# PicoCTF: A Game-Based Computer Security Competition for High School Students

Peter Chapman
peter@cmu.edu
*Carnegie Mellon University*

Jonathan Burket
jburket@cmu.edu
*Carnegie Mellon University*

David Brumley
dbrumley@cmu.edu
*Carnegie Mellon University*

## Abstract

The shortage of computer security experts is a critical problem. To encourage greater computer science interest among high school students, we designed and hosted a computer security competition called PicoCTF. Unlike existing competitions, PicoCTF focused primarily on offense and presented challenges in the form of a web-based game. Approximately 2,000 teams participated, with students playing for an average of 12 hours. We present the game-based competition design, an evaluation based on survey responses and website interaction statistics, and insights into the students who played. Further we have released our platform and challenges as an open source project, which has been adapted into the curricula of 40 high schools. Since its release in August of 2013, the PicoCTF platform has been used to host six other capture-the-flag competitions.

## 1 Introduction

There have been a series of recent calls to improve computer science [23] and computer security education [10] at all levels in the United States. In particular, a 2010 Presidential Commission recognized a severe nationwide shortage of tens of thousands of computer security experts [10]. The skills gap begins in high school, with fewer than 10% of high schools in the United States offering the computer science AP test, and only 15,000 students opting to take the exam in 2011 [24]. In comparison, over 900,000 US high school graduates took at least one AP exam, and more than 200,000 students took the AP Calculus AB test alone [18]. At the collegiate level, the opportunities for computer science students to specialize in computer security are limited. A survey of 260 of universities found that 60% of programs do not offer any courses in network or information security [5].

We organized PicoCTF, a large-scale computer security competition for high school students, to introduce more students to the field and give instructors the tools to integrate hands-on computer security exercises into lessons. PicoCTF is a capture-the-flag competition (CTF). CTFs are organized as a fun, legal way for computer security students and professionals to practice and demonstrate their skill. In a standard-format CTF, teams race to answer computer security challenges, searching for digital "flags" hidden in servers, embedded in encrypted text, or obfuscated in binary programs. During the competition, which typically lasts from one to two days, teams earn points for submitting discovered flags. The team with the most points at the end of the competition is the winner. Challenges are generally designed with many possible solutions and help students acquire skills in computer forensics, cryptography, reverse engineering, binary exploitation, and web security.

Most CTFs target a small community of security-minded students and professionals. While recognized as a valuable tool for outreach and education [4, 5, 8, 14], CTFs remain a niche hobby. PicoCTF made great efforts to be inclusive to students of all backgrounds while maintaining the authenticity of the major competitions.

We had three goals in building PicoCTF: to encourage students to purse degrees in computer science or related disciplines, to introduce key computer security topics to students at a younger age, and to provide instructors with the materials and ideas to enhance classroom lessons with hands-on computer security exercises. We also looked to capture the fun and authenticity that have made existing CTF competitions so successful.

In order to achieve these goals, we recruited a team of five developers specializing in interactive experiences to build a story-driven game around which to present the PicoCTF challenges. The result was *Toaster Wars*, an interactive game embedded into the competition that featured cut-scenes, sound effects, and multiple levels. After months of publicity and recruitment, we held PicoCTF for a ten-day period in late May 2013 with 1,938 teams competing (estimating the number of participating students is discussed in §4.5). The competition featured 57 challenges across five major categories: forensics (16),

cryptography (8), reverse engineering (9), web and scripting exploitation (13), and binary exploitation (11). For most students, PicoCTF was strictly an extracurricular activity. The average participant spent about 12 hours experimenting with advanced computer science topics such as cryptographic ciphers, the client-server paradigm of the web, file system forensics, command injection, and program representation. Survey results and informal feedback from students and teachers were overwhelmingly positive, with every surveyed teacher planning to encourage their students to compete in PicoCTF 2014.

Our main contributions in this paper are as follows:

**High-School CTF Design** We present the design of PicoCTF, a game-based capture-the-flag competition targeted at high school students that introduces a variety of complex computer security topics. We discuss the challenge content and organization, competition rules, publicity efforts, and integration of the *Toaster Wars* game. PicoCTF differs from existing security competitions by placing challenges in the context of a story-based game to encourage participation among students not intrinsically motivated to pursue computer security.

**Large-Scale Evaluation** We evaluate PicoCTF based on survey responses and user interaction logs. We investigate the effectiveness of critical design choices, finding that younger students preferred a game interface more than older students and that students generally disliked challenges that required learning new tools.

