CTF: State-of-the-Art and Building the Next Generation

Clark Taylor[1,3], Pablo Arias[2,3], Jim Klopchic[3], Celeste Matarazzo[3], and Evi Dube[3]

[1]Department of Computer Science, University of Arizona
[2]Department of Computer Science, North Carolina A&T State University
[3]Lawrence Livermore National Laboratory

Abstract

Capture the flag (CTF) style events have become increasingly popular events for recruitment, training, evaluation, and recreation in the field of computer security. Today, there exists a vast array of CTF software; this software may be divided generally into game engines and challenge components. Game engines, which determine the overall style of the competition, can be categorized into those which support dynamic challenges and those which support static challenges. A small number of game engines are open and available for any party to develop their own challenges on, though most are proprietary solutions.

Over the course of the last 8 years, the Cyber Defenders group at Lawrence Livermore National Laboratory hosted an annual CTF event for its interns, in the process evaluating different CTF types and engines and ultimately developing data on the state-of-the-art in this field. While these events resulted in a large degree of success with regard to the goals mentioned above, a critical evaluation of the software both used by the Cyber Defenders and generally across the entire field revealed several shortcomings of current CTF practices. In particular, current software may be improved with regard to challenge realism, costs and accessibility, educational applications, and research potential. Proposed herein is a new game engine which addresses these shortcomings. This paper details the architectures for and current progress towards implementing this game engine.

1 Introduction

Modern research in computer science education emphasizes the need for collaborative and engaging modes of instruction providing hands-on experiences to students and trainees [58]. One way to do this effectively is through gamification, wherein implementing aspects of education using methods common to games has been shown to generally result in desirable outcomes [26]. In computer science, gamification is often implemented in the form of competitions; typically, such competitions are dedicated to a sub-discipline such as algorithms, robotics, or, as discussed herein, computer security [51].

Cybersecurity competitions—often labeled, sometimes erroneously,[1] capture-the-flag competitions—provide much needed education and training in a field dominated by a general lack of human resources and, therefore, available expertise [53]. These competitions create a more competent, sizable, and diverse cyber security workforce by providing valuable training, experience, education, and recruitment for current and future security experts [40]. Specifically, the competitions place those current and future cybersecurity professionals into a adversarial, competitive, and ideally realistic environment wherein they use the tools and methodologies generally encountered in the field to deal with artificial situations. This, in turn, (1) attempts to hone skills which often prove useful in mitigating and responding to real threats and (2) has the potential to encourage students/trainees to pursue the subject matter academically and professionally by simply being fun, exciting, and concretely do-able.

1.1 Background and Motivation

Over the course of the last 8 years, the Cyber Defenders program[2] at Lawrence Livermore National Laboratory conducted a 3-day computer security competition annually in order to train and expose stu-

[1] Capture-the-flag is a specific type of competition, wherein participants attempt to find "flags" in various competition components. Not all competitions follow this pattern; despite this, they are often labeled "capture the flag" events, likely due to the dominance of the CTF mode of competition.

[2] The Cyber Defenders program offers summer internships for students interested in computer security. The interns range from high school through graduate studies, all of whom either participate in or build challenges for and administer the CTFs.

dent interns to the field. Almost all of the content and all of the game engines for these competitions were generated by third parties, including the TracerFire CTF by Sandia National Laboratories and the Dirtbags Competition by Los Alamos National Laboratory, with supplemental challenges prepared in-house [32, 34]. For the last 3 years, this same group also participated in a summer-long security competition, with challenges developed completely in-house and game infrastructure initially developed in-house before migrating to the PicoCTF game engine [16]. Most recently, the group administered similar challenges at the Bay Area Maker Faire [36] and at the Department of Energy National Science Bowl [50], with competitors from younger audiences. While we consider all of these competitions successful inasmuch as they accomplished the stated goals, they also revealed several shortcomings in security competition game engines.

1.2 Overview

In this paper, we contribute a current survey and evaluation of the state-of-the-art for computer security competition game infrastructure software given the experiences of the Cyber Defenders group. Ultimately, we conclude that the CTF frameworks may be vastly improved over the current state-of-the-art to meet educational and research goals and we propose a new CTF framework—emphasizing framework and content availability and extensibility as well as data collection for research purposes—to make those goals a reality. Our implementation approach to these implements is occurring in phases; the current status of our implementation is detailed as well.

2 Current State of the Art

In examining the current state-of-the-art, we combined our experiences derived from years of conducting CTF events and developing CTF software with a survey of 36 contemporary CTF competitions. These competitions were selected through informal surveying of LLNL employees and students at the University of Arizona who compete (or have competed) in external CTF competitions as well as through academic literature review;[3] all CTFs mentioned were evaluated by the authors of this paper (who viewed competitions' websites, related academic literature, discussed competitions with employees/students who had participated in them, and attempted to run and build challenges for competitions with open source frameworks) in order to determine competition type and format as well as whether the competition featured policy-based content and was open source. For competitions with open-source software, the authors also evaluated the requirements for using the software. The results of our analysis are summarized in Table 2. These results first and foremost demonstrate the pervasiveness of CTF competitions through their sheer number—and, though we examined numerous CTF events, countless more exist as well. Secondly, these results reinforce the ideas presented in Section 1—namely, that the goals of these competitions vary but tend to focus on educational experience. Similarly, the format of the competitions vary from in-person team competition to remote (via the Internet) competitions with individual participants. Beyond this, our study of the current state-of-the-art focused on aspects of competition that, over the years, the Cyber Defenders group found limiting or otherwise problematic. To this end, the research revealed several trends in computer security competitions: (1) CTF software implementations are designed to support either static or dynamic challenge modes; (2) current CTF competitions require a lot of expertise and resources to administer, and CTF content generation tends to be limited by framework extensibility; (3) despite a focus on education and training, current CTFs are highly focused on competition; and (4) researchers are attempting to leverage CTF events to study computer security, both in education/training efficacy and beyond.

