techniques. S. Ros et al [17] proposed a cybersecurity game called “Name Suppressed” based on the Cognitive Constructivism learning theories. The goal of this game is to analyzes students’ self-perception of success and learning effectiveness after using non-compulsory gamification in an online cybersecurity course. The results suggest a high correlation between playing the game and succeeding in the course. Therefore, by introducing gamification in the educational curricula improves student engagement and consolidates their knowledge on cybersecurity new teaching processes and in consequence improve the academic performance and minimize dropout rates.

III. Proposed Model

A. System Architecture

Based on the diagram shown in Figure 2, this research contains a web security learning system application that is built based on 3 main components:

- Web architecture based on microservices with REST API data distribution. In the context of vulnerable web application, microservice architecture have some advantages:
  - Variety of programming languages and database technologies implementation will provide different scope of security risks.

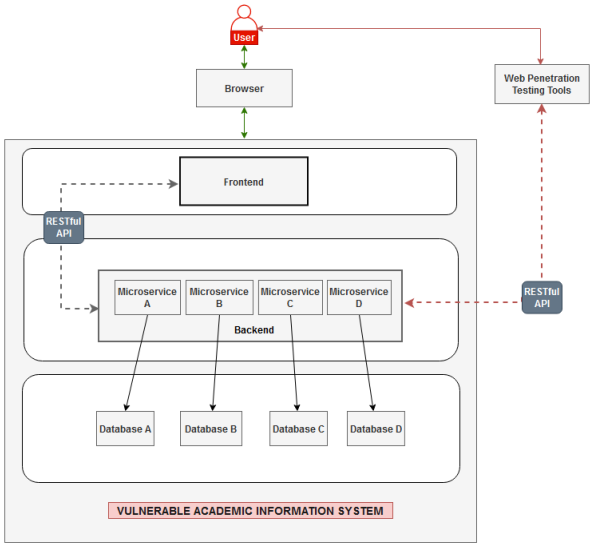

Fig. 2. Proposed System Architecture

- New features addition and code maintenance without the risk of breaking the entire application. Especially in term of security testing that have several test scenario which likely to break the vulnerable application.
- Scalability of specific service which likely has a lot of traffics caused by automate penetration testing tools from multiple users.
- Frontend service
  This service will be the face of the environment. The frontend should be implemented using javascript framework to provide an interactive and stylish learning environment such as VueJS, React, Svelte, Angular, etc.
- Backend service
  This service will act as a web service that will run in a web server. The main purpose of this service is to provide the data from database to the frontend via RESTful API calls. By using microservices architecture, we are able separate the backend to several microservices which able to implement different kinds of web programming languages such as slim php, express JS, lumen, spring, flask, etc and the web server technology which hold all of the web assets such as nginx, apache, Lighttpd, Node, etc.
- Database service
  This service will provide the data to the micro services using either SQL or NoSQL technology such as mariaDB, mySQL, PostgreSQL, MongoDB, Cassandra, etc.

- Container based application for easy configuration and deployment.
- Security learning environment with gamification techniques through a vulnerable application called Vulnerable Academic Information System (VAIS). VAIS emulates the content of academic information system applications commonly used in academia. The purpose of giving context to the proposed vulnerable application is to build a system that can relate closely to the user when conducting security testing to create a relevant test environment such as problems commonly found in real-world applications.

B. Use Case and Misuse Case

To map what features are offered by VAIS and the potential attack vectors against each of these features. This section describes the application features and the attack vectors through use case and misuse case models as shown in Figure 3.

