# A Comparison of Selected Systems For Learning About SQLi Vulnerability Suitable for Academic Uses

Djordje Madic, Danko Miladinovic and Zarko Stanisavljevic

***Abstract*— In this paper five popular platforms for secure software development training are analyzed from the perspective of their suitability for academic uses. In order to compare these platforms a novel taxonomy of interactive cyber training and education systems (Cyber Taxi) is used. Only parts of the taxonomy that are relevant are included in the analysis. The analyzed platforms are also compared to SQLiTrainer system, which was developed at the University of Belgrade, School of Electrical Engineering specifically to be used at the courses dealing with the SQL injection (SQLi) vulnerability. Based on the conducted analysis a suggestion is made regarding the requirements that a training platform should fulfill in order to be suitable for academic uses.**

***Index Terms*— SQLi; secure software development; hands-on training.**

## I. Introduction

Software engineering never had more learning resources than today, with the internet resources available at all times. It comes in many forms, as video courses, articles, e-books, and most of them are free. Most of the ones covering SQLi are structured to build theoretical knowledge while applying the knowledge and hands-on experience are missing.

On the other hand, courses covering topics like secure software development and network and system security are emerging across universities all around the world. Respectable authorities in the field of software engineering, i.e. ACM and IEEE, are issuing their recommendations regarding curriculum [1] and in these recommendations, as a rule, they state that courses covering topics from software engineering should be supported by hands-on experience for students.

There is a constant dilemma when introducing hands-on exercises at the university course, should this be done using existing technology or is there a need for a new dedicated tool to be created. In order to help teachers to resolve this dilemma regarding SQLi vulnerability as a topic to be covered at some course, in this paper analysis of five popular platforms for secure software development training is presented. These platforms are also compared to one dedicated tool for teaching SQLi vulnerability as a course topic (SQLiTrainer [2]). Based on qualitative analysis, a suggestion of requirements that a tool suitable for academic uses should fulfill is made. Finally, an example of quantitative analysis is presented. As far as the authors are aware, there are no similar studies published in open literature.

The remainder of the paper is organized as follows. The second section presents the three selected tools with hands-on SQLi exercises, one online tool with hands-on SQLi exercises, one framework for organizing secure software training, and one dedicated teaching tool. The third section compares the six selected tools using the taxonomy of interactive cyber training and education systems (Cyber Taxi) [3]. The section four describes the requirements that a tool should fulfill in order to be used in academia. The fifth section gives an example of quantitative comparison of the selected tools. The last section gives a conclusion and suggests future work.

## II. Description Of The Selected Tools

Five popular platforms are selected for this research based on their availability to the authors, their popularity in the secure software development community and their coverage of the selected topic (e.g., Juice Shop [4] is an official tool of the OWASP [5] community, Avatao [9] is used for the Serbian Cyber Security Challenge, etc.).

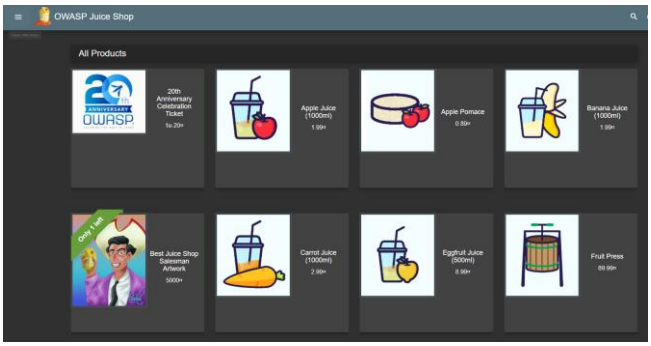


Fig. 1. OWASP Juice Shop

Juice Shop [4] (Fig. 1) is an open-source web application representing an insecure online shop. It is created by OWASP [5] organization and contains 100 challenges of varying difficulty where the user is supposed to exploit the underlying

Djordje Madic is with Zuehlke Engineering, Bul. Milutina Milankovića 1i, 11070 Novi Beograd, Serbia (e-mail: djordje.madic@zuehlke.com).

Danko Miladinovic is with University of Belgrade, School of Electrical Engineering, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (phone: +381-63-3439-97; e-mail: danko@etf.bg.ac.rs).

Zarko Stanisavljevic is with University of Belgrade, School of Electrical Engineering, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (phone: +381-11-3218-484; e-mail: zarko.stanisavljevic@etf.bg.ac.rs).