2.1 Challenge Mode

Generally, competition infrastructure supports either static or dynamic challenge types. The former modality is defined by teams solving challenges in isolation with files or systems that are "offline" in the sense that there exists no real-time changes to them. Points are gained by completing particular tasks with these files and systems.[4] The latter, by contrast, involves active systems which competitors attack and defend in real-time in order to gain points; these points are awarded for successfully carrying out these attacks and defenses.[5] No competition studied offered both

[3] This list, though sizable, is not exhaustive and the authors plan to maintain a similar list on the Catalyst website, revised to include other competitions as they become aware thereof. In particular, thus far the examined CTFs tend to be English language; the authors were unable, using their methodology, to examine CTFs which did not have English-language documentation.

[4] An illustrative example consists of finding an encrypted flag in a network traffic dump.

[5] For instance, a team may be awarded points for gaining access to a SSH service on a vulnerable machine owned by a

Competition Name	Static	Dynamic	Policy	Open Source	Requirements
Defcon CTF Finals [19]	No	Yes	No	No	
RuCTF [17]	No	Yes	No	No	
UCSB iCTF [52]	No	Yes	No	Yes	FVNCPIE5
RuCTFE [44]	No	Yes	No	No	
Defcon CTF Qualifiers	Yes	No	No	No	
OpenCTF [57]	Yes	No	No	No	
CCDC [13]	No	Yes	Yes	No	
Panoply [56]	No	Yes	No	No	
TracerFire [32]	Yes	No	No	No	
WhiteHat Wargame [55]	Yes	No	No	No	
PoliCTF [48]	Yes	No	No	No	
14th HUST Hacking Festival [27]	Yes	No	No	No	
ASIS CTF Quals [7]	Yes	No	No	No	
School CTF [45]	Yes	No	No	No	
Volga CTF Quals [41]	Yes	No	No	No	
Teaser CONFidence CTF [3]	Yes	No	No	No	
PlaidCTF [38]	Yes	No	No	No	
Hack Zone Tunisia [21]	Yes	No	No	No	
Nuit du Hack CTF Quals [35]	Yes	No	No	No	
UIUCTF [46]	Yes	No	No	No	
BackdoorCTF [2]	Yes	No	No	No	
0CTF Quals [1]	Yes	No	No	No	
BCTF [11]	Yes	No	No	No	
Securinets CTF Quals [43]	Yes	No	No	No	
B-Sides Vancouver [31]	Yes	No	No	No	
Codgegate CTF Preliminary [18]	Yes	No	No	No	
InCTF Quals [6]	Yes	No	No	No	
Boston Key Party CTF [12]	Yes	No	No	No	
Break In [29]	Yes	No	No	No	
WCTF [5]	Yes	No	No	No	
Ghost in the Shellcode [4]	Yes	No	No	No	
Insomni'hack teaser [28]	Yes	No	No	No	
OWASP Security Shepherd [22]	Yes	No	No	Yes	FVCPDWSE4
PicoCTF [16]	Yes	No	No	Yes	FNCDWE3
HackIM [23]	Yes	No	No	No	
Atlantic Council Cyber 9/12 [8]	No	No	Yes	No	

Table 1: Listed are 36 CTF implementations with information regarding whether the given engine supports static or dynamic content, whether the content supported contains policy-based problems, and whether the challenge engine and content is open-source. The requirements column contains (for available open-source entries) several alphabetical characters which indicate requirements; these include: F—filesystem configuration, including knowledge of framework file structures and formats; V—virtual machine or docker use; N—networking, such as DHCP configuration; C—command line execution without GUI support; D—native database access; P—high performance or additional hardware for scalability; W—webapp knowledge, such as installing and managing a web server; I—instructions/documentation lacking in clarity; S—single script setup available; and E#—extensible framework which allows new challenge creation and requires some work to place content into the framework software, where the digit indicates difficulty of new challenge creation on a scale of 1-5. Collectively, the Requirements column details who might be able to host the given CTF framework, whether it is possible to and the difficulty of integrating new challenges, and at what cost for both.

different team.

static and dynamic challenges.

2.1.1 Shortcomings

While there exist examples of highly successful CTF events which use both of these challenge types, neither completely aligns with realistic computer security scenarios. Outside of competition, computer security professionals can expect to encounter a huge variety of real-world challenges, which encompass both the static and dynamic challenges presented here. For instance, a system manager would likely have to know how to look through network and extract some semantic meaning (such as attribution or compromised component identification) for reports, aligning with the experiences provided by static challenges, but that individual would likely also have to know how to fix security vulnerabilities in the system, aligning with the dynamic challenge experiences. Simply offering only one mode limits the applicability of the competition experience to the real world.

