Capture-The-Flag Framework and Virtual Environment for Cyber Security Education

Jadiel A. Colón Pérez
Master in Computer Science
Advisor: Dr. Jeffrey Duffany
Electrical & Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract *— When studying cybersecurity or other computer-related technical careers, a problem among the students is the lack of hands-on experience. This can be solved by participating in internships, conferences, and competitions. Internship opportunities and local conferences are limited, and not every student has the flexibility to leave their homes, families, and even day jobs for a prolonged period. The solution that is being proposed is to incorporate a virtual environment that includes exercises like a Capture-The-Flag competition, which are contests designed to serve as an educational exercise to give participants experience in securing a machine, as well as conducting and reacting to the sort of attacks found in the real world. This environment will be coupled with a framework that allows students to submit the found flags and be evaluated instantly. Although it bears a resemblance to a Capture-The-Flag competition, this project is meant to be a form of guided self-learning.*

Key Terms *— Capture-The-Flag, Computer Science, Cryptography, Cyber Security.*

BACKGROUND

Many competitions are organized around the world as capture the flag (CTF) events. They require students and/or participants to solve problems, earn points, and thus demonstrate their skills in different areas of cybersecurity [1]. They consist of a series of challenges that vary in their degree of difficulty and require participants to apply different sets of skills. The development of a learning environment framework in the form of a CTF competition is what is being discussed in this project. As part of the creation and validation of the framework, three CTF frameworks were analyzed and one was chosen. The frameworks that were considered were: Facebook capture the Flag, CTFd, and Mellivora. A CTF based on the Mellivora framework was chosen for several reasons: simple to use, light, very fast, and fits our required needs.

For a long time, competitions have driven economies, research, and knowledge itself. The same could apply to education [1]-[2]. Its incorporation into an environment that is directly related to what students are learning in class could prove to be beneficial. The challenges include: network traffic analysis, steganography, open source intelligence, cryptography, among others.

PROJECT GOALS & SIGNIFICANCE

Capture the flag challenges can result in an increase interest from students into cybersecurity [3]. Our implemented framework serves to allow these students to immerse themselves into hands-on exercises within different levels of difficulty: beginner, intermediate, and advanced. It will provide them with the ability to use and gain important insight for several different cybersecurity software: Metasploit, Nmap, Wireshark and others. The framework will become an essential tool for participants and it will fulfil the following goals:

- Assist participants in developing a solid foundation of cybersecurity related threats and concepts.
- Assist participants in developing a high level of proficiency in several cybersecurity software tools.
- Provide participants from diverse backgrounds and experience with an adequate difficulty level.
- Encourage participants to enroll into regional and international Capture-The-Flag competitions.
- Assist participants to acquire the technical skills that employers currently desire.

- Help students familiarize with the linux terminal and other non-Windows-related tools and software.

INTRODUCTION

Although the concept of CTF (Capture-The-Flag) competitions has existed for a long time now, these events came to prominence in 1996, when it was announced that one would be hosted at Defcon 4, one of the largest hacker conferences in the world [4]. Since then, many other CTF competitions have been developed by different organizations, but the Defcon one remains one of the oldest and most respected ones out there. In a way, it was at Defcon 4 that CTF was formalized into a contest, since there were judges now, who decided when points should be awarded.

Back then, the events were not as organized as we know them today. As an example, in Defcon 5 and 6, participants could choose between providing targets or attacking targets provided by others to earn points [5]. This amount of flexibility in terms of the tasks to be performed at the competition proved to be unfavorable, therefore, the Defcon 5 and 6 CTF competitions are regarded as being highly disorganized. Since then, the game has advanced and grown into something almost completely different. The point scoring structures have become mostly automated, and organizers for the different competitions are named early as to give them time to arrange the necessary infrastructure.

