Cybersecurity Training Platforms Assessment

Vojdan Kjorveziroski[1*[0000-0003-0419-4300]], Anastas Mishev[1[0000-0001-7271-6655]], and Sonja Filiposka[1[0000-0003-0034-2855]]

[1] Faculty of Computer Science and Engineering,
Ss. Cyril and Methodius University, Rugjer Boshkovikj 16, 1000 Skopje,
North Macedonia
{vojdan.kjorveziroski,anastas.mishev,sonja.filiposka}@finki.ukim.mk

Abstract. Hands-on experience and training related to the latest cyberthreats and best practices, augmented with real-life examples and scenarios is very important for aspiring cybersecurity specialists and IT professionals in general. However, this is not always possible either because of time, financial or technological constraints. For cybersecurity exercises to be effective they must be well prepared, the necessary equipment installed, and an appropriate level of isolation configured, preventing inter-user interference, and protecting the integrity of the platform itself. In recent years there have been numerous cybersecurity training systems developed that aim to solve these problems. They can either be used as cloud or self-hosted applications. These solutions vary in their level of sophistication and ease-of-use, but they all share a single goal, to better educate the cyber community about the most common vulnerabilities and how to overcome them. The aim of this paper is to survey and analyze popular cybersecurity training systems currently available, and to offer a taxonomy which would aid in their classification and help crystalize their possibilities and limitations, thus supporting the decision-making process.

Keywords: cybersecurity exercises, cybersecurity training platforms taxonomy, training.

1 Introduction

The cybersecurity workforce gap grows larger every year, reaching up to 4.07 million in 2019 [1]. There is an increasing need for trained professionals that are able to design secure systems from the ground-up, taking into account modern threats, as well as ever stricter laws whose aim is to safeguard customer information and the associated challenges in terms of their implementation [2]. The number of cyberattacks increases rapidly [3], with numerous examples of high profile cases making headline news, affecting the everyday lives of thousands of users and incurring millions in financial losses [4, 5].

For these reasons, there has been a steady increase in the number of cybersecurity initiatives, with many programming and system design courses including security-related modules in their lectures, and increasing the overall amount of cybersecurity related programs, thus raising the security awareness even higher and bridging the

current workforce gap [6]. However, it is very important that any training is augmented with real-life examples and scenarios, so that the future cybersecurity professionals can get hands-on experience. It has been shown that learners' performance improves when there is an opportunity to experience first-hand the theoretical concepts [7].

Designing material with the aim to offer as much practical experience to participants as possible has proven challenging in the past. Namely, there are a lot of hurdles that are either significantly time consuming or difficult to overcome (not only in terms of complexity, but also financially), such as: 1) obtaining, installing and maintaining the underlying infrastructure where the exercises would be performed; 2) designing the exercises themselves and seamlessly integrating them both in the syllabus, as well as with the existing computer systems (e.g. assessment, training management) [8]; 3) ensuring the security and integrity of the system itself.

Despite all of these challenges, there are many examples of such systems being designed and used in the training process of new cybersecurity professionals [9–12]. Recently there has also been an increase in free or freemium [13] solutions which offer cybersecurity challenges to interested users. Entire online communities have been formed, such as the Open Web Application Security Project (OWASP) [14] whose aim is to produce articles, documentation, and open-source tools aimed at web application security. One of their most well-known contribution is the OWASP top 10 [15] list, which outlines the most common vulnerabilities in web applications today.

No matter the implementational details, all of these solutions share a common cause, the desire to raise the awareness for cybersecurity by offering hands-on experience, and in the process of doing so, stimulate adversarial thinking using an experimental approach [16]. In many cases this is reinforced by the psychological impact of the gamification aspect of the challenges and their induced competitive nature, i.e. they are structured in a game-like manner using objectives, points, and leaderboards.

The aim of this paper is to present, describe and compare existing cybersecurity training platforms and services and offer a standardized taxonomy for their classification. The rest of the paper is organized as follows: in section two we present the criteria for selecting a software for analysis, describe the evaluation methods, and offer a taxonomy which should aid in the classification. In section three, a brief overview of each solution is given, exploring its features and license model. In section four we compare and summarize their advantages and disadvantages. Finally, we conclude the paper with closing remarks and future research.

2 Methodology

The recent rise in the number of cybersecurity related solutions paved the way for new use-cases and scenarios. With the advances in virtualization technology and the ubiquitous nature of cloud computing, it is now possible to create full-fledged scenarios reminiscent of vulnerable systems found in the wild. For example, many training platforms are offering on-demand virtual machines susceptible to known attacks, that can be probed for weaknesses and exploited by participating users. This approach is in stark contrast to the more traditional web-based challenges approach. Furthermore,

in order to increase the competitiveness among the participants, it is common to offer capture-the-flag style challenges, inspired by major cybersecurity events [17], where the user is tasked with obtaining a unique token (flag), thus validating the successful completion of the task at hand.