 ISBN 978-86-7466-930-3

vulnerabilities. SQL injection is covered by 7 of them. The application automatically detects when a challenge is solved, and progress of the user is tracked on a score board. All components of the user interface can be customized including color theme, logos, banners, links, products and other items in the database. Interactive help, hints and „challenge solved“ notifications can be turned off, while the initial set of challenges and their solutions cannot be changed. Domain of the application is well chosen, having in mind the number of online shops today. According to a study from 2017 [6] there are 800.000 registered online shops only in Europe. Juice Shop application code [7] has more than 50 contributors. Instance of the application can be started on a personal computer or one of the cloud services.

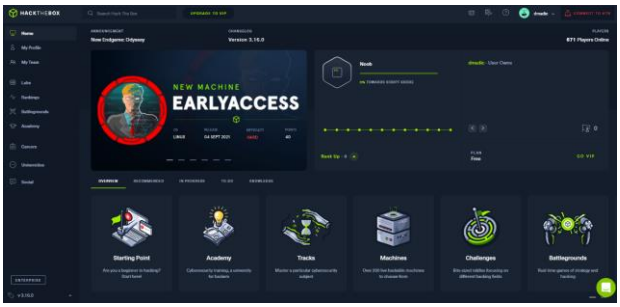


Fig. 2. Hack The Box

Hack The Box [8] (Fig. 2) is online training platform, helping individuals, companies and universities improve penetration testing skills. The platform is free for students and university professors. It includes a set of Capture The Flag (CTF) exercises grouped in 10 categories, like Web, Hardware and Reverse Engineering. Exercises are updated on weekly basis. Progress of the user is tracked and awarded with points, ranks and badges. Exercises are performed in a realistic environment, for example, against a dedicated web application instance or an executable that should be reverse engineered. Part of the exercises is followed by detailed documentation of the solutions. At the moment of writing, Web category contains 34 challenges designed as complex attacks where the user has to exploit multiple vulnerabilities to solve them. Some of them require SQLi to be solved.

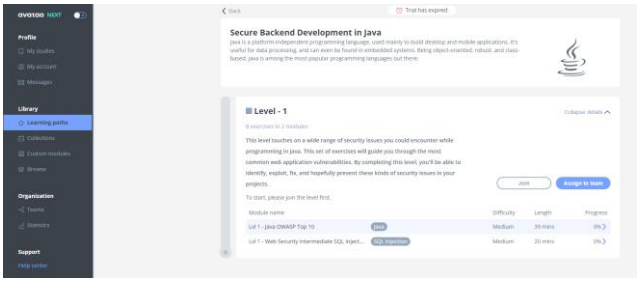


Fig. 3. Avatao

Avatao [9] (Fig. 3) is online training platform for companies created to improve security awareness of the employees and provide them with best practices for secure software development. It includes exercises in 8 programming languages, grouped in modules and learning paths. Exercises are performed in a realistic environment, for example, against a dedicated web application instance. There are 2 types of exercises, challenges and tutorials. Tutorials are interactive exercises guided by a chat bot acting as a teacher, while the challenges are designed as CTF. Most of the exercises are focused on single vulnerability, like SQLi. There are both attack and prevention oriented exercises. At the moment of writing, trial access to the platform provides 18 SQLi exercises in 5 programming languages. Many of them are different variants of SQLi Login Bypass.

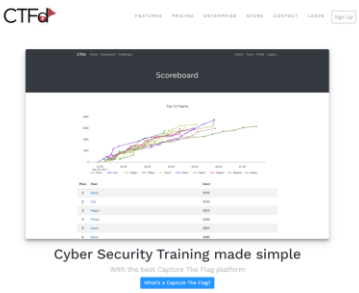


Fig. 4. CTFd Admin panel

CTFd [10] (Fig. 4) is online training framework for hosting CTF competitions and is used for helping individuals and companies improve cyber security skills using CTF challenges. Beside CTF challenges the platform supports a variety of other types of challenges such as multiple-choice and manual verification exercises. Some of these types of challenges are not available in the free version of the framework. Having said that, the framework is free, and it can be expanded by buying plugins and themes from the CTFd website. Unlike other tools, challenges do not come with the framework, but they can be added from the admin page. The admin can also add more pages, view the progress of individual users and teams on the platform and view success rate of each challenge. The framework supports individual and team competitions.