## 2 Related Work

**Computer Security Education** Computer security is rarely taught in high school classrooms. Both the College Board's Advanced Placement Computer Science examination [17] and the ACM Model K–12 Curriculum [19] only discuss computer security and privacy in a social context. With a shortage of professional computer security experts [10], there have been a number of efforts to apply hands-on exercises [9, 13], games [12], and challenge-based learning methods [4] to computer security education at the collegiate level. The SEED project, for example, provides a series of highly structured exercises on attacking, implementing, and exploring computer systems [9]. The CyberCIEGE video game trains students to make high-level decisions about building, operating, and maintaining the computer and network security of a hypothetical sensitive lab [12]. In contrast to most prior work that brings computer security into the classroom, PicoCTF is a competitive event targeted at high school students that focuses on real-world techniques, tools, and applications.

**Competitive CTFs** PicoCTF is not the only computer security competition for high school students. The High School Cyber Forensics Challenge [7] focuses primarily on computer forensics. While the preliminary round is held online, the finals are hosted in-person, in conjunction with the Polytechnic Institute of New York University's Computer Security Awareness Week (CSAW) CTF. The CyberPatriot National High School Cyber Defense Competition [22] is an annual defensive competition that hosts about 1,000 teams [1]. Both events have underlying stories to motivate challenges. In contrast to the skills emphasized in the Cyber Forensics Challenge and CyberPatriot, PicoCTF places a strong emphasis on offensive techniques, encouraging students to creatively explore and experiment with computer systems. More closely related, the US Cyber Challenge periodically hosts Cyber Quest [6] to test skills ranging from vulnerability analysis to cryptography. Cyber Quest releases small sets of challenges once a month around various themes (e.g., network analysis). PicoCTF was a ten-day competition containing 57 challenges covering a wide range of topics in computer security.

**Educational CTFs** The hacker mindset of testing edge cases and trusting no implementation has been posited as an effective way to teach students to build more secure systems [3]. The applied, creative nature of CTF exercises has been identified as an effective tool for outreach and education [4, 5, 8, 14]. In particular, there have been efforts to closely integrate CTFs with supplementary or classroom lectures on a small scale (fewer than 100 students) [2, 4, 5, 11, 14, 16, 20, 21]. The principal difference from work applying CTFs exercises to the classroom is that PicoCTF targets high school students, not university students, and is an order of magnitude larger in scale.

## 3 Design

PicoCTF is a series of 57 independent challenges in computer forensics, cryptography, reverse engineering, and program exploitation. Students in grades 6–12 from the United States were eligible to compete on teams of five or fewer over a ten day period in the spring of 2013.

**Registration** When registering, each team in the competition was required to be associated with both a teacher and an academic institution. Registration asked that teachers be the sole point of contact so that so that no personal information on students would be recorded. There was no fee to compete, and registration was open until the end of the competition. There was no limit to the number of teams that could represent a single school, and not all members of a team had to attend the same school. Students and teams that did not meet these eligibility requirements were still allowed to participate, but were not shown on the scoreboard and were not eligible for prizes. We enforced these restrictions via manual investigation of anomalous activity, automated analysis of submission

Figure 1: The *Toaster Wars* Game Viewer

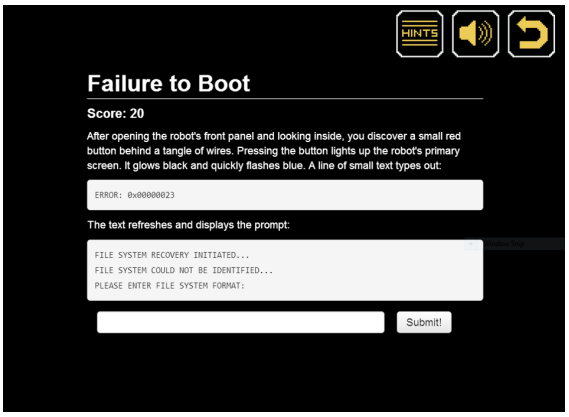


Figure 2: The Game-Based Problem Viewer

data, and for prize distribution, direct contact with school administrators.

**Gameplay** While CTF problems are traditionally presented in a straightforward text-based manner, we opted to build PicoCTF around a story-driven game experience called *Toaster Wars*. We designed the game to appeal to students who might not otherwise be interested in participating in a computer security competition. The game was divided into four levels, each advancing the story and increasing in difficulty. Level 1 began with the players discovering a crashed robot in their backyard. As the first level unfolded, players repaired the robot by completing challenges that required general critical thinking skills. The criteria for advancing was level-specific; for example, level 1 was completed after solving 3 out of the 5 available problems, at which point the player could continue to solve the remaining level 1 challenges or start level 2.