While there are numerous papers discussing the creation and implementation of such systems, or the integration of an existing open-source solution in cybersecurity courses [18], we are not aware of any work whose aim is to offer a comprehensive analysis of what is currently available.

During our research, two distinct types of systems were identified, which will be distinguished as either platforms or services. In the case of a platform, the system itself is completely extensible and additional challenges can be added with minimal effort. One such example is a system that offers its participants to define scenarios that are comprised of virtual machines, containers, or even allow the upload of the source code of a web application. In all cases either a known vulnerable software, or an unpatched version of a popular application is run, which others could then attempt to exploit. In this scenario, the system itself only acts as a middleman between the content creators and the content consumers and does not define the content itself. On the other hand, systems which themselves represent the vulnerable application that needs to be exploited, such as dedicated vulnerable training websites focusing on a finite number of scenarios, offering no extension path, are classified as services.

Taking into account the vast number of solutions currently available, mainly differentiated as being either cloud-based (hosted) or open-source (self-hosted), we have done the initial selection process based on the Alexa rank [19], for hosted services and on the number of GitHub stars [20], for self-hosted solutions. During this process, close attention was paid to include services and platforms from both camps, hosted and self-hosted, as to be able to compare the offerings between them. After manually testing each solution, and determining the advantages, disadvantages, and similarities with each other, we have identified a set of common parameters that are shared between them, discussed in detail below.

Two main types of systems are distinguished based on who is responsible for their hosting: 1) cloud-based, where the user can optionally register before using the service as-is; 2) self-hosted, where the source code can be used locally.

Regarding the product licensing, four different options have been identified: either completely free, with all features available to the end-user or, alternatively, a license fee has to be paid upfront before the user can access the challenges or download the necessary source files. A slight variation of these two options is the freemium model, where the user has access to certain introductory challenges, but then must upgrade to some form of premium membership before viewing the rest of the content. Finally, there are numerous examples of open-source solutions that also do not require any fees and can be freely downloaded and used by the end-users.

The number of available challenge types varies significantly, but the most common are: a) OWASP top 10 security risks which include SQL injection, cross-site scripting, authentication issues; b) networking challenges in the form of packet sniffing, man-in-the-middle attacks and replay attacks; c) exploitation challenges that aim to use a known vulnerability in some operating system or software package to gain un-

privileged access to a remote resource; d) cryptography challenges that range from breaking simple cryptographic ciphers, to more complex ones, such as exploiting a vulnerability in a popular application.

The vast number of exercises, as well as their diversity, make the answer verification process (whether the user has submitted the correct answer) quite challenging. The most common patterns seen across the different solutions are either manual verification or automatic verification. Keeping in mind that some of the challenges can be quite complex, it is not rare to see cases where the only verification option is to manually acknowledge that the challenge has been solved by simply clicking a button or revealing the correct answer after a period of time has passed. This value then needs to be manually pasted in a verification box. In this scenario it is up to the user to refrain from abusing this option and prematurely revealing the right answer without trying to complete the challenge by themselves. The other option is automatic verification where the system can automatically mark the challenge as passed after detecting that the necessary modification has been made to some watched resource (e.g. a new super user has appeared after exploiting an SQL-injection vulnerability). Of course, this is not possible with all challenge types. Finally, as a slight variation of the automatic option where the user has to verify the answer by themselves, an alternative approach is given, where another user, usually the administrator, has to check the working environment and verify that the challenge has been successfully solved before awarding any points. Should the user have issues with any of the challenges, in most cases, a collaborative space in the form of forums or dedicated chat rooms is available. Regardless of the communication method, sharing of complete solutions is strictly forbidden, keeping in line with the competitive nature of the challenges.

Another differentiating factor between the various training systems is the way that the challenges are presented to the end-user. Some of them simply offer the necessary source file where the task is described, without presenting any environment where the users can try to solve the challenge. However, in most cases, there is a way in which a per-customer instance of the challenge can be created. This is either done by instantiating dedicated virtual machines and accessing them through a remote console, or by creating containers where the vulnerable software is sandboxed and remote access to the end user is provided. A variation of these approaches is the case where the raw virtual machine disks or container image files are provided to the end-user to be run locally. Finally, when it comes to web-based challenges, usually just a URL to the vulnerable application is provided, hiding any additional details about the hosting infrastructure from the users. Most of the sites take pride in the quality of their content and restrict the creation of new challenges either to administrators or to distinguished members who have accumulated a large amount of points through frequent usage. Others may be more focused on the community character and in this case either all registered members can create new content or the privilege for content moderation can be acquired for a fixed fee usually in the form of a donation or a recurring subscription. Since most of the solutions offer a leaderboard where the members are ranked, as well as an option to track the progress through the various challenges, users are usually required to register an account upon their first visit.