2.1.2 Policy

The detrimental effect the static-dynamic limit has another side effect, compounding the negative effect on realism in the competition experience. Policy challenges involve questions of legal permissiveness and policy efficacy from a variety of viewpoints;[6] security professionals often come into contact with difficult policy questions—such as the legal ramifications of various "hackback" methods [33]—and are called upon to deal with these types of problems in reality. However, policy challenges by definition require a fairly substantial scope in order to consider the realistic effects of those challenges. Because allowing only static or dynamic challenges limits the scope of the competition by omitting particular aspects of real-world problems, policy questions can only exist in much more limited forms in current competition formats.[7]

2.2 CTF Costs and Extensibility

One significant barrier to leveraging CTF competitions in education, training, and other financial or expertise-constrained applications is the cost associated with hosting CTF competitions. These costs generally involve (1) hardware costs for hosting the competition, (2) the human resource expense required to administer the competition, and (3) the availability of and/or investment associated with competition material for the particular event. Among each of these categories, there exist both recurring and nonrecurring (stagnant) costs. How the costs are structured depends on the goals and desires of the CTF hosts, as well as strategies in dealing with these categories. Hardware costs may be viewed as recurring or stagnant, depending on hardware upgrade cycles and the use of rental or cloud infrastructure; administrative costs tend to be recurring for each CTF event; content development may be a recurring cost if the host desires new content on a regular basis. In creating a CTF framework, keeping cost requirements low allows the framework to be employed in circumstances where it would otherwise not be possible to administer a CTF. Herein, the cost aspects of CTFs are discussed categorically, in order to identify ways to minimize costs. Further empirical research into specific costs of hosting various CTFs presents future work.

2.2.1 Hardware Costs

Generally, potential CTF competitions must consider two main hardware costs: (1) the hardware required to host the CTF, and (2) the client terminals that competitors will use to compete. With regard to the latter, strategies such as using low-cost Raspberry Pis [37] with freeware Kali Linux operating systems [39] as terminals has allowed the Cyber Defenders group to administer CTFs with low terminal hardware costs. Having competitors bring their own terminals (which is feasible but less desirable in educational settings due to environment setup and uniformity problems) effectively eliminates this cost in some settings. In short, these costs are highly influenced by the CTF format; the game software does little to influence terminal costs except inasmuch as certain challenges may require specialized hardware to solve.

The former cost involving CTF hosting hardware, by contrast, is highly influenced by the CTF software. As suggested by related work, dynamic challenges often require significant hardware resources, while static challenges do not [42]. Our findings with regard to static challenges agree: The Cyber

[6] The Cyber Defenders CTF events attempted to introduce policy questions relating to company security policy and law enforcement perspectives.

[7] Only a single competition [8] identified by the authors supports policy challenges currently; that competition omits all technical aspects of a CTF and replaces them with scripted scenarios. The limited scope of this competition drastically reduces realism in the scenario with regard to computer security subject matter, though it may be an adequate foil to high-level policy making. Regardless, the methods employed in this competition offer a guide to policy-content integration in technical CTF events.

Defender group, in administering its PicoCTF-based static challenge CTF, utilized a single Raspberry Pi as the server hardware without any noticeable performance degradation for a competition involving 50 competition terminals. Because significant server resources were not required, costs were kept to a minimum.

Moving towards a more complicated CTF, however, it is likely that lower server performance could become problematic. In dynamic CTFs, complicated challenges which may involve various types of services introduce complexity; these types of challenges may have resources running on the main CTF server, on separate piece-mealed hardware, or virtualized on either.[8] The way in which these challenges interact with the server also effects the main CTF server hardware requirement; some challenges may place significant load on the main CTF server by, for instance, conducting near-real-time service polling. Keeping the framework's main competition server lightweight will allow it to be flexible; resource intensive challenges may be deployed in events with greater hardware resources, while more constrained events may employ less resource-heavy challenges.

Another factor in CTF hosting hardware is the network backbone required to connect the server to the competition terminals. The particular requirements for this will vary from CTF to CTF, according to the format of the event. Some competitions, for instance, connect to competitors in various geographic locations over the Internet. The Cyber Defender CTF events have, by contrast, been hosted in closed, controlled local area networks over hardware ethernet connections. This cost can grow dramatically with competition size, with additional competitors requiring additional network switches and cabling. Some of this cost, however, may be mitigated by employing wireless networking, which is not limited by network cables or the number of ports on network switches. Generally, the network hardware cost is not drastically effected by CTF software; in dynamic challenges, some consideration must be given to placing resources on the network, as more servers and network segmentation (the amount of which depends on the particular challenge) will generally incur additional costs unless it is possible to virtualize them though it is likely that the additional server hardware costs will outweigh network hardware costs by a significant margin, as the marginal cost of connecting to a network tends to be much lower than the cost of new server hardware—even if that server hardware is low cost.

