Lowering the Barriers to Capture The Flag Administration and Participation

Kevin Chung, *CTFd LLC*

Abstract

Capture The Flag (CTF) competitions have a rich history of incredibly technical individuals providing information security resources for each other. CTFs have been used by the information security community for education and assessment for over a decade. They're widely regarded as an excellent introduction to the information security industry given their competitive aspect, team building nature, and lack of long-term commitment.

CTF organizers have long experimented with exploring its relevance in areas beyond technical competitions. The foundation of CTF is rooted in technical education but one of the desired evolutions of CTF is as an e-sport [1]. In this paper, we present our perspective on why the current model of CTF is not a viable e-sport. This perspective will be presented alongside ideas around advancing the adoption of CTF, and CTFd, a readily available open-source framework for educators, recruiters, and companies to integrate CTFs into their pipelines.

CTFd eases the amount of development needed to bring a CTF to fruition. Aside from the challenge views for standard CTF functionality, it features score graphs, an administration panel, a built-in hint economy, and archival functionality. In addition, CTFd supports plugins and themes for additional customizability. This paper will be presented alongside a demonstration of the features of CTFd and different ways it can be integrated in computer education.

1. Introduction to Capture The Flag

Capture The Flags (CTFs) are a kind of information security competition. In a CTF, teams are provided a variety of problems (known as challenges). Each challenge contains some form of security vulnerability or security-related task that must be exploited or completed. Upon completion, the challenge will yield higher levels of access or reveal an answer. This answer can then be traded with a scoring system for tracking.

Capture The Flag competitions have been used in the security community since at least 1999. The first DEFCON CTF was in 1999 and many other CTF competitions have taken place since then such as CSAW CTF, PlaidCTF, PicoCTF, and UCSB iCTF amongst others. The amount of CTFs taking place each year has increased over time [1]. In the year 2016, more than 100 CTF events were recorded by the CTF tracking website CTFTime [2].

Capture The Flags generally follow two kinds of formats, Jeopardy and Attack/Defense [3] [4]. While CTFd is designed primarily for the Jeopardy format, the Attack/Defense format can be supported through its plugin interface.

The Jeopardy format presents a variety of challenges to teams of competitors on a game board. The teams then solve each of the problems and submit the "flags" or proof of solution to the scoring system (usually a website) to gain points respective to each challenge's difficulty. The team with the most points at the end of the competition wins.

The Attack/Defense format provides each team with a set of vulnerable services that are networked together, accessible by other team's services. Each team must find vulnerabilities in the shared services and exploit them on the other teams' infrastructure to steal and submit flags. In addition, each team must patch the vulnerabilities they find in their own services to protect their own flags. The team with the most service uptime and flags at the end of the competition wins.

1.1. The Problem with Capture The Flag

While some consider the non-technical aspects of CTF (e.g. team organization, challenge hints) to be a metagame [1], CTFs do not have an overarching competitive scheme dictated by game mechanics. For example, the game of chess has established rules for each piece, and Counter-Strike has an established meta-based on the properties of each gun in the game. Most games have some central underlying theory that shapes the choices and strategy made by the player.

There is an overall format as each CTF mimics the style, challenge categories, and interface of the CTFs run before it. Unfortunately, within that format there are many different challenges that can vary widely in scope, difficulty, and solution. Additionally, there is no standardized toolset, challenge format, or set of defined game rules to limit what a competitor or organizer can do.

Because of this lack of metagame, combined with a massive learning curve, current CTFs may, ultimately, never reach a widespread audience. Improvements need to be made by defining an overall structure for Capture The Flag games such that it improves the underlying approachability of the game. Without an increase in approachability, it will be difficult to increase the player base of CTFs.

The game Payday 2 by Overkill Software represents an interesting example of game development where random events and chosen paths influence the difficulty of a given situation. For example, completing a "map" without making any noise is difficult, but much quicker than opening fire against opponents. A similar set of in-game trade-offs could be leveraged by a gaming-focused version of a CTF where, for example, a fictitious bank heist could be conducted by teams of people and the different CTF challenges solved by each team would influence how easy or difficult the overall heist would be.