Fig. 5. PicoCTF

PicoCTF [11] (Fig. 5) is online training tool for helping users increase their hacking skills using CTF challenges. Users can register either as learners or as teachers. Learners can solve CTF challenges and join a classroom. Upon joining classrooms learners can get custom event scoreboards and track classroom member’s progress. Teachers can create

 ISBN 978-86-7466-930-3

classrooms and monitor student progress. Exercises are CTF based and are split into categories like web exploitation, cryptography, forensics, etc. There is no SQLi category, but there are around 10 SQLi challenges across all categories.

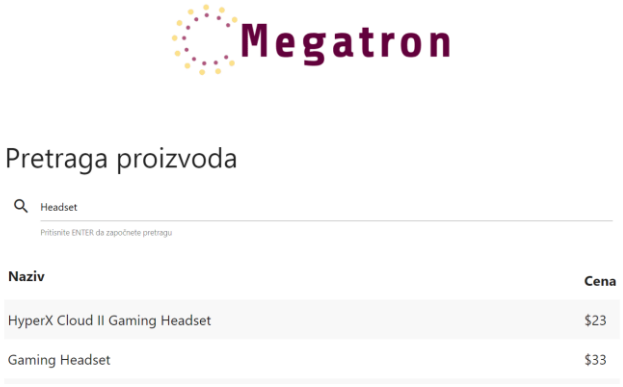


Fig. 6. SQLiTrainer

SQLiTrainer [2] (Fig. 6) represents a set of 4 vulnerable applications that can be used to demonstrate different types of SQL injection vulnerabilities. Implementation and the examples how to use the system are given in [2]. It is used for laboratory exercises at the Advanced Network and System Security course at the University of Belgrade, School of Electrical Engineering. Components of the user interface and initial state of the database can be customized. The applications can be run against different SQL databases, like MySQL and PostgreSQL, changing the solution of the exercises.

## III. Comparison Of The Selected Tools

In order to compare different tools a common ground and a set of measurable parameters are needed. For the purpose of suggesting requirements that a tool should fulfill to be used in academia a novel and promising taxonomy Cyber Taxi [3] was analyzed. Most relevant parts of the taxonomy are orchestration, proficiency level, and customization level. In [3] orchestration defines the automation level, although this term can have other meanings in a different context. Having a group of students, it is a requirement to be able to set up the training environment automatically, no matter the number of participants. In case of problems, it should be simple to restart the environment and debug it. Proficiency level defines required level of knowledge. In case it is above beginner level, the tool is not relevant for academic use. In order to provide enough learning material and assess different groups of students, training should be customizable, and should be able to produce similar exercises.

Classification of the selected tools based on the relevant parts from the Cyber Taxi is presented as a table in Fig. 7. Columns of the table represent the selected tools, while rows of the table represent components of the taxonomy, grouped by more general concepts. The table is filled based on the experience of the authors with the selected tools using their publicly available, free or trial versions. All of the tools except CTFd have trainings of similar purpose and include exercises for beginners. SQLiTrainer is the only tool requiring manual effort for scoring. Hack The Box, Avatao and picoCTF are ready-to-use products, while Juice Shop, SQLiTrainer and CTFd require the training facilitator to decide on the deployment strategy and execute it. SQLiTrainer has the most customizable exercises, while Hack The Box and Avatao can customize the set of exercises a training group will receive. CTFd can be customized based on the teacher needs.

## IV. Requirements

Based on the previous analysis and the authors previous experience with eLearning tools in computer engineering education, for a system to be suitable for academic uses, a recommendation can be made as a set of requirements that need to be fulfilled. Target audience of the system should be students with no previous knowledge of SQLi or other software vulnerabilities. This requires a system with exercises focused only on SQLi. Exercises combining SQLi with other vulnerabilities are out of scope, as the target audience should have knowledge of multiple vulnerabilities and understand how they can be exploited together. Further requirements can be defined using the Cyber Taxi, making it suitable for comparison with the existing tools.