In levels 2 and 3, players recovered the robot's lost ship at a nearby spaceport before heading to space to participate in an intergalactic hacking contest in level 4. Level 2 was designed for students with introductory programming experience in languages such as Visual Basic or Alice. Level 3 was targeted at AP Computer Science students with a stronger background in programming. Finally, level 4 had a diverse set of problems across all categories, ranging from difficult to professional-CTF difficult. We opted to divide the competition into levels to better account for the vastly different backgrounds of participating students, and to ensure students of all skill levels could leave PicoCTF satisfied with their accomplishment.

PicoCTF supported two challenge viewers: the *Toaster Wars* game viewer and a text-based problem viewer. The game viewer was an HTML5 game where the player could explore and interact with the world, (Figure 1) clicking on objects to view challenges (Figure 2). The text-based problem viewer simply displayed the description for each challenge, ideal for older browsers and serious competitors. Note that the actual challenges were the same regardless of the viewer used. Many challenges asked the student to perform a privilege escalation attack on a vulnerable executable. We hosted these problems on a Linux server, which students could access either with SSH or through a web client. Additionally we provided an IRC-based chat room for students to discuss challenges with competition organizers. Finally, we created a series of seven lectures on general security topics to provide background for certain classes of challenges.

**Challenge Design** PicoCTF featured 57 challenges (also referred to as problems) across five major categories: forensics (16), cryptography (8), reverse engineering (9), web and scripting exploitation (13), and binary exploitation (11). In forensics challenges, students search for hidden data in images, network traffic, and file systems. Cryptography challenges require students to decipher text and audio messages encoded with classic ciphers as well as more modern encryption schemes. Reverse Engineering problems involve understanding the behavior of compiled, obfuscated, or cryptic program code. In web and script exploitation challenges, students attack PHP applications and Python programs using common techniques such as SQL Injection. Finally, binary exploitation problems require students to execute buffer overflow, format string, and ROP (return-oriented programming) attacks to gain control of a target systems.

The challenges in PicoCTF were designed to encourage students to learn and practice technical skills beyond the traditional high school computer science curriculum. The game was divided into four levels, as described above. Table 1 shows the number of teams that completed each level. We overview select problems in Table 2. In *Try Them All!*, for example, we introduced 1,279 teams to salted password storage. Students demonstrated their understanding by implementing a brute-force dictionary attack on a leaked password hash. Understanding password

hashes and how to implement them correctly is critical, particularly in light of recent high-profile password disclosures [15].

Examining a more traditional topic, *Byte Code* considered program representation, requiring students to either decompile a Java program or manually examine the byte code in a hex editor to locate a hidden key. 96 teams solved *ROP 1*, a binary exploitation challenge on writing a return-to-libc attack. This form of return-oriented programming is a technique typically taught at the graduate level. Overall, we successfully introduced thousands of students to complex computer security and computer science topics on both ends of our target difficulty spectrum.

**Rules and Scoring** We assigned each challenge a point value based on its predicted difficulty. When teams solved a given challenge, the point value for that challenge was added to their score. Teams could submit unlimited guesses to a given challenge. The winner of the competition was the team with the most points (the time of the last problem solved was the tiebreaker). Teams were encouraged to use all available resources but were forbidden to receive direct assistance from outside persons.

**Awards and Sponsors** PicoCTF 2013 was sponsored by 11 different organizations, primarily to support prizes. A team was eligible to compete if each individual on the team was a 6th–12th grade student in the United States. The school affiliated with a team was eligible if each member attended that school. The top three winning teams and schools received cash awards ($1000–$8000), credit to Amazon Web Services ($250–$1000), a selection of books from Wiley publishing, trophies, printed and signed certificates, and t-shirts. Awards for schools were intended to incentivize teachers to promote participation among their students.

**Ethics** Hacking tends to be misrepresented as a malicious activity. Done legally, hacking is not only a creative way to garner interest in computer science, but also the best possible way to learn about how computer systems work—similar to taking apart and reassembling a car as an educational exercise. However, as with most technical skills, there are malicious applications, and we felt it paramount to stress wherever possible that we do not condone illegally breaking into systems, stealing personal information, or disrupting computing services. We detailed our stance on the ethical implications of hacking in an introductory video and throughout the site. Further, the competition's story emphasized constructive applications of hacking, with tasks centered around the participant helping a broken robot get home.