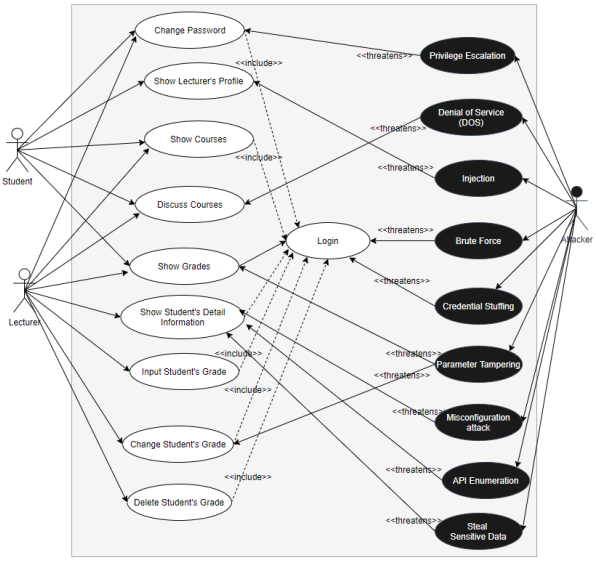

Fig. 3. VAIS Use and Misuse Case Model

C. Security Risks Scope

Based on [8] [9] [10] [11] [12] [13] [14], There are some existing vulnerable applications with several web security risks to help people in learning web application security in safe and legal environment. However, these applications are designed to have security risks based on the OWASP Top 10 Web Application whereas the proposed vulnerable application is focused on API Security that is designed based on the OWASP Top 10 API Security Risks 2019 which are:

- API1:2019 Broken Object Level Authorization
  The attacker can replace the object ID of their own resource with the ID of a resource belonging to another legitimate user in the API call. The lack of proper authorization checks allows the attacker to access the specified resources.

- API2:2019 Broken User Authentication
  Authentication mechanisms are often misimplemented, allowing attackers to compromise an authentication

token or exploit a flaw to guess other user's identities.

- API3:2019 Excessive Data Exposure
  Some APIs may be implemented before the exact specification becomes available. This leads to common implementation where the API exposes all object properties, exposes much more data than the API client legitimately needs and relies on the client to do some filtering. This may expose sensitive data such as personally identifiable information (PII).

- API4:2019 Lack of Resources Rate Limiting
  In most cases, the API imposes no limits on the size and number of resources a client can request. This not only leads to denial of service (DoS) affecting the performance of the API server, but also leaves the door open to authentication flaws such as brute force attack.

- API5:2019 Broken Function Level Authorization
  Unclear divisions of hierarchies, groups, and roles between the access control policies of administrative and general functions tend to lead to authorization flaws.

- API6:2019 Mass Assignment
  The API retrieves the data provided by the client and stores it without proper filtering of whitelisted properties.

- API7:2019 Security Misconfiguration
  Insufficient API server configuration allows attackers to exploit them.

- API8:2019 Injection
  An attacker can inject malicious data to an API endpoint, expecting it to blindly executed as code in the server.

- API9:2019 Improper Assets Management
  APIs tend to expose more endpoints than traditional web applications. With many available endpoints as target, attackers can look for non-operational or earlier versions of the API that are not protected and maintained properly like the current production API and use them to launch an attack.

- API10:2019 Insufficient Logging & Monitoring
  Due to inadequate logging, monitoring and inefficient integration with incident response, attackers can further attack the system unnoticed.

D. Gamification Design

The implementation of gamification in the proposed vulnerable application is designed to create a fun yet challenging learning experiment to the users. By following the effectiveness of recent studies [15] [16] [17], we design a gamification to the API security learning environment which consist of game elements and mechanics. In gamification, game elements are needed as core part of user and system interaction. The are 2 main elements used in the proposed application.

- Challenges
  Provide some scenarios and objectives for users to solve. The scenarios can imitate the real world incident and a custom fictious scenario which are categorized based on OWASP Top 10 API Security Risks 2019.
- Scoreboard
  After every challenge completion, the scoring system will calculate the maximum score of the challenge and give the result to the scoreboard.