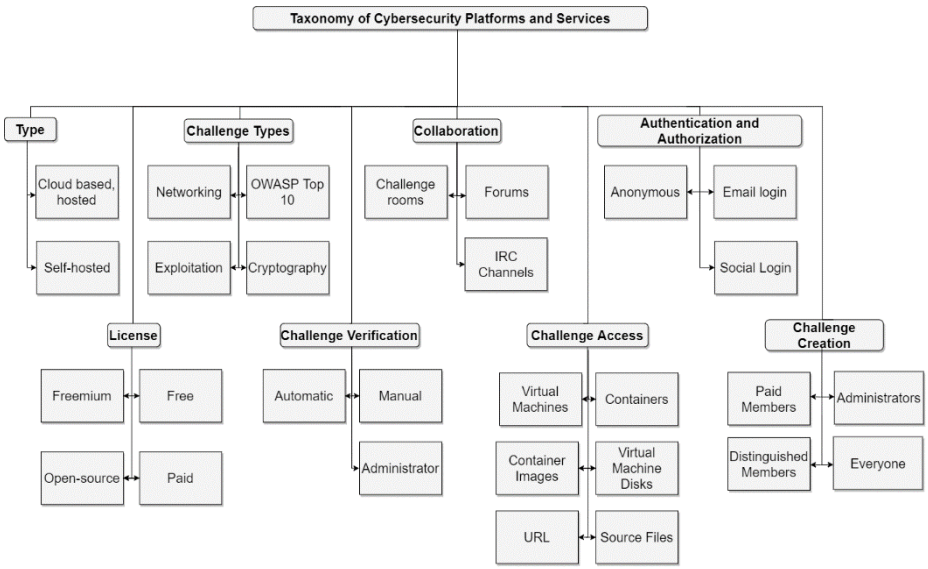

Fig. 1. Cybersecurity software taxonomy

This can either be done through a traditional email login or by using some of the widely popular social login options.

These observations were used as a starting point in the creation of the cybersecurity software taxonomy presented in Figure 1. The taxonomy consists of 8 categories with a total of 29 parameters, describing the nature of the solution, challenge type, licensing model, collaboration features, verification of the provided answers, authentication and authorization options, technology being used for the challenge environment as well as who can curate the content.

Using the popularity criteria discussed previously, we have narrowed down the initial selection of platforms and services to 13. In the section that follows, a brief overview of the selected solutions is provided, centered around the proposed taxonomy.

3 Solution Overview

The overview of the selected solutions has been divided into two general groups, cloud and self-hosted.

3.1 Cloud Solutions

We classify as cloud solutions all solutions that have a publicly accessible website where the user can list, preview, and attempt the challenges. In some cases, the way in which the challenges are accessed is mixed - there are both hosted challenges and challenges for which the user must download an additional virtual image locally.

Nevertheless, as long as the service has a web presence around which a community is built it is classified as a cloud service.

Enigma Group Challenges [21]. A well-known web site offering over 300 challenges with over 61000 members. The challenges are divided into seven categories:

- Basic challenges where the task is clearly stated, and it is up to the user to exploit the given vulnerability. Usually the basic challenges are centered around the OWASP top 10 and automatic answer verification is provided.
- Realistic challenges, where a real-world scenario is described, and it is up to the user to decide how to tackle it. These challenges are usually more involved since the user might have to install additional software, such as port scanning tools.
- Cryptographic challenges focus on encryption, decryption, brute forcing.
- Cracking challenges where the user is provided with only a binary file and is tasked with figuring out its purpose and how to exploit it.
- Auditing challenges where a block of source code is given and based on the detected bugs in it, the user should take appropriate action to exploit the vulnerability, for a purpose which has been stated beforehand.
- Patching challenges, similar to the auditing challenges, but now the user is tasked with fixing the bugs so that the application is no longer vulnerable.
- Programming challenges where a problem is stated, and the user must write a program solving it in a very short period of time, before a timer expires.
- Steganography challenges, where files in which hidden messages are present are downloaded by the user.

One of the main advantages of Enigma group is that automatic answer verification is provided. Depending on the complexity of the challenge the user is awarded points which count towards their rank. A lot of the challenges are restricted to paying members only. A public forum, an IRC channel and personal member blogs are available, all of which can be used for discussing challenges and sharing any cybersecurity related topics. Challenge creation is exclusively done by the administrators as to ensure their quality.