| | | Juice Shop | Hack The Box | Avatao | CTFd | picoCTF | SQLiTrainer |
|---|---|---|---|---|---|---|---|
| Techical Setup | Environment structure | Hosting. Each user has a dedicated instance of the target application. | Commercial E-Learning and Collaboration platform | Commercial E-Learning platform | Commercial E-Learning platform | E-Learning platform | Hosting. Each user has a dedicated instance of the target application. |
| | Deployment | On-premise and Cloud | | | Cloud | | On-premise and Cloud |
| | Orchestration | Using MultiJuicer [x] exercises can be fully automated | Full automation | Full automation | Full automation | Full automation | Full automation |
| Audience | Sector | Academic and Private | Academic, Private and Public | Academic and Private | Academic, Private and Public | Academic | Academic and Private |
| | Purpose | Raise awareness and increase skill level | Raise awareness and increase skill level | Raise awareness and increase skill level | Raise awareness and increase skill level | Raise awareness and increase skill level | Raise awareness and increase skill level |
| | Proficiency level | Beginner to expert | Beginner to expert | Beginner to expert | Beginner to expert | Beginner to expert | Beginner |
| | Target audience | Students and IT Professionals | Students, IT Professionals and IT Specialists | Students and IT Professionals | Students, IT Professionals and IT Specialists | Students | Students |
| Training Environment | Training Type | Jeopardy style Capture The Flag (CTF) | Attack-only and attack-defense CTF, Cyber Training Range | CTF | CTF | CTF | CTF |
| | Scenario | Problem-driven, not supervised | Problem- and storyline-driven, not supervised | Problem- and storyline-driven, not supervised | Problem-driven, not supervised | Problem-driven, not supervised | Problem-driven, not supervised |
| Training Setup | Scoring | Awarding. Solved challenges are automatically detected. | Awarding. User is awarded with points, ranks and badges. | Awarding | Awarding. Solved challenges are automatically detected. | Awarding. Solved challenges are automatically detected. | Manual assessment |
| | Roles | No specific roles | Red and Blue teams when done in team setup | No specific roles | No specific roles | No specific roles | No specific roles |
| | Training Mode | Single | Single or Team | Single | Single or Team | Single | Single |
| | Customization Level | Specific. User interface and initial database state can be customized. Challenges cannot be customized. | Specific. Training can be customized using Dedicated Labs where administrators can pick set of exercises for training group. | Specific. Administrators can pick set of exercises for training group. | Specific. User interface and initial database and target database can be customized. | None | Specific. User interface, initial database state and target database can be customized. |

Fig. 7. Clasification of the selected tools

 ISBN 978-86-7466-930-3

**Audience**
- Target audience: Students
- Sector: Academic
- Proficiency level: Beginner
- Purpose: Raise awareness and increase skill level

**Training environment**
- Training Type: Capture The Flag
- Scenario:
    - Non supervised
    - Problem- or Storyline- driven
    - Target of the challenge is application solved by exploiting SQLi vulnerability

**Training setup**
- Scoring: Assessment, as results of the participants should be comparable
- Roles: No specific roles
- Training mode: Team, or Individual during assessments
- Customization Level: Specific, as it should be customized for each training group

**Technical setup**
- Environment structure: Online platform in the sense of E-Learning platform, or hosting
- Deployment: Both On-premise and Cloud
- Orchestration: Full degree of automation with modular design of the application

Capture The Flag is the most practical training type for students. Students can be split in groups, each group having the same flag. As flags are known upfront, they can be checked automatically once the user submits them. Depending on modularity of the application, additional customization may go along, like each group having a specific user interface, initial database state, and SQL database, like PostgreSQL or MySQL.

For the selected tools to be fully compliant with the requirements, the following features should be introduced:

- Juice Shop: Customization of the exercises, providing different flags per training group
- Hack The Box: Exercises focused only on SQLi
- Avatao: More exercises focused on SQLi or customization of the existing ones
- picoCTF: introduce a specific SQLi category

As it can be seen additional effort is needed in order to use existing platforms in academia since the two very important requirements, i.e. exercises focused only on SQLi and customization of exercises cannot be expected to be fulfilled. On the other hand dedicated tools, like SQLiTrainer and frameworks, like CTFd, will fulfill all of the requirements, but will require even more effort to be created.

## V. Example Of Quantitative Analysis

In this section we will give an example of a quantitative analysis using the results from the table represented in Fig. 7. and the requirements of the previous chapter. Each category in the previous mentioned table (technical setup, audience, training environment and training setup) will be graded on the scale from 1 to 5 based on how much the tools and the CTFd framework satisfy the requirements from the previous chapter.