An example CTF "map" might feature a fictitious bank vault being burglarized by the CTF players and each action inside of the CTF corresponds to an action in the bank (perhaps visualized as a board game or an actual 3d game world). One challenge may be to get the access code to the bank vault from an employee by either phishing, breaking into the employee's smart lock protected office, or hacking the bank's online infrastructure. Each method would have a different difficulty and impart different game changing elements on top of the access code itself. Elements like this would allow CTF teams to choose their path and make active decisions in the game world.

In order to go from a hobby to a game and eventually an e-sport, CTFs need to shed parts of its origins in vulnerability discovery and exploitation and focus more on a simple game structure such that meaningful play can be ascertained by those in the midst of learning information security techniques. Without a metagame, different organizers will create different CTFs, which the general public will have difficulty grasping without an existing, firm command of information security concepts.

2. CTFd, an open source Capture The Flag framework

One of the difficulties in organizing a CTF is providing a scoring system and interface for the challenges [5]. Released in 2015 after the conclusion of CSAW CTF[1], CTFd is an open source framework for administering a CTF with just a web browser. A standardized interface is one of the first steps to breaking down the barriers in running a CTF.

CTFd was created to address the following issues in CSAW CTF and allow others to build a CSAW CTF-like environment for themselves:

- All database queries were written without an ORM or done manually
- Password resets were done manually and account confirmations were never implemented
- Desire to bring more visualizations into the CTF
- There were issues with vulnerable game infrastructure with both CSAW CTF and other CTFs [6]

In addition to the aforementioned issues, the CSAW CTF organizers wished to leverage more CTFs in our weekly educational workshop, Hack Night. Because of the difficulty in reusing the CSAW CTF software it was cumbersome to run a CTF every week.

Written in Python and Flask, CTFd can be setup very quickly, extended very easily, and can be modified to support many competition constructs used in CTFs today. For example, a CTFd reCAPTCHA plugin released by TAMUctf[2] was written in under 100 lines of Python code without leveraging any existing reCAPTCHA plugins or libraries. CTFd is both open source and built with customization as a goal.

CTFd's programming language choice and design is very easily adopted by newcomers compared to other CTF platforms, as shown by user testimony and increased public usage (see Figure 1).

> *... Alongside Bootstrap it uses Flask, so most of the programming is Python with dashes of HTML here and there. Flask is light and really easy. We already had one team member who knew Flask, but we all knew Python. As one of the team members who didn't know Flask, I can say that Flask is leaps and bounds easier to learn and understand than Facebook's PHP template system.*

Figure 1: User Testimony from Casey Erdman, Columbus State University, Black Box Society.

CTFd has been used by CSAW for its Capture The Flag and High School Forensics events since 2014. CTFd has also been used for Flare-On 3 (FireEye), the Car Hacking Village CTF (BugCrowd) at DEFCON 24, Hack The Vote (RPISEC), Boston Key Party 2017 (a DEFCON qualifier), RC3 CTF 2016 (RIT RC3), the

[1] https://github.com/CTFd/CTFd

[2] https://github.com/tamuctf/ctfd-recaptcha-plugin

IoT Village at DEFCON 24, and other events either hosted for the public or internally for companies and schools [6].

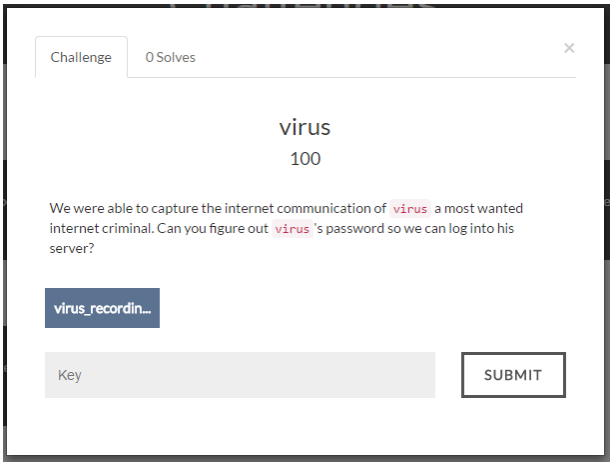

Figure 2: Example of a standard challenge in CTFd.