2.2.2 Competition Administration Costs

CTF competitions to date have generally been written with computer-savvy audiences in mind. With good reason, CTF creators assumed that those seeking to administer an event would generally know how to manage software components such as the CTF server, network components such as the domain name service, and hardware components as listed above in Section 2.2.1.[9] Making these assumptions, however, has two side effects. First, it increases the human resources required to administer a CTF by introducing additional responsibilities. To illustrate, the Cyber Defenders group recently hosted a CTF for the Department of Energy National Science Bowl, where we brought a large team including 3 individuals with CTF administration backgrounds, all of whom had large amounts of experience with the engine, the hardware, and computer networking. Educators, less experienced computer scientists (such as undergraduates or high school students), and other groups who may want to try hosting a CTF but do not have experience in network and web applications may not have the expertise to conduct the functions required to host a CTF. In order for these groups to host a CTF, they would need to find and recruit human resources to aid them, increasing costs and deterring CTF hosting; still other groups may be deterred by the cost in human resources, even if such resources are available, when compared to the expected value of hosting a CTF. Though little can be done in software to mitigate hardware setup requirements, designing software for with human resource costs in mind can decrease such costs, making CTF hosting easier and more cost effective.

2.2.3 Content Creation and Availability

As mentioned, CTF events require two main software components—the competition infrastructure and challenge content. As shown in Table 2, the research here identified three CTF implementations that are open source; those seeking to build a CTF could feasibly employ one of these frameworks, as the Cyber Defender team did with PicoCTF. However, those seeking to host a CTF often do not have access to the latter. Even if a CTF organizer was able to use an open source competition frameworks (see Section 2.2.2 for discussion of human resource difficulty), there does not exist significant amounts of

[8] Virtualized here refers to either traditional virtualization or, as suggested in related work, more efficient forms such as Docker containers [42].

[9] Additional detail regarding specific knowledge-based setup requirements is located in Table 2's "Requirements" column.

content available for an event. Creating content requires huge investment; the PicoCTF development team, for their flagship event, consists of 21 individuals [14]. While this level of investment is permissible for large events,[10] more informal and smaller events as may be found in educational or training settings often cannot afford such investment. Challenge content sharing and reuse can, potentially, fill this gap. Currently, however, this is only done in limited form, with CTF framework developers offering example challenges to build from. Moreover, challenge sharing introduces other unresolved problems—how does one deal with previously published solutions; how does one alter the content to fit new narratives or audiences; how easy is it to import and setup challenges; how can challenge creators most efficiently share their content? Current content sharing paradigms only offer static code for reuse, which requires significant time and effort (and expertise) for modification and integration into a CTF event.

In addition to general content-creation costs, CTF organizers may incur additional costs if they wish to provide content which does not fit into current frameworks. In addition to limited support for pre-build content, current framework design tends to highly enforce particular software design formats, to the detriment of extensibility. ICTF, for example, requires challenges to plug into its Debian makefile combined with Python configuration and scoring format [49]. While some extensibility exists for ICTF in that the services can be in any desired format, the framework itself requires the challenge content service to run on the main CTF server and be scored in a single, standard way. Moreover, the service is limited to a single instance, in what might be described as a "king-of-the-hill" instance. Attempting to build challenges outside of this format—such as hosting individual services for each team, or connecting physical services to the network for scoring—requires significant integration coding.

In addition to all of this, current documentation of CTF frameworks for content creation varies greatly. Combined with the constrained and proprietary formats for CTF framework integration, a CTF organizer may face significant costs in developing content for an event.

2.3 Modes of Gameplay

Part of what makes CTF events effective in pedagogical settings derives from the competitive nature introduced by gamification. However, the Cyber Defender group's experience reflects that there exists a limit to the efficacy of competition. When competitors do not have sufficient background to compete, as is the case when introducing individuals to completely new concepts and topics, competition can have a negative effect on educational outcomes [47]. This concept became apparent in several of the 3-day intensive CTFs administered by the Cyber Defenders group, wherein the organizers observed teams and individuals withdrawing from the competition due to poor performance relative to others, in the process halting the educational elements provided by the event for those individuals. While prior work enumerates factors which lead to this type of educational stalling, it is not always possible to ensure these ideal conditions when hosting a CTF with the goal of teaching unfamiliar concepts to competitors. It may, however, be possible to remove some gamification from the CTF environment when teaching difficult concepts, and thus leverage the CTF environment for education without alienating poor performers. None of the current competitions examined offer any such mode.

2.4 CTF Based Research

Currently, it is common practice to collect some data from CTF events for research purposes. Illustrative examples of this include extensive surveys with sophisticated psychological metrics analyzing whether CTF events are effective recruitment tools [9]; self-reported assessments in The MIT Lincoln Laboratory Capture-the-Flag Exercise [54]; focus-group interviews on teamwork in the ICTF event [30]; and situational awareness measurements in the CCDC competition generated using team scores, questioning of team members, examining network and individual device logs, and audiovisual data collected at the event [25]. Despite the apparent desire to leverage CTF events for research, very little work has been done beyond collecting informal (and often self-reported) data from the events. In previous years, the Cyber Defenders group has also collected such data, but desired to collect additional data in order to expand research potential. In particular, the group believes CTF events provide a potential test-bed for computer security research generally, with significant implications, in particular, to experiment repeatability and vulnerability detection as competitors utilize both known and unknown security vulnerabilities to solve problems in a realistic but sandboxed environment [10]. However, the state-of-the-art provides no methods for collecting data useful for this purpose.