Hack Yourself First [22]. Hack Yourself First is different than most of the other services discussed in the sense that it is not a full-fledged platform where various challenges can be posted, graded and points awarded. Instead, it is simply a vulnerable website with over 50 security holes that the visitor can try and exploit, such as add new entries, impersonate other users, delete data, steal cookies. There are neither points awarded for successfully exploiting a vulnerability, nor any sort of validation. The user knows that the attempt was successful if the state of the website has been altered. The site offers account registration, but this is part of the scenario and it is up to the user to exploit the registration process. No official collaboration spaces are provided, but a video course [23] is available discussing the main concepts behind the vulnerabilities present on the site and reinforcing the message of how bad coding practices can negatively impact users. The underlying database is periodically rebuilt so that new users can start fresh.

Hack This Site [24]. Hack This Site is similar to the previously introduced Enigma group challenges, since they both offer various tasks which are divided into subgroups. In the case of Hack This Site the term mission is used to identify a challenge, and the mission groups are basic, realistic, application, programming, phone phreaking, JavaScript, forensic and steganography missions.

Most of the groups closely match the ones described in the Enigma group section, however, one of them is exclusive to Hack This Site - the phone phreaking missions. Here the users are tasked with the exploitation of a public branch exchange (PBX) telephone system, where they must use a real phone to conduct the mission.

The site requires the users to be registered before they can access any of its content and has a strong community character with regular blog posts, articles, and open forums where the users can discuss cybersecurity related topics. Most of the missions support automatic verification either through detection that the required actions have been performed or through capture the flag style answers, where the user has to provide the answer in a textbox before points are awarded. All of the challenges are free to access once the user has created an account; the site is funded by donations and through the online shop [25] where branded apparel as well as everyday home and office items are sold.

Root-Me [26]. Root-Me is a commercial platform that offers some of its content for free, while the rest is restricted behind a subscription paywall. Premium accounts can either be obtained by paying a monthly or yearly fee, or by paying a much smaller fee for a contributor's account and then creating new challenges on the platform. Challenge verification is done by submitting a flag that has been obtained and points are awarded for a successful competition of a challenge. Scoreboards and public rankings are available to encourage the competitiveness between the users. Each challenge has a hall-of-fame section where all the users that have previously completed the given challenge are listed.

Most of the challenges are hosted in virtual machines, where access is restricted to the IP addresses of the currently online members on the platform. Less experienced users have the option of taking a learning path which groups the available challenges and orders them based on difficulty and required prerequisite knowledge. In this way, novice users can first solve challenges related to programming basics, before moving on to networking concepts, and finally exploitation of web applications or other systems. Forums and an IRC channel are available so that users can seek help and discuss security related topics.

Try Hack Me [27]. Try Hack Me is one of the more advanced platforms that were evaluated. The challenges are divided into rooms. Each room is represented by a challenge description in a blog-post format where the vulnerability is described, and the necessary tools are listed. The users progress through a challenge by capturing flags during the exploitation process and validating them on the room's home page. Most of the rooms have virtual environments associated with them, where the vulnerable software is installed in a virtual machine accessible only to the user attempting the challenge. Access to the virtual machines is provided through a virtual private network (VPN) connection. Using this architecture, it is possible to host any type of

challenge, from exploitation of a vulnerability in an operating system's kernel to simpler web-based challenges.

The site offers a paid tier whose benefits are faster setup of the virtual environment and access to a wider selection of rooms. An additional benefit awarded to paying customers is a dedicated Kali Linux [28] virtual machine that users can access remotely. Login is only required for VM creation and answer validation, while the content of the free rooms can be accessed even without an account.

Any member can submit a new challenge, along with an associated virtual environment. It is up to the user to design the whole scenario and submit a virtual disk image to the platform, which would later be used as a master image for the environment creation. Also, many of the rooms freely offer the base image to the end-users, so that they can run the scenario locally instead of using the cloud resources of the platform. Any room can be forked, making a 1:1 copy of its content which can be improved by changing the scenario or modifying the base image. Each room has a dedicated forum associated with it, along with a writeups section where other users can explain the process of solving the presented challenges in detail. A scoreboard is used for listing the top performing users in each of the challenges.

Should the user want to learn a new skill, for example "web fundamentals", appropriate learning paths are provided, where through a series of exercises the fundamental concepts of a given topic are presented.

Hack Me [29]. Hack Me is similar to Try Hack Me in the sense that it allows any member to create full-fledged scenarios. However, unlike Try Hack Me, these scenarios are not sandboxed in a remote virtual environment and are limited to uploading PHP scripts. Additional required PHP libraries can be selected during the creation process as well as an optional MySQL instance, along with seed data. Once the challenge has been uploaded and made public, users can attempt to solve it. For each user, a unique sandbox of the scenario is created, accessible via a public random URL.