Having that in mind we can draw a conclusion on how much each subcategory in each category weights in the final score of that category. Because all of the analyzed tools and CTFd framework satisfy the requirements of categories audience and training environment, on these two categories they all get the grade of 5.

That leaves us with two categories left to be graded, and those are training setup and technical setup. Here we can define how much each subcategory has weight in the final grade of its category based on the values for those subcategories from the previous chapter and from the table represented in Fig. 7. These specific weights are defined based on authors personal experiences with laboratory exercises on the Advanced Network and System Security course at the University of Belgrade, School of Electrical Engineering.

**Training setup**
- Scoring
    - Automatic assessment weights 40%
    - Manual assessment weights 20%
- Roles – 0% of the category grade
- Training mode
    - Single and team mode weights 20%
    - Only single or team mode weights 10%
- Customization Level
    - None weights 0% of the category grade
    - Ability to only pick a set of exercises for group weights 10%
    - Interface and initial database can be customized weights 20%
    - Initial database state and target database can be customized weights 30%
    - Initial database state, target database, and challenges can be customized weights 40%

**Technical setup**
- Environment structure
    - Noncommercial platform weights 50%
- Deployment
    - On premise and cloud weights 20%
    - Only on premise or on cloud weights 10%
- Orchestration
    - Full automation weights 30%

Based on these assessments we can see the final grade of each category and the average grade of each analyzed tool in a table represented in Fig. 8.

 ISBN 978-86-7466-930-3

| | Juice Shop | Hack The Box | Avatao | CTFd | picoCTF | SQLiTrainer |
|---|---|---|---|---|---|---|
| Techical Setup | 4 | 2 | 2 | 2 | 4 | 5 |
| Audience | 5 | 5 | 5 | 5 | 5 | 5 |
| Training Environment | 5 | 5 | 5 | 5 | 5 | 5 |
| Training Setup | 4 | 4 | 3 | 5 | 3 | 3 |
| Average | 4.5 | 4 | 3.75 | 4.25 | 4.25 | 4.5 |

Fig. 8. Grades of the selected tools

## VI. Conclusion

This paper presented the need for having tools to learn SQLi vulnerability in an interactive way and listed the existing tools. Selected tools were categorized based on the parts of the Cyber Taxi. The same taxonomy was used to compare the selected tools, leading to suggestion of requirements that a tool to be used in academia needs to fulfill. Conclusion is that effort needed to adapt existing tools to fulfill these requirements sometimes can exceed the effort needed to create a dedicated tool from scratch and sometimes it is even not possible to customize existing tool. However, once the requirements are clear it is always useful to check if there are existing tools that can fulfill them before starting to create a new tool. The quantitative analysis that was conducted in this paper showed how once the requirements are defined it is possible to quantify them and create a unique benchmark for each individual teacher and each individual course.

## References

[1] IEEE Computer Society and ACM, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, Available at: http://www.acm.org/binaries/content/assets/education/se2014.pdf (Accessed September 2021)
[2] Đ. Madić, Ž. Stanisavljević, SQLITRAINER - Sistem za učenje o SQLi sigurnosnim propustima u aplikacijama, ETRAN 2021, RT1.2, September 2021
[3] Knüpfer, M., Bierwirth, T., Stiemert, L., Schopp, M., Seeber, S., Pöhn, D. and Hillmann, P., 2020, September. Cyber Taxi: A Taxonomy of Interactive Cyber Training and Education Systems. In International Workshop on Model-Driven Simulation and Training Environments for Cybersecurity (pp. 3-21). Springer, Cham.
[4] Juice Shop, Available at: https://owasp.org/www-project-juice-shop/ (Accessed September 2021)
[5] OWASP, Available at: https://owasp.org/ (Accessed September 2021)
[6] E-Commerce news, Available at: https://ecommercenews.eu/800000-online-stores-europe/ (Accessed September 2021)
[7] Juice Shop code repository, Available at: https://github.com/bkimminich/juice-shop (Accessed September 2021)
[8] Hack The Box, Available at: https://www.hackthebox.eu/ (Accessed September 2021)
[9] Avatao, Available at: https://avatao.com (Accessed September 2021)
[10] CTFd, Available at: https://ctfd.io/ (Accessed March 2022)
[11] PicoCTF Available at: https://picoctf.org/ (Accessed March 2022)

 ISBN 978-86-7466-930-3