[10] PicoCTF's 2013 event involved more than 10,000 competitors [15].

3 The Catalyst Framework

Given the current state-of-the-art presented here, the Cyber Defenders group began designing and implementing a new, highly extensible CTF framework in the summer of 2016, naming the project the Catalyst Security Challenge Framework. Addressing these problems, we focused on challenge extensibility to support any feasible challenge type, keeping costs low through architectural efficiency and framework ease of use, offering different modes of gameplay for educational use, and data collection methods for research. Despite significant amounts of progress implementing several of the proposed framework's components—currently, only the data collection component is mature—the architectural design is still in draft-form and will likely change before being finalized.

3.1 Framework Components

The proposed Catalyst framework contains several components, as shown in Figure 1: (1) the main competition server and database; (2) service hosts, which may be either physical or virtualized; (3) competitor terminals; (4) administrator terminals; and (5) network infrastructure.

All interaction between organizers and the framework may be done via a web-based UI, making the system easy to use for almost any level of user. Also included in the main server are typical network management components such as DHCP and DNS, which are configurable via the UI. Competition organizers use the web interface on the main competition server to select challenges from local files or remote code repositories. Before the competition begins, administrators register service hosts with the main server through either a command-line or web interface; the main server may be instructed to generate and register virtual machine hosts automatically. The administrators then assign challenges to registered hosts. After this configuration phase completes, the competition server connects to and runs the challenge configuration script on the assigned hosts in a provisioning stage. Depending on game configuration, competitors may also be required to register terminals, which would be provisioned with components such as data collection software and helpful tools.

3.2 Challenge Types and Extensibility

Our approach to challenge type involves offering an extensive and scoring API in order to support any type of challenge and supplementing that API with built-in support for several particular problem types that generally occur in static competition. Challenges are all presented on the game server's web page during competition for competitors to view; static challenge components are also shown on the main server web page.

3.2.1 Challenge Customization

In order to allow custom challenge configuration, each challenge component is also configured with variables supplied by the CTF organizer. In practice, this is done by having the organizer override default values supplied with a given challenge; these are passed to the challenge as a configuration file collocated with a given challenges' configuration script. The script can pass the configuration to other components as necessary; the end result is that content developers may add dynamic content which organizers can change to fit different competitions.[11]

3.2.2 Scoring API

In order to provide the maximum amount of extensibility, the proposed framework includes a web-based scoring API. In the provisioning phase, the main server executes challenge-specified commands to launch a scoring program. This scoring program connects to the scoring API on the localhost, where it can instruct the system to add points. The scoring program should probe competition services or be connected to by those services in order to grade them and issue appropriate scores.[12] Then, the scoring program connects to the Catalyst Scoring API via a JSON-formatted HTTP request on the localhost, instructing that API to award points [24, 20]. Because JSON and HTTP (and TCP) are well standardized and supported in almost every programming language, challenge developers are free to develop as they please, having only to conform to an incredibly simple web API.

3.2.3 Static Challenges

In addition to the scoring API, challenges may also utilize static components on the main competition web server. These components include the challenge description, specific questions with an answer form,[13]

[11] As an illustrative example, a vulnerable webapp may be configured with particular text components with fit a CTF storyline.

[12] As an example, a scoring program may probe a web service on a team's host determine it is functioning fine, and add points for that grade.

[13] The answer form may be configured to accept text, files, and other types of questions such as multiple-choice.