No native way in which answers can be validated is offered, it is up to the challenge developer to include verification logic in their scripts. A comment section hosted by a third-party service is available for each challenge. No subscription is required, except for a creation of a free account before the challenges can be accessed.

3.2 Self-hosted Solutions

Self-hosted solutions differ from their cloud counterparts in the sense that they offer no infrastructure of their own, instead only the source code is provided to the user. This is different from the option where a cloud solution would offer either the base virtual image for download or the source files of a vulnerable web application, since in the hosted case there is a dedicated location for challenge download, collaboration with other members and answer submission.

Even though the self-hosted experience might seem barebones and lacking at first, it offers greater flexibility since the user can inspect, modify, and adapt the code both of the software itself and any included challenges as they see fit. Additionally, this approach offers the option of altering the software so that it can be integrated with any

existing systems in a given corporation or institution, which can prove useful in terms of users import from an external database or export of any points to another system.

Of course, the main drawback is that users first need to have access to an infrastructure where they can host the applications, and also the necessary system administration experience to deploy them. This is not always a realistic expectation, especially in an academic environment where students might have different levels of system expertise. However, this can be overcome by either deploying a central instance internally, in the case where the application supports multiple users and can be used concurrently without any interference, or multiple instances which can be shared among teams of users.

OWASP Juice Shop [30], OWASP NodeGoat [31], OWASP Mutillidae II [32], OWASP WebGoat [33]. All of the OWASP Juice Shop, OWASP NodeGoat, OWASP Mutillidae II and OWASP WebGoat applications focus on presenting and educating the user about the OWASP top 10 vulnerabilities. The main difference between them is the technology stack with which they have been developed, allowing the user to choose the technology that they are most familiar with. This is a major advantage, having in mind that all of the applications are completely open-source and the user can either choose to treat the system as a black-box and discover and exploit the vulnerabilities exclusively through the web interface, or take a look at the source code and explore all of the bad code practices that have been employed, finding weaknesses along the way.

Many of these projects have also been extended by the community to offer additional modules and integrations. One such example is the OWASP Juice Shop command line tool [34] that can generate capture the flag style questions and answers for popular capture the flag contest hosting platforms like Facebook Capture the Flag and CTFd. This approach makes it possible to set up a local contest or to include these tools as part of a course with minimal effort required.

The installation procedure varies between the projects, but since in all cases the source code is readily available, it usually comes down to installing a database and the appropriate application server. Most of them also offer either dedicated installation scripts, virtual machine images, or Docker images as to ease the installation process.

Reinforcing the educational purpose of these applications, some of them have published instructional material that explains the vulnerabilities in detail, both from a theoretical point of view, as well as from a practical one. This material is available either as an ebook, as is the case with OWASP Juice Shop [35], or as sections within the application itself, like OWASP WebGoat.

Haaukins [36]. Haaukins uses YAML files and either Docker images or VirtualBox OVA files for the instantiation of new challenges. The YAML file specifies all of the container images or VirtualBox templates from which the scenario should be created, along with additional parameters, such as memory limits and environment variables. All of the containers and VMs that are part of a given challenge are isolated in a separate network. Users access a Kali Linux instance deployed inside this network through

a web RDP client. As was the case with other applications, Haaukins can also integrate with the popular CTFd platform for answer verification.

Unfortunately, the platform itself does not offer more advanced challenge customization through the YAML file itself, it is up to the image developer to integrate it into the base image. However, this simplifies the deployment since no additional orchestration tool is needed.

Facebook Capture the Flag [37]. Facebook Capture the Flag is a software for hosting capture the flag style contests using a video game-like interface. All of the questions are associated with a real-world country placed on a geographic map and it is up to the users who are divided into teams to try and capture as many territories as possible by correctly answering the accompanying questions. The software itself does not come with any prepopulated question bank, it is up to the administrators to develop questions that best reflect their type of event and usage scenario. One of the main benefits of the software is the possibility to import and export questions in popular markup formats, thus easing the integration process with external systems, as was the case with the aforementioned OWASP Juice Shop application, where an additional module is capable of generating questions regarding all of the present vulnerabilities.

Facebook Capture the Flag supports multiple ways in which user accounts can be created and authenticated, such as email based registration, restricted registration based on pre-generated invite codes, or integration with external systems such as the Lightweight Directory Access Protocol (LDAP). The challenge parameters are very granular and can be tweaked by the administrators, allowing the challenges to be time restricted, alter the number of points awarded depending on the complexity of the question, whether a given country can be conquered more than once (whether a given question can be answered multiple times), etc.