**Outcome** 1,938 teams from 955 schools participated in the competition. The three winning teams came from three different schools and each completed all of the 57 challenges. On average, students completed 15 security-based challenges, spending an estimated 11 to 12 hours on the competition. Figure 3 shows the number of teams that attempted and successfully solved each challenge for a representative sample of the challenges in the competition, ordered by point value from most (top) to least (bottom). Reaction to the competition was overwhelmingly positive.

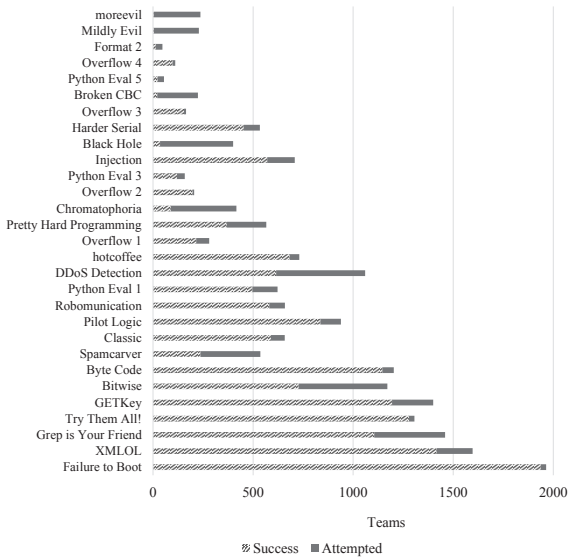


Figure 3: Challenge Completion Rate

## 4 Evaluation

All student data was collected and analyzed under the guidance and approval of our institutional review board (Tracking Number HS13-497).

**Data Collected for the Competition** Teams that registered for the competition provided a team name, affiliated school, and teacher name. No information about individual students was recorded.

During the competition, we kept track of every answer submitted by every team, both correct and incorrect. For each submission, we recorded the time, content and relevant problem identifier, as well as the IP address of the submission. Over the course of the ten day competition, we recorded 172,482 submissions from the 1,938 eligible teams. Using the provided names of schools and IP addresses of the corresponding submissions, we were able to infer the geographic location for 1,588 of the participating teams.

**Survey** On the final day of the competition, we contacted every team via email, requesting that they complete an online survey related to their experience competing in PicoCTF. Survey questions focused primarily on which aspects of the competition the student enjoyed the most, which components could be improved, and the computer

| Level | 1 (No Programming Experience) | 2 (Introductory Programming) | 3 (AP CS) | 4 (All Challenges) |
|---|---|---|---|---|
| **Completions** | 1,541 | 1,297 | 930 | 3 |

Table 1: The number of teams completing the four levels of PicoCTF

| Level | Challenge | Team Completions | Target Skills |
|---|---|---|---|
| 1 | First Contact | 1,368 | Network Traffic Analysis |
| 2 | CFG to C | 1,321 | x86 Assembly and Control-Flow Graphs |
| 2 | Try Them All! | 1,279 | Password Hashes, Salts, and Dictionary Attacks |
| 3 | DDoS Detection | 615 | Defensive Traffic Analysis |
| 3 | Byte Code | 1,146 | Program Representation |
| 3 | SQL Injection | 571 | Command Injection Attacks |
| 3 | RSA | 228 | RSA Implementation |
| 4 | Overflow 1 | 216 | Buffer Overflow |
| 4 | ROP 1 | 96 | Return-to-libc Attack |

Table 2: A sample of PicoCTF challenges and their educational contribution

science background of the participant. Responses were anonymous and cannot be linked with team-specific submission data. The survey questions are available in Appendix A. The survey was completed by 394 students and 21 instructors. Comparing the scores reported by the students in the survey to those of all of the teams in the competition, we found the distributions to be quite similar, with the exception of participants with scores under 500 (out of 5485), who were significantly underrepresented.

## 4.1 Educational Impact

One of the primary goals of PicoCTF was to introduce pre-collegiate students to advanced topics in computer science and computer security. As evident in the survey data, we largely achieved this goal. 83% students had never participated in a computer security competition, and the vast majority had no experience with hacking. Teams in the competition completed an average of 15 challenges (median of 12), and in doing so most teams learned to forge an HTTP cookie, read a control flow diagram, and brute force a hashed password.

According to both instructors and the students, PicoCTF was a positive educational experience. 76% of teachers surveyed reported that their students put more effort into the competition than they normally did in class. 67% of students believed they learned more playing PicoCTF than they normally did in class. Perhaps most importantly, every instructor surveyed would encourage their students to compete in PicoCTF 2014.