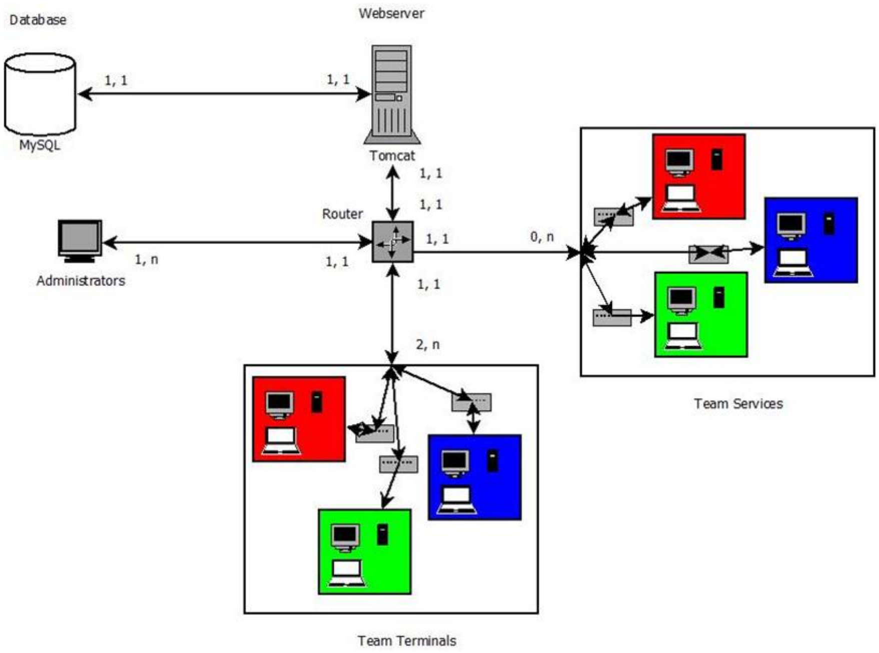
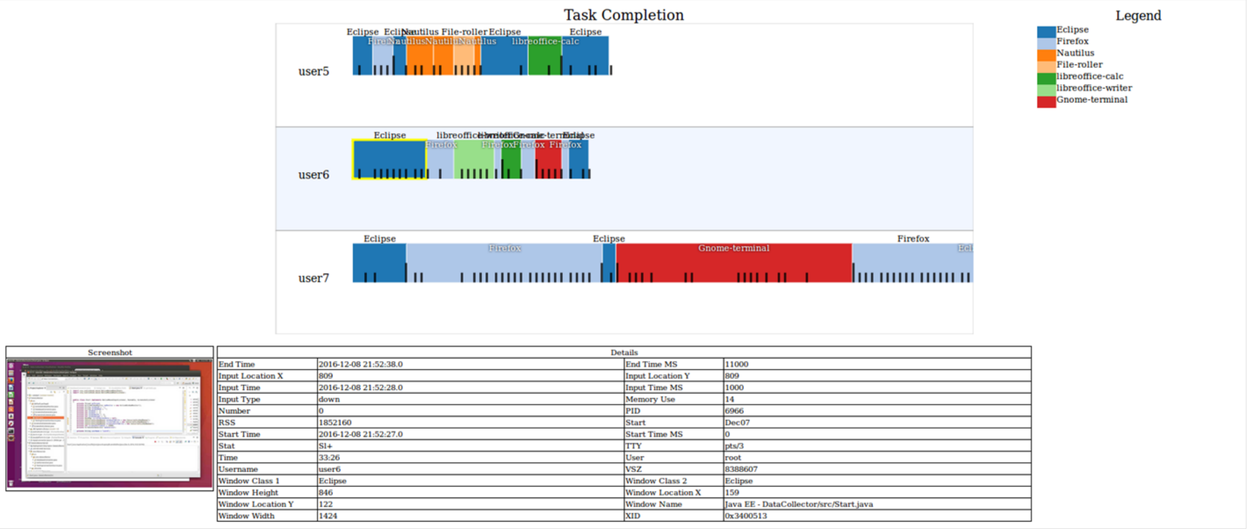


Figure 1: Catalyst primarily consists of a main server, various service servers, and terminals. The exact form of these pieces in actual hardware is not specified, as the software may be deployed in any number of different physical and virtual environments. Colors here indicate team affiliation for particular components.



Details			
End Time	2016-12-08 21:52:38.0	End Time MS	11000
Input Location X	809	Input Location Y	809
Input Time	2016-12-08 21:52:28.0	Input Time MS	1000
Input Type	down	Memory Use	14
Number	0	PID	6966
RSS	1852160	Start	Dec07
Start Time	2016-12-08 21:52:27.0	Start Time MS	0
Stat	Sl+	TTY	pts/3
Time	33:26	User	root
Username	user6	VSZ	8388607
Window Class 1	Eclipse	Window Class 2	Eclipse
Window Height	846	Window Location X	159
Window Location Y	122	Window Name	Java EE - DataCollector/src/Start.java
Window Width	1424	XID	0x3400513

Figure 2: This figure depicts the current progress on data collection visualization. The Gantt Chart features a timeline of in-focus programs for given users, along with event (challenge) start/completion indicators, a user input bar chart, a screenshot for the selected point on the timeline, and running process information.

a flag portal to submit flags that might be hidden elsewhere in the challenge, and hints which may apply to any of these or the service components. The specific question and answer questions can be graded via simple matching or via manual grading.[14] Natural language processing methods may eventually be added for automated grading of long answers.

[14] Manual grading may also take into consideration external activity such as presentations.

3.2.4 Competition Mode

When configuring a CTF, organizers configure the main Catalyst server with the desired difficulty level, according to educational need. This variable controls the verbosity of challenge components, showing additional hints for easier difficulties and removing all hints for more experienced—and competitive—levels. Additionally, the difficulty level will determine whether and how to present team and/or individual scores. A low difficulty level will disable the scoreboard altogether, mitigating problems associated with overly competitive environments.

3.3 Data Collection and Evaluation

In order to provide greater research potential and performance assessments, data collection components are also included in the Catalyst Framework. This component, which has already been implemented and is currently in testing, collects several pieces of data from each competitor terminal, including: (1) the current in-focus program; (2) associated process information, including file access; (3) background process information; (4) user input (keyboard and mouse); (5) periodic screenshots; and (6) challenge completion information. This data, curated on the game server, may be mined or visualized. Figure 2 shows the data visualized in a task-completion context with each user on a timeline showing activity from challenge beginning to challenge completion.

In all, these pieces of data capture summarizes what a given competitor does on their terminal device. Mining this could, for instance, demonstrate what tools or methods the competitor used to solve a given problem. If the given problem is regarded as a hard, research worthy problem, then the data could be used for that research. Ultimately, these competitions could be used as a form of crowdsourcing research on how to solve difficult problems. Other research, such as into teamwork factors, may benefit from more data inputs such as audio or video capture; integrating sources for this data is not yet supported but will be added in the future.