CTFd [38]. Another application for organizing and hosting capture the flag challenges, similar to Facebook Capture the Flag. However, CTFd supports more advanced question types such as unlockable challenges, where the user is required to correctly answer a given question before being allowed to progress to the next one; multiple choice challenges, where instead of providing the answer in a text field, the user can choose the correct answer from a list of given answers; dynamic challenges where each subsequent correct answer to a question lowers the amount of points awarded; manual verification challenges where a privileged user is required to grade the question and finally programming challenges. Programming challenges are the most advanced challenge type since the user is tasked with writing a program which needs to solve the described problem. Additionally, the platform is able to evaluate the code locally by supporting some of the most popular languages such as Java, Python, C/C++ and NodeJS. Once submitted, it is evaluated whether a correct solution has been provided for the given standard input (`stdin`) by matching the standard output (`stdout`) of the program with the expected `stdout` provided by the administrator during the challenge creation.

Similar to other offerings, users can also be grouped into teams and a dedicated statistics page is available providing useful information about the registered users and challenges.

Table 1. Software Comparison

		Enigma	HYF	HTS	Root-Me	TryHackMe	HackMe	OWASP JS	OWASP NG	OWASP ML	OWASP WG	Haaukins	Facebook CF	CTFd
Type		cb	cb	cb	cb	cb	cb	sh	sh	sh	sh	sh	sh	sh
License		fm	fr	fr	fm	fm	fr	os	os	os	os	os	os	os
C. Types	**n**	×	×	×	✓	✓	✓	×	×	×	×	✓	×	×
	o	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×
	e	✓	×	✓	✓	✓	✓	×	×	×	×	✓	×	×
	c	✓	×	×	✓	✓	✓	×	×	×	×	✓	×	×
C. Ver.	**au**	✓	×	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓
	m	×	✓	×	×	×	✓	×	✓	✓	✓	✓	×	×
	ad	×	×	×	×	×	×	×	×	×	×	×	×	✓
Collab.	**cr**	×	×	×	×	✓	✓	×	×	×	×	×	×	×
	f	✓	×	✓	✓	✓	×	×	×	×	×	×	×	×
	ic	✓	×	✓	✓	×	×	×	×	×	×	×	×	×
C. Access	**vm**	×	×	×	✓	✓	×	×	×	×	×	✓	×	×
	c	×	×	×	×	×	×	×	×	×	×	✓	×	×
	ci	×	×	×	×	×	×	×	×	×	×	✓	×	×
	vd	×	×	×	×	✓	×	×	×	×	×	✓	×	×
	u	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	sf	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓
AA	**a**	×	✓	×	×	×	✓	✓	×	×	✓	×	×	×
	e	✓	×	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
	s	×	×	×	×	×	×	✓	×	×	×	×	✓	×
CC	**pm**	×	×	×	✓	✓	×	×	×	×	×	×	×	×
	a	✓	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	✓
	dm	×	×	×	✓	×	×	×	×	×	×	×	×	×
	e	×	×	×	×	✓	✓	×	×	×	×	×	×	×
Rank		A. 3.3M	A. 2.4M	A. 93K	A. 91K	A. 533K	A. 734K	G. 2.9K	G. 1.1K	G. 330	G. 3.1K	G. 71	G. 6.2K	G. 2.5K

Type - cb (cloud-based) or sh (self-hosted); **License** - fm (freemium), fr (free), os (open-source), p (paid); **C. Types:** Challenge Types - n (networking), o (OWASP top 10), e (exploitation), c (cryptography); **C. Ver.:** Challenge Verification - au (automatic), m (manual), ad (administrator); **Collab:** Collaboration - cr (challenge rooms), f (forums), ic (irc chat); **C. Access:** Challenge Access - vm (virtual machine), c (container), ci (container images), vd (virtual machine disks), u (url), sf (source files), **AA**: Authentication and Authorization - a (anonymous), e (email login), s (social login); **CC:** Challenge Creation - pm (paid members), a (administrators) , dm (distinguished members), e (everyone); **Rank** - A (Alexa), G (GitHub stars).

4 Solution Comparison

Table 1 compares all of the applications that were discussed in the previous section using the taxonomy presented in Fig. 1. To increase readability, some of the parameters have been abbreviated, for which a detailed explanation is given in the footer of the table. A note should also be made about the automatic verification parameter when it comes to self-hosted applications. Even though many of them do not support automatic answer verification, a tick is present in the corresponding column if there is a possibility to integrate the application with an external system which can provide the answer checking functionality, for example a capture the flag hosting application. An additional column, not included in the taxonomy is the rank column, stating the relative popularity of the given service, where numbers prefixed with an A, in the case of cloud platforms, represent the Alexa rank of the service, while the ones prefixed with G the number of stars on the GitHub code hosting platform.