Among the students who took the survey, 15% reported that they competed in PicoCTF to fulfill a class requirement. Students required to participate tended to have lower scores and spend less time on the competition. We also observed that students competing as a class requirement tended to offer different explanations for why certain challenges were their favorites. Students participating for "Fun" or "To Learn," who made up the vast majority of students who took the survey, often described their favorite problems as challenging and that they learned a lot by completing them. In contrast, students competing as a class requirement preferred problems that were easy and more relevant to their existing knowledge.

## 4.2 The Game Viewer

The *Toaster Wars* game is one of the key aspects of PicoCTF that differentiates it from CTFs targeting college-age students. Designed to make the competition more inviting to beginners, *Toaster Wars* presents the challenges of PicoCTF within the context of an interactive story. More advanced students could opt to instead use the text-based problem viewer, which simply displays the description of the challenge.

In the survey, we asked students multiple questions about their experience with the *Toaster Wars* game viewer. 52% of students reported that they spent the majority of their time using the game as opposed to the text-based challenge viewer. This rate was considerably higher among middle school students, 79% of whom preferred the game viewer. Middle school students also had a higher opinion of the game, rating it an average of 4.05 on a scale from 1 ("Hated it") to 5 ("Loved it") compared to the average of 3.47 among high school students. We also observed that use of the game viewer was inversely related to overall performance in the competition (Figure 4). We suspect that this is not because the game viewer inhibited performance, but rather because younger and less experienced players tended to prefer the game.

Given the feedback in the survey, we believe that the game was an important and valued element of PicoCTF. In particular, the average age and performance of students

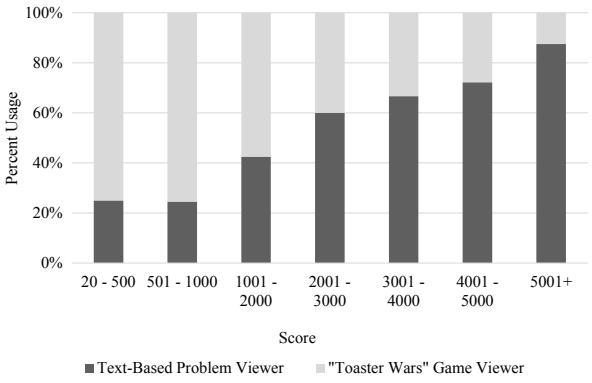


Figure 4: Interface Preferences

using the game viewer suggests that it succeeded in engaging younger students and those with less experience. Older and higher-scoring students cared less about the game but did not object to its presence. We plan to include a similar game in future instances of PicoCTF.

## 4.3 Challenge Preferences

PicoCTF contains a wide variety of security-related challenges, including ones that focus on cryptography, binary exploitation, and reverse engineering software. As evident in the survey data, students had strong opinions about which types of challenges they preferred and which they did not. 31% of students requested more web problems, while 22% wanted more cryptography. On the other end of the spectrum, almost half of the participants demanded fewer challenges focusing on binary exploitation, while forensics and reverse engineering challenges also proved to be unpopular.

Although many students claimed that certain types of problems were simply too difficult, we suspect that one of the key reasons students disliked classes of challenges was that they required the use of daunting and unfamiliar technology. Among the problems disliked most by students were *Black Hole*, a challenge which required editing the hex of a file system image then mounting it in Unix; *Yummy*, a web-based problem which required editing the value of a cookie; and *Evergreen*, a reverse engineering challenge that was best solved with a Java decompiler. While these were certainly difficult challenges, even simpler problems involving shell commands caused many students to struggle. One of the earliest challenges in the competition, *Grep is Your Friend* required students to find a specific string in a set of files, a task easily accomplished via the Unix command *grep*. What we expected to be a straightforward exercise was cited in the survey as the sixth most disliked problem in the competition.

In contrast, the most popular challenges were moderately difficult and required little familiarity with the command line or additional tools. The top three favorite challenges required exploiting an *eval* command in Python, devising an input that satisfied a series of equations in a different Python program, and deciphering an audio message which used a thinly veiled version of Morse Code. In addition to requiring few external tools, these favorite challenges were, unsurprisingly, more in line with what students might learn in a high school math or computer science class. The Advanced Placement Computer Science curriculum, for example, focuses primarily on design, implementation, and analysis of Java-like languages, skills that are far more relevant to the script and web exploitation challenges than to those that involved forensics or analysis of assembly [17].