3.4 Policy Based Challenges

As mentioned in Section 2.1.2, however, policy content may be valuable in many circumstances in order to more accurately reflect computer security as a profession. Policy challenges are difficult to implement without a realistic scenario, as they become disjointed and unintelligible without strong concepts of what competition entities (participants, teams, administrators, software and hardware components, for example) represent with regard to the policies in play.[15] The Catalyst framework is designed with policy challenges in mind: Catalyst includes challenge configurability (discussed in Section 3.2.1) which allows CTF content creators to readily adapt scenarios to better fit policy challenges. Catalyst's combined-mode competition also fuels policy by allowing more realistic scenarios which parallel the situations for which policies were designed. Finally, the native support Catalyst contains for written answers and manual grading allows the proper forms to easily support policy challenges.

4 Conclusion

Gamification in computer security education generally leads to positive learning outcomes; the experiences of the Cyber Defender group over 8 years of CTF competition supports this notion. However, current CTF software frameworks can be improved in several ways in order to make CTFs more extensible to support novel challenges, easier and cost effective to implement and thus available to more groups, more flexible to be a valuable teaching tool in more contexts, and more valuable as a research tool through better data collection. The Catalyst Framework attempts to meet these ambitious goals, and several components are actively being tested at this time.

5 Acknowledgements

This work was prepared as a joint effort between Lawrence Livermore National Laboratory (LLNL), the University of Arizona, and North Carolina Agriculture and Technical State University. Work prepared under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 is released under LLNL Review and Release number LLNL-CONF-731180. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.

[15] To illustrate, one of the Cyber Defenders CTF events included a challenge wherein teams were to act as a law enforcement agency in order to provide training regarding chain-of-custody, attribution, and the Computer Fraud and Abuse Act. This challenge was difficult for students because the other challenges at best provided a subset of what such an agency would be doing, and participants became confused when that material did not align with the policy challenges.

References

[1] 0ctf 2017. https://ctf.0ops.net/.

[2] Backdoor - security platform. https://backdoor.sdslabs.co/.

[3] Confidence ds ctf. https://ctf.dragonsector.pl.

[4] Ghost in the shellcode. http://ghostintheshellcode.com/.

[5] WTCF. http://www.wirelessvillage.ninja/wctf.html.

[6] Amrita Center for Cyber Security Systems and Networks. Amrita inctf junior - india's first high school hacking competition7. http://portal.inctf.in/.

[7] ASIS.io. Asis ctf. https://asis-ctf.ir/home/.

[8] Atlantic Council. Cyber 9/12 student challenge. http://www.atlanticcouncil.org/programs/brent-scowcroft-center/cyber-statecraft/cyber-9-12.

[9] Bashir, M., Lambert, A., Wee, J. M. C., and Guo, B. An examination of the vocational and psychological characteristics of cybersecurity competition participants. In *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)* (2015), USENIX Association.

[10] Benzel, T., Braden, R., Kim, D., Neuman, C., Joseph, A., Sklower, K., Ostrenga, R., and Schwab, S. Experience with deter: a testbed for security research. In *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on* (2006), IEEE, pp. 10–pp.

[11] blue-lotus. http://bctf.xctf.org.cn/.

[12] Boston Key Party. Boston key party. http://bostonkeyparty.net/.

[13] Carlin, A., Manson, D., and Zhu, J. Developing the cyber defenders of tomorrow with regional collegiate cyber defense competitions (ccdc). *Information Systems Education Journal 8*, 14 (2010).

[14] Carnegie Mellon University. picoctf - cmu cyber security competition. https://picoctf.com/about.

[15] Chapman, P., and Brumley, D. picoctf: Teaching 10,000 high school students to hack, 2013.

[16] Chapman, P., Burket, J., and Brumley, D. Picoctf: A game-based computer security competition for high school students. In *3GSE* (2014).

[17] Childers, N., Boe, B., Cavallaro, L., Cavedon, L., Cova, M., Egele, M., and Vigna, G. Organizing large scale hacking competitions. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2010), Springer Berlin Heidelberg, pp. 132–152.

[18] Code Gate Security Forum. Codegate 2017. https://www.codegate.org/en/hacking/general.

[19] Cowan, C., Arnold, S., Beattie, S., Wright, C., and Viega, J. Defcon capture the flag: Defending vulnerable code from intense attack. In *DARPA Information Survivability Conference and Exposition, 2003. Proceedings* (2003), vol. 1, IEEE, pp. 120–129.

[20] Crockford, D. The application/json media type for javascript object notation (json).

[21] CSI. Hack zone v. http://www.hackzone.csi-ensi.tn.

[22] Danihan, M., and Duggan, S. Owasp security shepherd. https://www.owasp.org/index.php/OWASP_Security_Shepherd.

[23] Datta, A., Hatti, A., Ajith, Mahajan, A., Shrivastava, A., Bhargava, A., Shah, A., Machiry, A., Jakhar, A., Das, H., Forshaw, J., Bharmal, M., Mahajan, P., Walikar, R., Saint, Mittal, S., and Chauhan, S. Nullcon hackim 2017. http://ctf.nullcon.net/.

[24] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. Hypertext transfer protocol–http/1.1. Tech. rep., 1999.