Based on the results presented in Table 1, according to their Alexa rank, cloud-based solutions where only static challenges are provided are less popular. This is due to the fact that they do not offer dedicated virtual machines or containers which users can access, unlike their counterparts which do offer their users full-fledged environments. It can be argued that challenge creation also plays a role, with community-based platforms such as Root-Me, Try Hack Me and Hack Me being more popular than their counterparts such as, Enigma Group Challenges and Hack Yourself First, where challenges are only created by the site administrators.

When it comes to the self-hosted solutions, Facebook CTF is a particularly popular option, whose popularity can be attributed to its interface design and the fact that it gives an additional incentive to the user with its world domination aspect. CTFd is an option that has either been integrated as an optional plugin or as a dependency to some of the other platforms, such as Haaukins. The applications under the OWASP umbrella also enjoy high popularity and very active development. Their main advantage is the fact that they are not merely vulnerable applications where the user has to possess prior knowledge to find and exploit the vulnerabilities, but also offer dedicated sections explaining bad coding practices and how they can be overcome. Some of the options, like Mutillidae II even offer a secure and non-secure mode of operation, along with web-based reset controls which can bring the application to its initial state after any database modifications.

While cloud-based solutions are easier to use and do not require extensive maintenance, almost all require payment to unlock additional features. Additionally, cloud security training platforms might be attractive targets for potential hackers which may lead to information leakage. To overcome these problems, free and open-source solutions need to be chosen with permissive licenses that allow altering of the source-code so that they can be integrated with existing systems.

In summary, solutions that offer a more realistic approach to the challenges, with virtual machines or containers are more popular and offer a better learning experience for the end-user. However, these options are also much more complex, dealing with issues such as provisioning, network isolation and remote access.

5 Closing Remarks

The increasing cybersecurity requirements of modern applications and the popularization of this topic through numerous high-profile breaches that have made newspaper headlines around the world, have led to the proliferation of training platforms and services that aim to provide challenging tasks as to enhance the practical cybersecurity experience globally. However, faced with so many options, users, as well as system integrators who would like to use some of these systems in their existing infrastructure, are faced with difficult choices, since no comparative analysis is available. In order to mitigate this, a survey of publicly available solutions was made and through the analysis of their characteristics and main features, a taxonomy was proposed which can aid in cybersecurity training software classification.

The proposed cybersecurity taxonomy includes parameters such as deployment type, license model, types of challenges supported, methods for verifying the challenge solutions, challenge deployment options, user authentication and authorization options and finally challenge creation. An overview of the two main application categories, cloud and self-hosted was given. A short description of 13 platforms and services, selected based on their popularity, followed by a more thorough discussion about the various advantages and disadvantages stemming from their chosen architecture was provided. Taken together, these resources should provide a more complete picture about the currently available cybersecurity training solutions, and the various use-cases supported.

Future work will focus on the design and implementation of a general-purpose cybersecurity training platform to be used as part of cybersecurity courses, which will be based on the best-practices and some of the open-source tools introduced.

References

1. (ISC)²: 2019 Cybersecurity Workforce Study. https://www.isc2.org/-/media/ISC2/Research/2019-Cybersecurity-Workforce-Study/ISC2-Cybersecurity-Workforce-Study-2019.ashx (2019). Accessed 26.02.20
2. Poritskiy, N., Oliveira, F., Almeida, F.: The benefits and challenges of general data protection regulation for the information technology sector. DPRG (2019). https://doi.org/10.1108/DPRG-05-2019-0039
3. Department for Digital, Culture, Media & Sport: Cyber Security Breaches Survey 2019. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/813599/Cyber_Security_Breaches_Survey_2019_-_Main_Report.pdf. Accessed 25.02.20
4. Ghafur, S., Kristensen, S., Honeyford, K., Martin, G., Darzi, A., Aylin, P.: A retrospective impact analysis of the WannaCry cyberattack on the NHS. NPJ digital medicine (2019). https://doi.org/10.1038/s41746-019-0161-6
5. Berghel, H.: Equifax and the Latest Round of Identity Theft Roulette. Computer (2017). https://doi.org/10.1109/MC.2017.4451227