The interface to CTF challenges is an integral aspect of the CTF [5] and within this lays an area for improvement. For example, CTFs expect teams to leverage their own toolset on their own systems. In order to achieve the aforementioned metagame it is practically required that a singular set of tools be defined as a CTF toolset. A video game like Counter Strike does not expect players to provide their own weapons or ammo, similarly CTFs should aim to provide tools for users as part of the game itself.

While CTFd is not yet part of an overall e-sport meta, its interface (see Figure 2) allows integration and customization into other platforms and games. CTFd is the beginning of a CTF ecosystem that can be adopted immediately and easily for security education, automated assessment, and perhaps a truly gamified CTF.

2.1. The Scoring Logic within CTFd

Central to CTFd is its scoring logic. The scoring system factors in the amount of points each challenge was valued at, the speed of a team's solution relative to the start of the competition, and any additional points that a CTF organizer awarded to a team. Organizers have the ability to award points to competitors for exemplary behavior or detract points for undesirable behavior.

In addition, CTFd supports purchasable hints which allow a competitor to unlock additional information about a challenge if they wish to exchange their hard-earned points for a leg up in a yet unsolved problem.

This scoring logic is derived from the logic used in CSAW CTF prior to the development of CTFd. Hints and awards were added after receiving feedback from the community after open sourcing the project.

CTFd visually represents these fluctuations in scoring with a graph representing score over time for each individual team. The top ten teams have their score graph featured on the scoreboard page (see Figure 3).

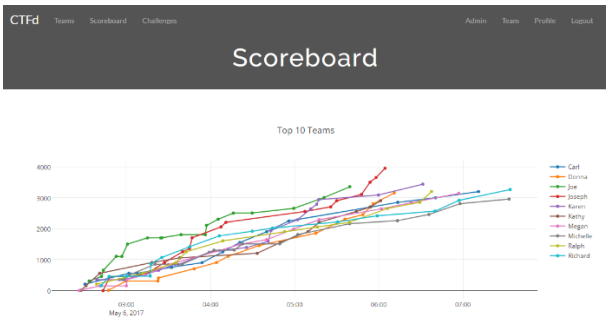

Figure 3: Score over time graph for the top ten teams.

The visualizations presented by CTFd can be leveraged to generate reports, provide dashboards, or quickly understand the state of a team. In addition, useful visualizations provide spectators with insight into the state of the game without needing to be actively involved.

For CTF organizers interested in leveraging CTFTime's scoreboard feed feature[3], CTFd provides all scores as a JSON feed written according to the CTFTime minimal feed format. This allows organizers to simply download a JSON blob emitted by CTFd and subsequently upload it to CTFTime for tracking.

2.2. The CTFd administration panel

CTFd was developed to replace the ad hoc solutions used for previous CSAW CTFs. Back then, various administrative tasks were accomplished by running raw SQL queries. Over time, CTFd expanded to support the many features expected of consumer facing web applications such as forgot password emails, email confirmations, and brute force protection. Before the release of CTFd, very few CTFs supported many features besides the common challenge interface. In addition, some of these CTFs, including CSAW CTF, had their scoring systems exploited due to vulnerabilities in the scoring logic [7].

A large portion of CTFd's feature set is in its administration panel, which provides a simple interface for organizers to create and modify competition data

[3] https://ctftime.org/json-scoreboard-feed

(see Figure 4). CTFd's administration panel prevents the accidental modification of said data and allows organizers to create challenge descriptions, point values and flags very quickly.