The very foundation of a CTF competition is meant to test computer and network security knowledge. Although that may not seem like a very broad area, it's important to realize that cyber security is a very large and diverse field [6]. It has been found to be impractical to try and cover as many topics as possible in a CTF competition. Instead, it's a better practice to simply cover a few areas with varying degrees of difficulty. As this practice became more common and teams started playing more regularly, many CTF competitions implemented a method of qualifying the participants. In the case of the Defcon, a qualification weekend pits the teams against each other, and against the clock. The teams that end up with the most points are invited to participate in person at the actual Defcon.

In modern times, CTF competitions are usually composed of some subset of the following categories: poorly implemented or configured crypto software or algorithms, SQL-injections, Cross-site-scripting, buffer overflows, timing attacks, heap exploits, malformed network constructs, custom interpreters, logic problems, steganography, base conversions, among many other possible exercises. Furthermore, there are different types of CTF challenges, the main ones being the attack-defense CTFs, Jeopardy-style CTFs, and a mixture between the previous.

Attack-Defense CTFs

Out of the most popular CTF types, attack and defense CTFs are the least common. This is because they are a lot more complex than other CTF types and have more moving parts, figuratively speaking. Because of the previous reasons, attack-defense CTFs are rarely done for the general public. The basic principle of attack-defense CTFs is that each team is given the same set of vulnerable server software. Teams configure and analyze the software they are given before the actual competition starts [7].

Once the competition begins, the teams usually connect their servers to an isolated and self-contained network, so they can join the CTF. Once inside the network, the teams must scan other team's servers in order to find vulnerabilities that they can leverage. The teams must then launch attacks against other teams by exploiting the vulnerabilities that they found. Likewise, the teams need to scan the different services and software that they are provided so that they can patch them and avoid attacks from the other teams, while still carrying out the required functions. As to the point award structure, the different teams receive points for extracting other team's flags, patching their software and monitoring the vulnerabilities in their

systems for the purpose of protecting their own flags, and most importantly, to keep their servers working correctly after all the patches and updates made to their systems.

Jeopardy-style CTFs

Jeopardy CTFs are the most common kind of CTFs. These usually involve certain challenges which are provided for the competitors by the people that are in charge of organizing the event [5]. The teams are usually assembled by the competitors themselves, although in some occasions, competitors may be placed in teams by organizers. Once the teams are formed and the competition formally starts, the competitors must solve each challenge in order to unlock a flag in the form of a small piece of text. The flag is then copied and submitted in some way, like email, website, or scoring engine, to judges or reviewers in exchange for points. The challenges usually range from very easy, to hard. Because of this, the points awarded are typically based on the set difficulty for the exercises. In jeopardy-style CTFs, competitors have a set time, usually between 24 to 72 hours, to complete as many challenges as possible. Scoring could be as straight forward as getting the exact amount of points assigned to each exercise and adding them up, or there can be complex formulas behind each category.

Mixed CTFs

Mixed CTFs have elements of both attack-defense CTFs and Jeopardy-style CTFs. Many CTF competitions use a Jeopardy-style CTF for the team qualification round, and then a much more complex attack-defense CTF for the final round. Rules for mixed CTFs might also be different than those of other CTF competitions, due to its flexible nature. In mixed CTFs, the contestants can often solve trivia questions, attack other team's machines, and in some instances, hacking the CTF itself is allowed and scored.

A classic example of a mixed CTF is the current configuration of the Defcon CTF. Since the infrastructure for hosting the attack-defense CTF is limited, contestants must first go through a Jeopardy-style CTF to qualify for the finals. Once in the finals, the teams compete in the attack-defense competition. Because of the relaxed rules of the event, a group called ddtek managed to work its way into becoming the organizers of the event during Defcon 17. There were ten teams competing that year, but nobody suspected that the people sitting in team sk3wl0fr00t were the ones running the competition instead of being just another competitor, since ddtek was a subgroup of sk3wl0fr00t.