6. Lopez-Cobo, M., De Prato, G., Alaveras, G., Righi, R., Samoili, S., Hradec, J., Ziemba, L., Pogorzelska, K., Cardona, M.: Academic offer and demand for advanced profiles in the EU. Artificial Intelligence, High Performance Computing and Cybersecurity, JRC113966. Joint Research Centre (Seville site). http://publications.jrc.ec.europa.eu/repository/handle/JRC113966 (2019)
7. Bell, R.S., Sayre, E.C., Vasserman, E.Y.: A Longitudinal Study of Students in an Introductory Cybersecurity Course. In: 2014 ASEE Annual Conference & Exposition. ASEE Conferences, Indianapolis, Indiana (2014)
8. Shumba, R.: Towards a more effective way of teaching a cybersecurity basics course. SIGCSE Bull. (2004). https://doi.org/10.1145/1041624.1041671
9. Furfaro, A., Piccolo, A., Parise, A., Argento, L., Saccà, D.: A Cloud-based platform for the emulation of complex cybersecurity scenarios. Future Generation Computer Systems (2018). https://doi.org/10.1016/j.future.2018.07.025
10. Acosta, J.C., McKee, J., Fielder, A., Salamah, S.: A platform for evaluator-centric cybersecurity training and data acquisition. In: MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM). 2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, 23.10.17 - 25.10.17, pp. 394–399. IEEE (23.10.17 - 25.10.17). https://doi.org/10.1109/MILCOM.2017.8170768
11. Kalyanam, R., Yang, B.: Try-CybSI: An Extensible Cybersecurity Learning and Demonstration Platform. In: Zilora, S., Ayers, T., Bogaard, D. (eds.) Proceedings of the 18th Annual Conference on Information Technology Education - SIGITE '17. the 18th Annual Conference, Rochester, New York, USA, 04.10.17 - 07.10.17, pp. 41–46. ACM Press, New York, New York, USA (2017). https://doi.org/10.1145/3125659.3125683
12. Mirkovic, J., Benzel, T.: Teaching Cybersecurity with DeterLab. IEEE Secur. Privacy (2012). https://doi.org/10.1109/MSP.2012.23
13. Kim, W.: A practical guide for understanding online business models. Int J of Web Info Systems (2019). https://doi.org/10.1108/IJWIS-07-2018-0060
14. OWASP Foundation, the Open Source Foundation for Application Security. https://owasp.org/. Accessed 27.02.20
15. OWASP Top 10. https://owasp.org/www-project-top-ten/. Accessed 19.02.20
16. Schneider, F.B.: Cybersecurity Education in Universities. IEEE Secur. Privacy (2013). https://doi.org/10.1109/MSP.2013.84
17. Nunes, E., Kulkarni, N., Shakarian, P., Ruef, A., Little, J.: Cyber-Deception and Attribution in Capture-the-Flag Exercises. In: Jajodia, S., Subrahmanian, V.S., Swarup, V., Wang, C. (eds.) Cyber Deception, vol. 23, pp. 149–165. Springer International Publishing, Cham (2016)
18. Chicone, R., Burton, T.M., Huston, J.A.: Using Facebook's Open Source Capture the Flag Platform as a Hands-on Learning and Assessment Tool for Cybersecurity Education. International Journal of Conceptual Structures and Smart Applications (2018). https://doi.org/10.4018/IJCSSA.2018010102
19. Alexa. Keyword Research, Competitive Analysis and Website Ranking. https://www.alexa.com/. Accessed 25.02.20

20. GitHub Stars. https://help.github.com/en/enterprise/2.13/user/articles/about-stars. Accessed 27.02.20
21. Enigma Group Challenges. Web application security training. https://www.enigmagroup.org/. Accessed 18.02.20
22. Hack Yourself First. https://hack-yourself-first.com/. Accessed 18.02.20
23. Hunt, T.: Hack Yourself First: How to go on the Cyber-Offense. https://app.pluralsight.com/library/courses/hack-yourself-first/table-of-contents. Accessed 19.02.20
24. Hack This Site. https://www.hackthissite.org/. Accessed 18.02.20
25. Hack This Site Online Shop. https://www.cafepress.com/htsstore. Accessed 25.02.20
26. Root Me. https://www.root-me.org/. Accessed 18.02.20
27. Try Hack Me. https://tryhackme.com/. Accessed 18.02.20
28. Kali Linux. Penetration Testing and Ethical Hacking Linux Distribution. https://www.kali.org/. Accessed 25.02.20
29. Hack Me. https://hack.me/. Accessed 18.02.20
30. OWASP Juice Shop. https://owasp.org/www-project-juice-shop/. Accessed 18.02.20
31. OWASP NodeGoat. https://owasp.org/www-project-node.js-goat/. Accessed 18.02.20
32. OWASP Mutillidae II. https://github.com/webpwnized/mutillidae. Accessed 18.02.20
33. OWASP WebGoat. https://owasp.org/www-project-webgoat/. Accessed 18.02.20
34. OWASP Juice Shop CTF CLI. https://www.npmjs.com/package/juice-shop-ctf-cli. Accessed 25.02.20
35. Kimminich, B.: Pwning OWASP Juice Shop (2019)
36. Haaukins. A Highly Accessible and Automated Virtualization Platform for Security Education. https://github.com/aau-network-security/haaukins. Accessed 18.02.20
37. FBCTF. Platform to host Capture the Flag competitions. https://github.com/facebook/fbctf. Accessed 18.02.20
38. CTFd. https://github.com/CTFd/CTFd. Accessed 18.02.20