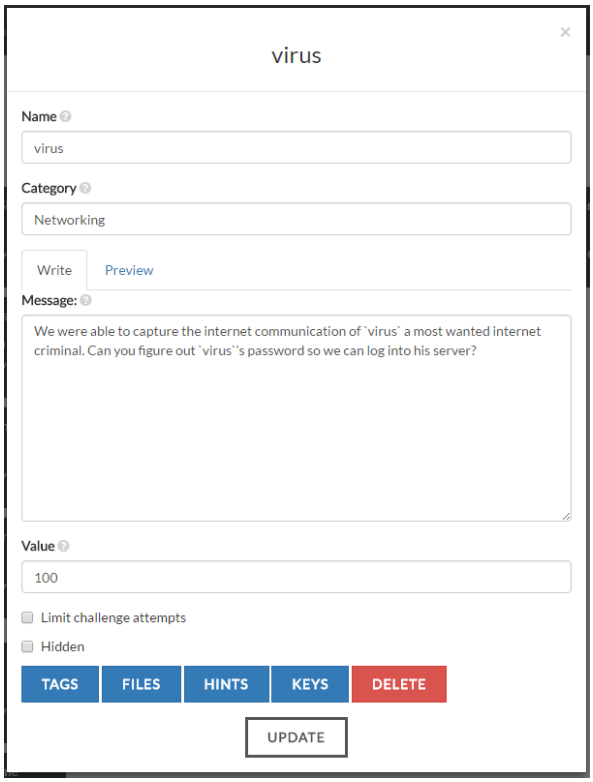

Figure 4: Challenge editing modal from which all aspects of a challenge can be modified.

The administration panel was designed with extensibility in mind. As an example, challenges can be tagged with some property by which a plugin can identify and modify behavior accordingly (see Figure 5). Examples include placing challenges in a different location of a map based on the tag[4] or only allowing a challenge to be solved with a certain programming language specified in a tag.

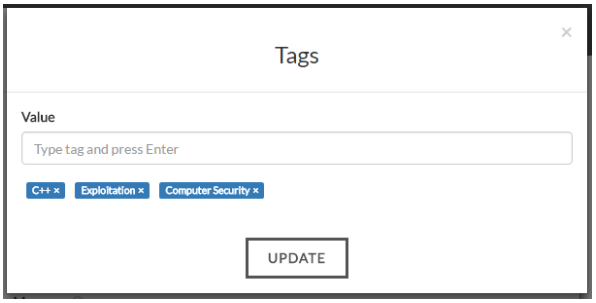

Figure 5: Interface for tagging a challenge

[4] https://github.com/ColdHeat/UnitedStates

The administration interface is also capable of modifying team data, calculating basic statistics, and acting as an HTML CMS. Organizers can also configure settings about their CTF like

- its start time and end time,
- whether scores are hidden,
- public/private registration,
- public/private scoreboard,
- or allowing team name changes.

2.3 Plugins and Themes

CTFd features both a plugin and theme interface allowing organizers to customize their instance of CTFd using additional Python and HTML code without needing to modify the underlying CTFd codebase. While source code modifications are quick and easy to make, they can become difficult to maintain over time as CTFd is updated. A plugin allows the core CTFd codebase to remain updateable to newer releases but retain customized behavior. Not using a plugin can cause a deployed CTFd release to become out of date or lose custom modifications upon update. A similar problem can be seen by Wordpress blogs where modifications to some files can be lost after an update[5].

CTFd's plugin interface allows organizers to create entirely new types of challenges via custom challenge and flag types.

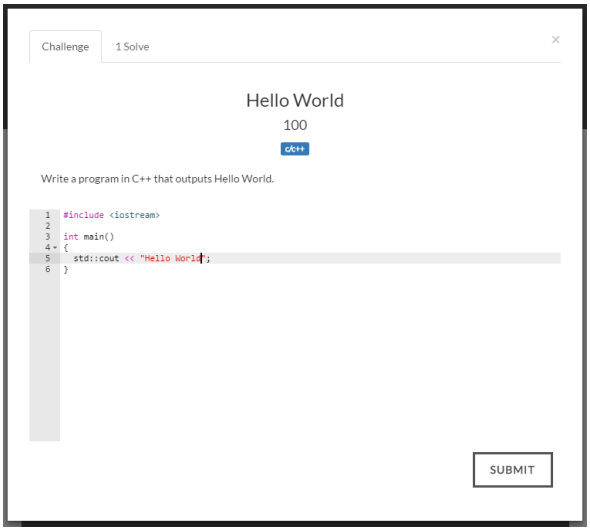

Figure 6: Custom programming challenge allowing a user to submit C++ code to CTFd.