IMPLEMENTATION

The project consists of two parts. The first was to develop the actual CTF framework, where the participants would be submitting the found flags. This involved studying different CTF frameworks and choosing the one that best fit the project goals. The second part of the project was to build the actual virtual machines that would host the vulnerabilities that the participants would need to exploit, as well as different exercises that the students need to solve so that they can extract flags to be submitted in the framework.

The CTF Framework

Because of the different needs and requirements of the environment, a custom CTF framework had to be built by using existent ones as reference. For this purpose, different CTF frameworks were evaluated. Some of these frameworks were the Facebook CTF framework, CTFd framework, and the Mellivora framework. The CTFd framework was discarded because, although it had a nice scoreboard, it costed around $100.00 dollars a month to license, and it was more complicated to customize and edit than anticipated. For the Facebook CTF, the caveats encountered were that it was only available for Ubuntu 16.4 x64, it also had database configuration problems, and the interface was too complex. In the end, it was decided that the best choice was the Mellivora framework. This decision was made for several

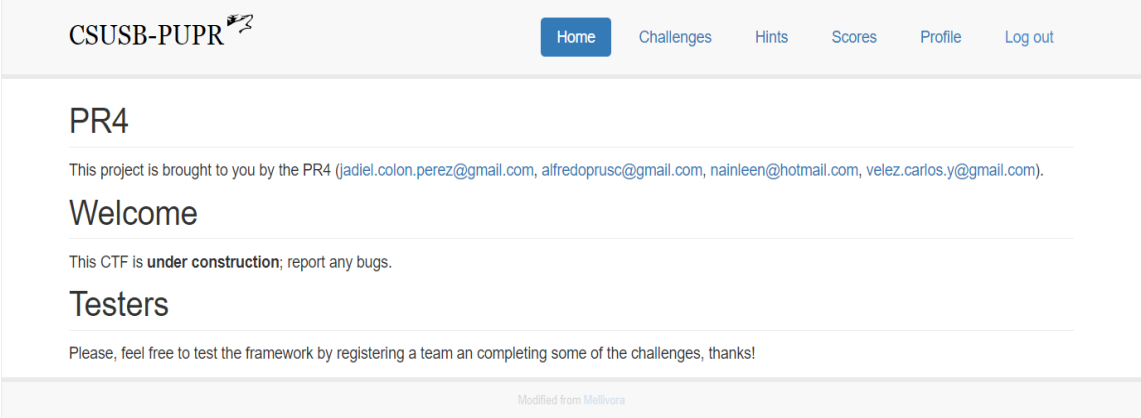

Figure 1
Finished Framework Layout

reasons, like the fact that the Mellivora framework is written in PHP which is a simple language. It also works with a MySQL database, and the project itself is easy to deploy in an Apache Webserver. No public scoreboard was available for it, but it was added after the framework was implemented.

The interface of the finished framework can be seen in figure 1. A normal user would see the six buttons listed in the navigation bar. These are the home button, the challenges button, the hints button, the scores button, the profile button, and the log out button. An administrator has additional buttons that help him or her manage the different challenges and options for all of them. When a user presses the home button, he is taken to the welcome page, where information about the CTF framework and virtual environment developers is given. They also see several announcements informing him that the CTF is still under development, and that he can feel free to test the framework and challenges currently available. The information gathered from user tests has been extremely valuable for determining the kind of changes and improvements that need to be made in the framework. In the challenges tab, all the available challenges can be seen, as well as the time limits, constraints, and the number of attempts permitted per each challenge. The scores tab, on the other hand, displays each player's individual score and compares it to other players. It also displays other milestones, like who was the first to solve a certain challenge, and time records. The profile tab allows the user to update their personal information. They could change their display name, which is what other students see on the scoreboard, they could change their contact information, like their email, for example, and they could also change their login information, like passwords.