Note that we are not suggesting that the distribution of challenges should be designed to match student preferences. Instead, problems that focus on topics not traditionally taught in high school need to be introduced more slowly and with additional learning materials. In addition to techniques, more effort also needs to be placed on teaching students how to use relevant tools. In future competitions, we plan to separately introduce useful resources such as the command line, debuggers, and web developer tools.

## 4.4 Evaluating Activity

In order to evaluate student engagement over the course of the competition, we determined the periods of time during which teams were most active. Based on the submission data, we modeled a team's *active periods* as the set of time intervals starting thirty minutes before and ending thirty minutes after each right or wrong submission made by that team. While one might expect teams to only submit a solution when they obtain a clear "flag," we observed that almost every team submitted significantly more incorrect answers than correct answers as they worked through the various stages of a problem. For this reason, we believe that our estimates are conservative, but still decent approximation of time students spent playing. Using this metric, we found that the average team was active on the site for 14 hours. Again, this is likely a conservative value, as normalized survey data shows that individuals participated in PicoCTF for an average of 11–12 hours.

Combining the active periods for all teams in the competition, we calculated the number of teams active at any given point over the course of the competition. We used the geographic data about each school to remove the effect of different time zones on the data. Unsurprisingly, overall activity declines over the course of the competition as teams are unable to solve more difficult problems or lose interest. Rather intriguingly, however, more teams were active on weekdays than on weekends. We suspect that because many teams had three or more students, it was more convenient for teams to work together after school

than on weekends. Indeed, though schooldays were more active, we found that students spent almost three times longer working after school than during school hours.

### 4.5 Measuring Individual Participation

Like most capture-the-flag competitions, PicoCTF used a single account for each team. Unfortunately, this setup makes it particularly difficult to calculate the number of students that participated in the competition. One potential method of measuring individual participation is by the number of unique IP addresses that accessed the competition. This metric undercounts students behind NAT connections (common at schools) and overcounts individuals using multiple connections over the course of the event. Another potential method is to count the number of unique cookies, the measurement used by Google Analytics and similar services. By this metric, PicoCTF had 17,831 unique participants. This method, however, overcounts students using multiple browsers and undercounts students using anti-tracking extensions. Ultimately, as long as teams share a common account, no method can account for multiple users working together on a single machine or a single user using multiple machines.

An alternative approach is to ask students directly. In a pre-competition survey, teachers reported an average team size of 4.07 students per team. In the post-competition survey, however, the average reported size was 2.38. These results suggest that the number of participants was somewhere between 4500 and 8000 students. Including the ineligible teams, this number increases to 5000–8500 students. The only accurate way to measure the number of participants is to use individual, rather than group accounts. Group accounts, however, offer more privacy to the competition participants, which is especially important to consider for high school-age students. For future PicoCTF competitions, we will evaluate the trade-off between increased privacy and an accurate measurement of the competition's impact.

## 5 Demographics

PicoCTF was targeted primarily at high school students in the United States, though middle school students could also participate. The 1,938 teams came from 955 different affiliations, with 55 schools having five of more teams. The largest number of teams from a single school was 49.

Over 90% of teams surveyed were affiliated with a high school, and individuals were mostly in grades 11 or 12. A large portion of students reported prior programming experience in object-oriented languages (61%) and on web applications (49%). Surprisingly 30% of students even claimed to have prior experience hacking, though the survey did not define the term. The vast majority of students become aware of the competition through a teacher (61%), classmate (19%), or parent (7%). Most students reported participating for fun or as a learning experience.

## 6 Future Work

The first iteration of PicoCTF was an informative experience in hosting large-scale, highly technical competitions for a wide range of students. Perhaps the most surprising result was that high school students were able to solve challenges easily presentable in undergraduate and graduate courses. We did observe, however, that students had difficulty simultaneously learning new concepts and new tools. In future iterations of the competition we plan to create small interactive environments where students are introduced to concepts and then apply what they learned with real-world tools.

With PicoCTF, we hope to teach students about computer science and computer security while encouraging further study in these fields. For PicoCTF 2013 we did not record longitudinal data on student ambitions and skills, preventing us from fully gauging our success at achieving these goals. In the future we plan to add a pre-competition survey and perform closely controlled experiments on smaller groups outside of the competition.

## 7 Conclusion

Capture-the-flag competitions are often used to practice computer security at the collegiate level. With PicoCTF, we successfully adapted the format for a high school audience. The innovations we introduced, including the interactive game and level system, were largely effective at engaging students at range of experience levels. Our platform is available as an open source project[1] and has been adapted into the curricula of 40 high schools through Project Lead the Way. The platform has also been used to host six other capture-the-flag competitions: PlayTJ CTF, BoilerQuest 2013, CiscoCTF 2013, IOCTF 2013, ACTF 2014, and HSCTF 2014. The open, approachable, and engaging nature of PicoCTF has proven highly desirable as computer security training materials for a diverse set of audiences.