For example, most CTFs only support a single static flag as the solution to a challenge. In contrast, CTFd

[5] http://www.wpbeginner.com/wp-themes/how-to-update-a-wordpress-theme-without-losing-customization/

allows the organizer to write custom code to define a format for the flag. Instead of a single flag, a user could submit code, entire files, or answer a multiple-choice question as their solution (see Figure 6). By providing different challenge types, CTFd can support challenges that wouldn't typically be seen in a traditional CTF. This idea is currently being used to support non-traditional CTF settings such as classrooms and workshops where offensive security is not the main teaching point.

2.4. Team Profiles

CTFd provides the full state of the CTF to all competitors and the public during the competition. During the CTF, a team can look at another team's solved challenges, list of awards, unlocked hints, and other team data (see Figure 7). This allows one team to take similar paths to another team in an effort to catch up or otherwise gauge their abilities in relation to another team's.

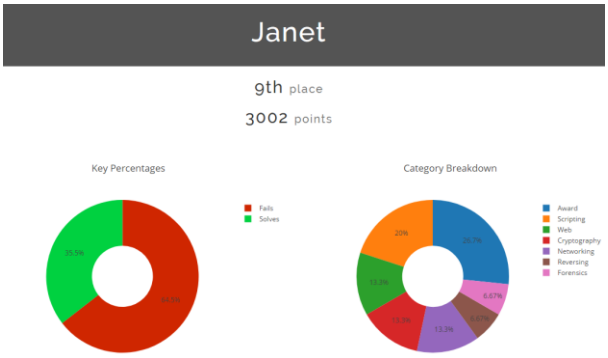

Figure 7: Team profile showing team specific visualizations.

CTFd allows for organizers to hide this information but the default behavior is to display this information so that teams can extrapolate data points from other team's behavior.

3. Comparison to other CTF platforms

Other released CTF platforms include FBCTF, OpenCTF, picoCTF, TinyCTF, Mellivora, and the iCTF framework.

CTFd differs from the aforementioned in its combination of existing features and extensibility. Every released CTF platform is written to achieve the simple challenge presentation and flag checking, but CTFd is the only platform written in Python with customization as a goal and has been tested over incrementing years of CSAW CTF.

FBCTF features a user interface where each challenge represents a country of the world. CTFd by default does not do this, but with a new theme the same functionality can be achieved (see Figure 8). Beyond its UI, FBCTF is written in Hack, Facebook's PHP dialect, which many developers are unfamiliar with. Hack can be difficult for many to use since Python and JavaScript are commonly taught as introductory languages and many developers are familiar with both.

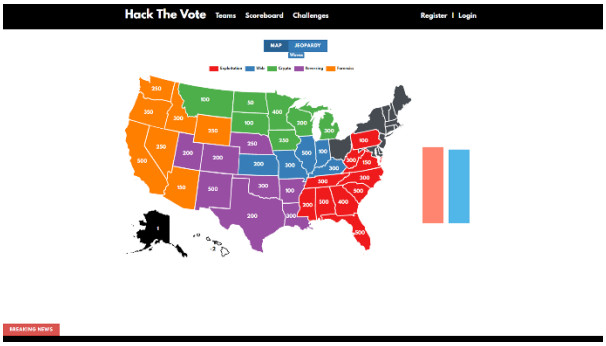

Figure 8: Hack The Vote from RPISEC implementing a CTFd Challenge UI mimicking a map of the United States.

picoCTF is well written and contains many parallels to CTFd but many aspects of the picoCTF platform require direct modification of the database or filesystem. CTFd takes a different approach and all customization is done through the web accessible administration panel. To picoCTF's credit it features an "autogen" challenge feature, which provides unique challenges to each team/user. CTFd does not support this feature but it has been implemented by the community through a plugin[6].

Mellivora is written in PHP, which typically has a negative connotation and is also not as popular a language as Python. CTFd integrates a Markdown CMS in contrast to Mellivora's news posts which are BBCode-based and do not create new routes.