On the hints tab, students will get certain clues as to how to solve the puzzle or exercise that they are being given. On usual CTFs, the clues are not too revealing, because it is of paramount importance to preserve the competitive nature of the event. The challenges range from easy to hard, so the clues must not interfere by making the challenges easier than they should be. The framework being developed in this project, however, has a different purpose. As stated before, this framework's purpose is not just competitive in nature, but also a means of guided self-learning for students. The clues posted in the hints section are meant to point the students towards assigned readings, specific pages in a text book, provided video tutorials, web pages, or other kinds of materials related to the challenges being presented. This way, the student can find the answer to a given problem, while also coming about the required knowledge on his own.

As part of the framework, it is also possible to upload files that could be included as part of a certain challenge. These files could be password hashes that the students would have to crack by using a dictionary attack or rainbow tables in order to find the plaintext password, which could be used

as a flag. Another possibility is chaining several exercises together, so that the answer to one exercise is the beginning of another exercise. Using the example above, should the student be able to get the plaintext password from the file, then that password could be used to access a virtual machine.

The Virtual Environment

For the virtual environment that was to accompany the CTF framework, it was important to include exercises and vulnerabilities that still represented real threats, even if they did so in a different way than presented in the project.

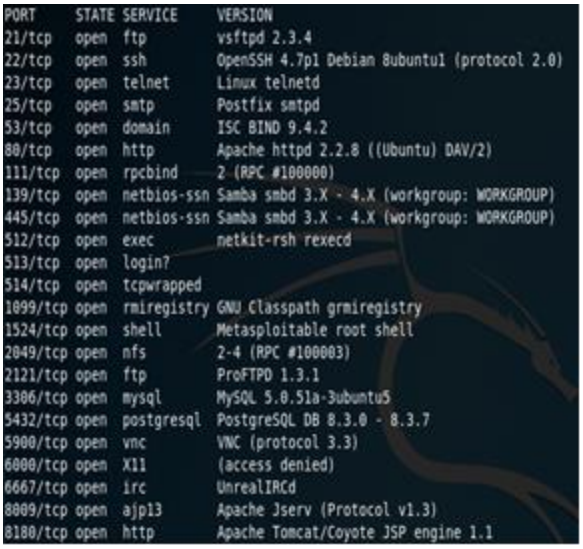

Figure 2
Partial Scan of the Vulnerable Services

It was decided that a metasploitable virtual machine should be included in the virtual environment. In figure 2, there's a partial list of the vulnerable services that are running on the metasploitable machine.

A common problem that still affects developers and users alike is when hackers hijack an applications code and insert backdoors or malicious code into them. To demonstrate this vulnerability, the vsftpd includes an unintentional backdoor that can be exploited both manually, and by using a Metasploit module. Misconfigured services are also a security risk that plagues many businesses even today. In the TCP ports 512, 513, and 514, there are services known as "r" services. These have been purposely misconfigured to allow access to any remote host. A simple client called the "rsh-client" is all that is required to take advantage of these simple vulnerabilities. On port 6667, there is a vulnerability that went unnoticed for a long time on a popular service. The UnrealRCD IRC daemon service contains a backdoor that is triggered by sending "AB" followed by a system command on any listening port. This can be done manually or by the use of a preexisting Metasploit module, which can be used to gain an interactive shell. In port 1524, whose service is listed simply as "shell", is actually a backdoor known as "ingreslock".

Another common problem is the use of weak passwords [8]. Some of the accounts, including the root account, have a very weak password that can be broken by a simple brute force attack or a simple password list. Some of these passwords are "user", "batman", "123456789", and "service". The database services, like the PostgreSQL, can be accessed with the username "postgres", and the password "postgres", while the MySQL service can be accessed by "root" with a blank password.

Perhaps the most important vulnerabilities are in the Apache web server, on port 80. It has different web pages, each of which is susceptible to different types of attacks. SQL injections and Cross site scripting were featured in the OWASP 2010 and remain in the 2017 updated list as two of the biggest risks to web applications [3]. Besides these, other vulnerabilities that can be found on the web server include JavaScript validation bypass, Log Injection, system file compromise, unencrypted database credentials, JSON injection, and parameter pollution.