**Acknowledgments** We would like to thank each of our sponsors for making PicoCTF possible: Symantec, the Entertainment Technology Center, Intel, Microsoft, CloudShark, Amazon Web Services, the Software Engineering Institute, the Information Networking Institute, Pearson, Wiley, and the National Security Agency. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 0946825.

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

## References


[1] Acohido, B. Cyberpatriot Competition Preps Young Cyberdefenders. http://www.usatoday.com/story/tech/2013/03/18/cyberpatriot-high-school-cyber-defenders-contest/1995303/, 2013.

[2] Adams, W. J., Gavas, E., Lacey, T., and Leblanc, S. P. Collective Views of the NSA/CSS Cyber Defense Exercise on Curricula and Learning Objectives. In *Cyber Security Experimentation and Test* (2009).

[3] Bratus, S. What Hackers Learn that the Rest of Us Don't: Notes on Hacker Curriculum. *IEEE Security & Privacy Magazine 5*, 4 (July 2007), 72–75.

[4] Cheung, R. S., Cohen, J. P., Lo, H. Z., and Elia, F. Challenge Based Learning in Cybersecurity Education. In *International Conference on Security and Management* (2011).

[5] Cheung, R. S., Cohen, J. P., Lo, H. Z., Elia, F., and Carrillo-Marquez, V. Effectiveness of Cybersecurity Competitions. In *International Conference on Security and Management* (2012).

[6] Counter Hack Challenges. US Cyber Challenge: Cyber Quests. http://uscc.cyberquests.org/, 2013.

[7] Cyber Security Awareness Week. High School Cyber Forensics Challenge. http://www.poly.edu/csaw2012/highschool-cyberforensics, 2012.

[8] Doupé, A., Egele, M., Caillat, B., Stringhini, G., Yakin, G., Zand, A., Cavedon, L., and Vigna, G. Hit'em Where it Hurts: A Live Security Exercise on Cyber Situational Awareness. In *Computer Security Applications Conference*, ACM Press (New York, New York, USA, 2011), 51–61.

[9] Du, W., Jayaraman, K., and Gaubatz, N. B. Enhancing Security Education with Hands-On Laboratory Exercises. In *Symposium on Information Assurance*, no. 0618680 (2010), 56–61.

[10] Evans, K., and Reeder, F. A Human Capital Crisis in Cybersecurity. Tech. Rep. November, CSIS Commission on Cybersecurity for the 44th Presidency, 2010.

[11] Fanelli, R. L., and O'Connor, T. J. Experiences with Practice-Focused Undergraduate Security Education. *Workshop on Cyber Security Experimentation and Test* (2010).

[12] Irvine, C., and Thompson, M. Active Learning with the CyberCIEGE Video Game. In *Workshop on Cyber Security Experimentation and Test* (2011).

[13] Irvine, C. E. Amplifying Security Education in the Laboratory. In *First World Conference on Information Security Education* (1999).

[14] O'Leary, M. Small-Scale Cyber Security Competitions. In *Colloquium for Information Systems Security Education* (2012), 103–110.

[15] Popkin, H. A. LinkedIn Confirms Password Leak, eHarmony Has One, Too. NBC News, 2012.

[16] Schepens, W., and James, J. Architecture of a Cyber Defense Competition. In *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, IEEE (2003), 4300–4305.

[17] The College Board. Computer Science A Course Description. Tech. rep., The College Board, 2010.

[18] The College Board. AP Report to the Nation. Tech. rep., The College Board, 2012.

[19] Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., and Verno, A. A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee. Tech. rep., Computer Science Teachers Association, 2003.

[20] Vigna, G. Teaching Network Security through Live Exercises. In *Conference on Information Security Education* (2003), 3–18.

[21] Werther, J., Zhivich, M., Leek, T., and Zeldovich, N. Experiences In Cyber Security Education: The MIT Lincoln Laboratory Capture-the-Flag Exercise. In *Workshop on Cyber Security Experimentation and Test* (2011).

[22] White, G. B., Williams, D., and Harrison, K. The CyberPatriot National High School Cyber Defense Competition. *IEEE Security & Privacy 8* (2010), 59–61.