[25] Fink, G., Best, D., Manz, D., Popovsky, V., and Endicott-Popovsky, B. Gamification for measuring cyber security situational awareness. In *International Conference on Augmented Cognition* (2013), Springer, pp. 656–665.

[26] Hamari, J., Koivisto, J., and Sarsa, H. Does gamification work?–a literature review of empirical studies on gamification. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on* (2014), IEEE, pp. 3025–3034.

[27] Honglk University Security Team. 15th hust hacking festival; resurrection. http://festival.hust.net.

[28] Insomni'Hack. Swiss security conference and ethical hacking contest. https://insomnihack.ch/.

[29] Jain, A., Gupta, N., and Nama, S. S. Break in ctf. https://felicity.iiit.ac.in/en/threads/breakin/.

[30] Jariwala, S., Champion, M., Rajivan, P., and Cooke, N. J. Influence of team communication and coordination on the performance of teams at the ictf competition. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (2012), vol. 56, SAGE Publications, pp. 458–462.

[31] Mainland Advanced Research Society. Capture the flag. https://bsidesvancouver.com/capture-the-flag/.

[32] McClain, J., Silva, A., Aviña, G. E., and Forsythe, C. Measuring human performance within computer security incident response teams. Tech. rep., Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States); Sandia National Laboratories, Livermore, CA (United States), 2015.

[33] Messerschmidt, J. E. Hackback: Permitting retaliatory hacking by non-state actors as proportionate countermeasures to transboundary cyberharm. *Colum. J. Transnat'l L. 52* (2013), 275.

[34] Namin, A. S., Aguirre-Muñoz, Z., and Jones, K. S. Teaching cyber security through competition: An experience report about a participatory training workshop. In *International Conference on Computer Science Education Innovation & Technology (CSEIT). Proceedings* (2016), Global Science and Technology Forum, p. 98.

[35] nuitduhack. Nuit du hack xv qualifications. http://www.hackzone.csi-ensi.tn.

[36] O'Brien, N. National labs' 'makers' to share their energy tech at the greatest show (and tell) on earth. https://www.llnl.gov/news/national-labs%E2%80%99-%E2%80%98makers%E2%80%99-share-their-energy-tech-%E2%80%98greatest-show-and-tell-earth%E2%80%99.

[37] Pi, R. Raspberry pi. *Raspberry Pi 1* (2013), 1.

[38] Plaid Parliament of Pwning. Plaid ctf. http://www.plaidctf.com.

[39] Pritchett, W. L., and De Smet, D. *Kali Linux Cookbook*. Packt Publishing Ltd, 2013.

[40] Pusey, P., Gondree, M., and Peterson, Z. The outcomes of cybersecurity competitions and implications for underrepresented populations. *IEEE Security & Privacy 14*, 6 (2016), 90–95.

[41] Pyatkin, A., hdhog, Tikhonov, D., and Volkov, A. International interuniversity open cybersecurity competition. https://github.com/VolgaCTF.

[42] Raj, A. S., Alangot, B., Prabhu, S., and Achuthan, K. Scalable and lightweight ctf infrastructures using application containers.

[43] Securinets. http://www.ctfsecurinets.com/.

[44] Sextos, G. Cyber wargame environment. Master's thesis, $\Pi\alpha\nu\epsilon\pi\iota\sigma\tau\acute{\eta}\mu\iota o$ $\Pi\epsilon\iota\rho\alpha\iota\acute{\omega}\varsigma$, 2015.

[45] SiBears. School ctf. http://school-ctf.org.

[46] SIGPwny. Uiuctf 2017. https://sigpwny.github.io/ctf.html.

[47] Silva, A., McClain, J., Reed, T., Anderson, B., Nauer, K., Abbott, R., and Forsythe, C. Factors impacting performance in competitive cyber exercises. In *Proceedings of the Interservice/Interagency Training, Simulation and Education Conference, Orlando FL* (2014).

[48] towerofhanoi. polictf 2017. http://www.polictf.it.

[49] ucsb-seclab. ictf service samples. https://github.com/ucsb-seclab/ictf-service-samples.

[50] U.S. Department Of Energy. National science bowl (nsb). https://science.energy.gov/wdts/nsb/.

[51] Uskov, A. Gamification in computer science. In *IIMSS* (2013), pp. 41–50.

[52] Vigna, G., Borgolte, K., Corbetta, J., Doupe, A., Fratantonio, Y., Invernizzi, L., Kirat, D., and Shoshitaishvili, Y. Ten years of ictf: The good, the bad, and the ugly. In *3GSE* (2014).

[53] Vykopal, J., and Barták, M. On the design of security games: From frustrating to engaging learning. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)* (Austin, TX, 2016), USENIX Association.

[54] Werther, J., Zhivich, M., Leek, T., and Zeldovich, N. Experiences in cyber security education: The mit lincoln laboratory capture-the-flag exercise. In *CSET* (2011).

[55] WhiteHat. Computer hacking contest. https://wargame.whitehat.vn.

[56] Williams, D., Archer, K., and Archer, J. Ctf in a box. http://www.cyberpanoply.com.

[57] Zang, M., Hou, D., and Wang, J. Ctf in a box. https://github.com/EasyCTF/openctf.

[58] Zendler, A. Computer science education teaching methods: An overview of the literature. *International Journal of Research Studies in Computing 4*, 2 (2015).