The attacking machine that would be used in the virtual environment was chosen to be Kali linux from the very beginning. This was determined taking into account that most of the tools that would be needed in order to solve the challenges already come preinstalled in Kali. Also, knowledge of Kali linux and the penetration testing tools that it contains is essential for any cyber security professional. As the virtual environment progresses and more exercises and exploitable machines are

added to the network, it's safe to say that Kali linux and its tools will stay current, and this will inherently help the team working on the virtual environment save valuable time by not having to build new images of the attacking machines every time a minor update or change is added.

RESULTS AND DISCUSSION

The environment and framework were initially developed by and for the NSF Scholarship for Service students as a means of practicing for other CTF competitions. Faculty and student training was later identified as another opportunity. There are several college courses that could benefit from a hands-on laboratory in the form of a CTF competition, but the ones that could fit best with what has currently been developed are Network Security (CECS 7230), Computer Forensics (CECS 7235), and Computer Security (CECS 7570). All of these courses belong to the Master Program in Computer Science, IT Management and Information Assurance area of interest, which is approved by the NSA. Preliminary testing of the virtual environment demonstrates that the total amount of machines per each user still requires a lot of resources. Around 3 to 4 GB of RAM are necessary to keep each user's subnet functioning properly. Although the exploits work correctly, and the virtual environments are completely reproducible, scaling the environment to the point where it can be used in an actual CTF challenge or full college course would require specialized infrastructure, and still presents a challenge. Agreements are being made in order to host the virtual environment externally.

FUTURE WORKS

In the future, the virtual environment should be adapted to handle NCL and CCDC type competitions. Online compilation is also in the process of being added to the framework, so that coding challenges can be included. The challenges also need to be modified so that they can resemble real-world scenarios. Vulnerabilities are in the process of being added and updated. The new machines should be deployed and tested inside the environment. Once the environment reaches a larger scale, a regional CTF competition should be hosted in order to get people interested in the topics related to cyber security.

ACKNOWLEDGEMENT

This material is based upon work supported by, or in part by the National Science Foundation Scholarship for Service (NSF-SFS) award under contract/award #1563978.

REFERENCES

[1] K. Leune and S. Petrilli, "Using Capture-the-Flag to Enhance the Effectiveness of Cybersecurity Education", in *18th Annual Conference on Information Technology Education*, SIGITE, 2017, pp. 3.

[2] V. Ford, et al. "Capture the Flag Unplugged: An Offline Cyber Competition", in *Technical Symposium on Computer Science Education*, SIGCSE, 2017, pp. 228.

[3] N. C. Idika, "Maximizing network security given a limited budget," in *TAPIA '09 The Fifth Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations*, Portland, 2009.

[4] National Cyber League. (2017). *About NCL* [Online]. Available: https://www.nationalcyberleague.org/about. [Accessed: May 18, 2018].

[5] Over The Wire. (n. d.). *Wargames* [Online]. Available: http://overthewire.org/wargames/. [Accessed: May 18, 2018].

[6] S. Cobb. (2016, October 6). "Mind This Gap: Criminal Hacking and the Global Cybersecurity Skills Shortage, a Critical Analysis," in *Virus Bulletin* [Online]. Available: https://www.virusbulletin.com/conference/vb2016/abstracts/mind-gap-criminal-hacking-and-global-cybersecurity-skills-shortage-critical-analysis. [Accessed: May 17, 2018].

[7] National Collegiate Cyber Defense Competition. (2018). *History of CCDC* [Online]. Available: http://www.nationalccdc.org/index.php/competition/about-ccdc. [Accessed: May 17, 2018].

[8] H. Venter and J. H. Eloff, "Vulnerability Assessment: Assessment of Vulnerability Scanners," in *Network Security*, vol. 2003, no. 2, February 2003, pp. 11-16.