The iCTF Framework is an aforementioned platforms as it focuses entirely on Attack and Defense CTFs. The framework itself is battle tested and has been used in the iCTF competition but a core aspect of CTFd is its incredibly low setup time, to which the iCTF framework does not cater, perhaps intentionally.

OpenCTF at time of writing is described as "not production-ready yet" and TinyCTF is currently vulnerable to CSRF. Neither has a web accessible admin interface allowing for the addition of challenges.

[6] https://github.com/tamuctf/ctfd-instancing-plugin

4. CTFs and their value to education and recruitment

CTFs present problems in a non-committal way that rewards dedication and allows competitors to gauge their abilities against others. Players are rarely forced to join a CTF but gamified aspects, like the competition scoreboard, can keep them engaged and learning [8]. Students can solve problems in their own time with solutions that they determine on their own. This promotes a self-reliance that can be difficult to impart in a more traditional educational setting that relies on teachers and textbooks.

In information security specifically, CTF challenges can represent real world problems that a security engineer or security consultant might encounter on a day-to-day basis [8]. The open-ended nature of CTFs very often rewards individuals who are dedicated to a problem and thus CTFs can be used to identify both very dedicated and very technical individuals [8]. Due to these properties, a company looking to recruit security engineers can leverage a CTF as a means of filtering out resumes or candidates. Whereas a resume provides a high-level view into a candidate's history, their abilities to complete security challenges in a CTF automatically provides a baseline skill assessment for the recruiting company.

4.1. The Future of CTF

CTFd is a first step into bringing CTF administration to a wider audience. As more CTFs are run, it is assumed that players who are at exposed to CTFs will increase and CTFs will find increased usage outside of the security industry. CTFd's feature set caters to the security-oriented nature of CTF but is flexible enough to expand into education and recruitment (see Figure 9).

> *What is impressive about the platform is its extensibility. Not only can I host challenge based technical questions, but I can use CTFd for quizzes/trivia/code-review and anything else really. We have even designed CTFd challenges for non-security people to help train in security awareness or teach them about internet privacy.*

Figure 9: User Testimony from Jason Haddix, Head of Trust and Security, Bugcrowd

As CTF evolves into more environments, more tooling will emerge to further lower the barriers to entry to the security field. Nonetheless while the history of CTF is rooted in technical competition and education, CTFs have diverse applications due to their competitive nature and simple schema. Despite the far-reaching capabilities, the overall lack of structure inhibits the expansion of CTF in an otherwise exploding technology and e-sports industry. CTF is still currently only enjoyed by a niche group of hardcore individuals despite best efforts at changing that. While the evolution of CTFs is unclear, a defined structure could be the first step towards achieving the simplicity and approachability that would catapult CTFs into the mainstream.

4. References

[1] T. Nighswander, "CTF Meta-Game," [Online]. Available: http://copyfighter.org/ais3/slides/ctf_day3.pdf.

[2] "CTFTime 2016 CTF Events," [Online]. Available: https://ctftime.org/event/list/2016.

[3] V. Genovese, "Capture the Flag: An Owner's Manual," [Online]. Available: https://www.usenix.org/sites/default/files/conference/protected-files/enigma16_slides_genovese.pdf.

[4] CTFTime, "CTF? WTF?," [Online]. Available: https://ctftime.org/ctf-wtf/.

[5] fuzyll and psifertex, "The Many Maxims of Maximally Effective CTFs," 17 12 2015. [Online]. Available: http://captf.com/maxims.html.

[6] K. Chung, "https://medium.com/ctfd/recruiting-and-teaching-with-capture-the-flags-39115b071c2," CTFd LLC, 6th March 2017. [Online].

[7] Eindbazen, "Eindbazen - CSAW 2012 – Web 600," 30th September 2012. [Online]. Available: https://eindbazen.net/2012/09/csaw-2012-web-600/.

[8] T. Nighswander, "Building a Competetive Hacking Team," [Online]. Available: https://www.usenix.org/sites/default/files/conference/protected-files/enigma_slides_nighswander.pdf.