[23] Wilson, C., Sudol, L. A., Stephenson, C., and Stehlik, M. Running on Empty: the Failure to Teach K-12 Computer Science in the Digital Age. Tech. rep., Association for Computing Machinery, 2010.

[24] Yadav, A., and Korb, J. T. Learning to Teach Computer Science. *Communications of the ACM 55*, 11 (Nov. 2012), 31.

# A Survey

**1** With what kind of school are you affiliated?
- ☐ Middle School
- ☐ High School
- ☐ Home School
- ☐ Magnet School
- ☐ Other

**2** Should the competition have been longer or shorter?
- ○ More time would have been helpful
- ○ 10 days was great
- ○ The competition was too long

**3** What kind of problems would you want to see more of?
- ○ Binary Exploitation
- ○ Script Exploitation (non-web)
- ○ Web
- ○ Cryptography
- ○ Forensics
- ○ Reverse Engineering
- ○ Other

**4** What kind of problems would you like to see less of?
- ○ Binary Exploitation
- ○ Script Exploitation (non-web)
- ○ Web
- ○ Cryptography
- ○ Forensics
- ○ Reverse Engineering
- ○ Other

**5** What did you think of the problem difficulty?
1 (Too Easy) – 5 (Too Hard)

**6** What was your favorite problem? Why?

**7** What was your favorite problem? Why?

**8** What did you think of the prizes? What prizes would you like?

**9** Are you a competition participant or a team advisor?
- ○ Competition Participant (Student) — Go to 10A
- ○ Team Advisor (Teacher) — Go to 10B

**Student-Only Portion of the Survey**

**10A** What grade are you currently in?

**11A** How many members are competing on your team?

**12A** How many points did your team earn?
- ○ 0
- ○ 20–500
- ○ 501–1000
- ○ 1001–2000
- ○ 2001–3000
- ○ 3001–4000
- ○ 4001–5000
- ○ 5001+

**13A** What types of competitions have you previously participated in?
- ○ Programming
- ○ Computer Security
- ○ Math
- ○ Science Bowl
- ○ Other

**14A** How did you hear about the competition?
- ○ Teacher
- ○ Friend
- ○ Parent
- ○ News Organization
- ○ Social News
- ○ Other

**15A** What kind of background do you have in computer science?
- ☐ Tinkering with computer hardware
- ☐ Tinkering with computer software
- ☐ Discrete math
- ☐ Scripting
- ☐ Object oriented programming
- ☐ Hacking
- ☐ Application programming
- ☐ Web programming
- ☐ Robotics
- ☐ Other

**16A** Why did you compete?
- ○ Class Requirement
- ○ Help a friend
- ○ Parental Suggestion
- ○ Fun
- ○ Prizes
- ○ To Learn
- ○ College Application
- ○ Other

**17A** Roughly how many hours did you spend working on challenges?
- ○ 0–3
- ○ 3–6
- ○ 6–9
- ○ 9–12
- ○ 12–18
- ○ 18–24
- ○ 24–40
- ○ 40+

**18A** Of the 10 days, how many did you spend working on at least one problem?

**19A** Did your team receive time off from their/your other classes to work on picoCTF 2013?
- ○ Yes
- ○ No

**20A** Compared to typical course work, how much do you think you learned?
1 (Less) – 5 (More)

**21A** Which of these pages did you use?
- ☐ Game Viewer
- ☐ Basic Viewer
- ☐ Chat
- ☐ Shell
- ☐ News
- ☐ Learn

**22A** In which problem viewer did you spend most of your time?
- ☐ Game Viewer
- ☐ Basic Viewer

**23A** What did you think of the game viewer?
1 (Hated It) – 5 (Loved It)

**24A** Do you feel the competition benefited from the game?
1 (Worse) – 5 (Better)

**25A** Did you watch any of the videos in the Learn section?
- ☐ Yes
- ☐ No

**26A** How helpful was the material in the Learn section?
1 (Unhelpful) – 5 (Helpful)

**27A** How helpful was the Chat?
1 (Unhelpful) – 5 (Helpful)

**28A** How likely are you to major or pursue a career in computer programming or computer science?
1 (Unlikely) – 5 (Likely)

**Teacher-Only Portion of the Survey**

**10B** How did you hear about the competition?
- ☐ Teacher
- ☐ Friend
- ☐ Parent
- ☐ News Organization
- ☐ Social News
- ☐ CSTA
- ☐ Other

**11B** Will you encourage students to compete again next year?
- ☐ Yes
- ☐ No

**12B** How would you compare the effort your students put forth during the competition to their normal class effort?
1 (Less effort than usual) – 5 (More effort than usual)