After the game elements are specified. We designed a game mechanics based on DAST(dynamic application security testing). In this game mechanic, there are two types of learning model which can be applied to the vulnerable application:

- Exploit The Endpoint
  The goal of this mode is to encourage users to learn and perform DAST or black box testing approach on API security by exploiting the vulnerabilities on the API endpoints.
- Enumerate The Endpoint
  The goal of this mode is to encourage users to learn and perform API enumeration in order to continue the challenges. The interaction and flow between this game mechanics and the game elements can be seen in fig.4.

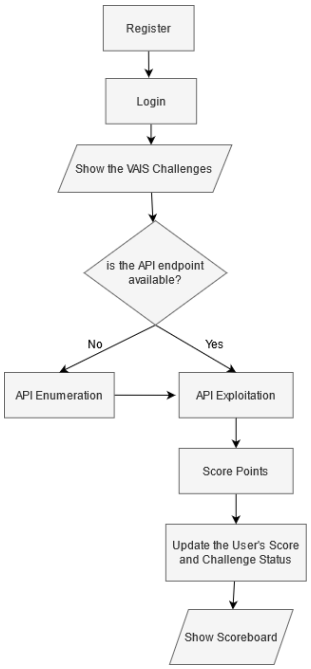

Fig. 4. Gamification based on DAST

E. Deployment Plan

The deployment plan on this research will be implemented using container technology.The purpose of this design is to ensure that VAIS can be accessible and tested simultaneously by many users at the same time with a high level of server

availability. The deployment plan design of VAIS can be seen in Fig.5.

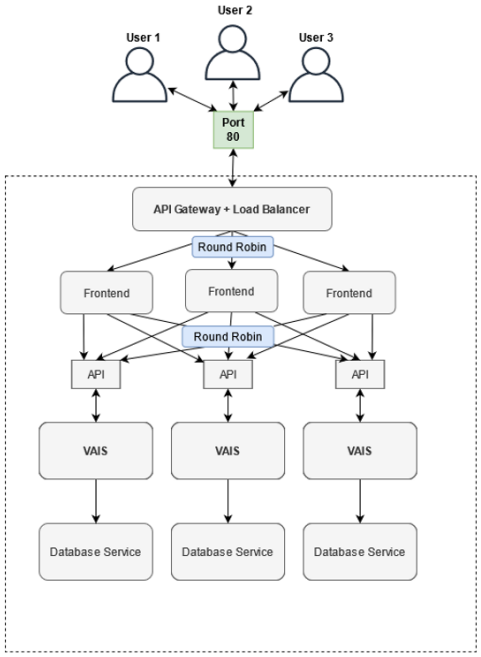

Fig. 5. VAIS Deployment Diagram

IV. Conclusions

VAIS as the vulnerable web application will bring a whole new avenue about API security for either learning or research purposes. By using microservices architecture, VAIS can present a new vulnerable modern web application that has the capability to provide a variety of vulnerabilities related to the specific programming language, database technology, and web server technology in one place of API security learning environment. Furthermore, the methodologies in web security such as penetration testing and ethical hacking to simulate the web attacks against the vulnerable application make the applicability of gamification in API security learning more straightforward to implement. In the end, this research gives an early-stage development of vulnerable web API application which has the goal to make people can learn API security risks and their attack vectors in fun yet challenging environment without worrying about the consequences of the action during the learning progress.

V. Future Works

- Implementation of VAIS by using popular microframeworks and database management system based on the proposed design
- Evaluation and Analysis of each API Security Risks in VAIS challenges
- Stress Test Analysis on containerized VAIS againts common web attack vectors
- Analysis VAIS game based existing gamification frameworks

VI. Acknowledges

We would like to thanks Politeknik Negeri Batam and Politeknik Elektronika Negeri Surabaya to make this project possible.

References

[1] S. Segura, J. A. Parejo, J. Troya and A. Ruiz-Cortés, "Metamorphic Testing of RESTful Web APIs," in IEEE Transactions on Software Engineering, vol. 44, no. 11, pp. 1083-1099, 1 Nov. 2018, doi: 10.1109/TSE.2017.2764464.

[2] "Imvision's 2021 Enterprise API Security Survey", Imvision.ai, 2021. [Online]. Available: https://www.imvision.ai/2021-api-security-survey/. [Accessed: 24- Aug- 2021]

[3] "API Security Trends", Salt.security, 2021. [Online]. Available: https://salt.security/api-security-trends. [Accessed: 04- Jun- 2021].

[4] "DVWA - Damn Vulnerable Web Application", Dvwa.co.uk, 2021. [Online]. Available: https://dvwa.co.uk/. [Accessed: 24- Aug- 2021]

[5] "OWASP WebGoat - Learn the hack - Stop the attack", Owasp.org, 2021. [Online]. Available: https://owasp.org/www-project-webgoat/. [Accessed: 24- Aug- 2021].

[6] "OWASP Top Ten Web Application Security Risks — OWASP", Owasp.org, 2017. [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed: 24- Aug- 2021].

[7] "OWASP API Security - Top 10 — OWASP", Owasp.org, 2019. [Online]. Available: https://owasp.org/www-project-api-security/. [Accessed: 1- June- 2021].

[8] J. Su, M. Cheng, X. Wang and S. Tseng, "A Scheme to Create Simulated Test Items for Facilitating the Assessment in Web Security Subject," 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media), 2019, pp. 306-309, doi: 10.1109/Ubi-Media.2019.00067.

[9] S. Shin et al., "Development of Training System and Practice Contents for Cybersecurity Education," 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), 2019, pp. 172-177, doi: 10.1109/IIAI-AAI.2019.00043.

[10] P. Chen, M. Zhao, J. Wang and H. Yu, "Exploration and Practice of the Experiment Teaching of Web Application Security Course," 2019 10th International Conference on Information Technology in Medicine and Education (ITME), 2019, pp. 381-384, doi: 10.1109/ITME.2019.00092.

[11] N. A. Aziz, S. N. Z. Shamsuddin and N. A. Hassan, "Inculcating Secure Coding for beginners, 2016 International Conference on Informatics and Computing (ICIC), 2016, pp. 164-168, doi: 10.1109/IAC.2016.7905709.

[12] A. K. Priyanka and S. Sai Smruthi, "Web Application Vulnerabilities: Exploitation and Prevention," 2020 International Conference on Electrotechnical Complexes and Systems (ICOECS), 2020, pp. 1-5, doi: 10.1109/ICOECS50468.2020.9278437.

[13] P. Xiong and L. Peyton, "A model-driven penetration test framework for Web applications," 2010 Eighth International Conference on Privacy, Security and Trust, 2010, pp. 173-180, doi: 10.1109/PST.2010.5593250.

[14] S. Tyagi and K. Kumar, "Evaluation of Static Web Vulnerability Analysis Tools," 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2018, pp. 1-6, doi: 10.1109/PDGC.2018.8745996.

[15] D. Alami and F. Dalpiaz, "A Gamified Tutorial for Learning About Security Requirements Engineering," 2017 IEEE 25th International Requirements Engineering Conference (RE), 2017, pp. 418-423, doi: 10.1109/RE.2017.67.

[16] S. A. Kumar, N. R. Kumar, S. Prakash and K. Sangeetha, "Gamification of internet security by next generation CAPTCHAs," 2017 International Conference on Computer Communication and Informatics (ICCCI), 2017, pp. 1-5, doi: 10.1109/ICCCI.2017.8117754.

[17] S. Ros, S. González, A. Robles, L. Tobarra, A. Caminero and J. Cano, "Analyzing Students' Self-Perception of Success and Learning Effectiveness Using Gamification in an Online Cybersecurity Course," in IEEE Access, vol. 8, pp. 97718-97728, 2020, doi: 10.1109/ACCESS.2020.2996361.