pico-Boo!: How to avoid scaring students away in a CTF competition

Kentrell Owens
kentrelo@cmu.edu

Alexander Fulton
afulton@andrew.cmu.edu

Luke Jones
ltjones@cmu.edu

Martin Carlisle
carlislem@cmu.edu

Carnegie Mellon University
Pittsburgh, PA, USA 15213

Abstract - The lack of computer security experts poses a challenge for the private sector and national security. To encourage middle & high school students to learn more about cybersecurity, picoCTF was created in 2013. picoCTF is a "capture the flag" computer security exercise built on top of a video game that teaches students technical skills such as reverse engineering, forensics, cryptography, and binary exploitation. The challenges are specifically designed to be hackable and provide a safe and legal way to explore cyber security. Since the first competition in 2013, picoCTF has grown from around 2,000 teams to 8,000 eligible middle & high school US & CA teams and over 27,000 total global participants in the 2018 competition. Two key changes have been implemented since the competition's inception to improve learning outcomes and increase student engagement. More introductory and intermediate difficulty problems were added to each category, gradually increasing in difficulty. Also, a new "classroom" feature was added to the competition that allows teachers to create *internal scoreboards and track student progress. An analysis of the results of the 2018 competition shows that these new problems kept students engaged for more problems in the competition, and students with teachers who utilized the classrooms feature performed better than students with teachers who did not.*

Keywords: *cybersecurity, capture the flag, education*

1. INTRODUCTION

Computing devices evolve quickly and unpredictably. The Internet, starting as ARPANet in the 1970's evolved to be a critical part of worldwide infrastructure in less than a single human lifetime. However, not only is the Internet *another* critical infrastructure, it is infrastructure that critically supports other infrastructure. In other words, the ability of computing devices to make almost anything faster, more efficient or cheaper has invaded travel, commerce, banking, warfare and communication - among other industries. Computing devices have evolved from oversized calculators to the glue that holds life as we know it together.

As the famous malicious software worm "Stuxnet" showed, that glue can be subtly compromised with disastrous effects. The nuclear power facility that Stuxnet compromised was not even directly connected to the Internet, and yet the malware managed to infiltrate and subvert significant infrastructure. Even more insidious, Stuxnet did not merely destroy the hardware of the power facility, but instead created random disturbances that affected the scientists and engineers like psychological warfare, and ultimately destroyed much more hardware over time than a more overt attack.

Stuxnet represents the most sophisticated cyber-based attack that has ever been detected. At the present, cyber-criminals are light years from this sort of sophistication; however, now is the time to prepare the next generation of cyber-defenders. What better way than to teach them the principles behind something like Stuxnet in hands-on, bite-sized CTF challenges?

A CTF or "capture-the-flag" competition requires participants to use various techniques to solve a challenge/problem and acquire a "flag." To acquire these flags, participants must use various cyber security exploits. While most CTF competitions are college-level or above, picoCTF specifically targets middle & high school students. The code for its framework is open-source and has been used to host other CTF competitions. The top ranking teams are eligible for cash prizes and trips to Carnegie Mellon University (CMU), where the competition is run. The 2018 picoCTF competition had 27,221 total participants from 154 countries with 13,596 of those participants being eligible for prizes. 3,313 of the competitors were high school seniors.

In this paper we highlight the following contributions within the picoCTF competition:

- We describe our methodology for breaking down complex problems into simpler ones to gradually teach students the skills they need to meet specific learning objectives.
- We introduce our novel "Classroom" feature from the competition. The use of this feature correlated with increased student engagement in the 2018 competition.
- We improve the state of cybersecurity education and awareness for high/middle school students by sharing our findings.

2. RELATED WORK

[redacted]PICOCTF

Although CTFs originated as fun, extracurricular activities, research has shown that they can be useful components of undergraduate cybersecurity courses. Students displayed higher motivation, more self-directed learning, and the ability to push their boundaries of their own knowledge when playing CTFs as part of coursework [7]. picoCTF extends this idea to middle & high school education. When picoCTF first began, it was a novel, game-based CTF competition that used offensive-strategies to teach middle & high school students cybersecurity concepts. In its inaugural running, the competition had around

2,000 teams competing and 57 problems [8]. This has grown to over 8,000 teams and 108 problems. A paper written after the first competition outlined the design of the overall competition, and a companion paper written by the designers of the game ("Toaster Wars") detailed the storyline and implementation of the game portion [6, 8]. A subsequent paper introduced "Automatic Problem Generation," which gave users a semi-unique instance of a problem and allowed the competition's administrators to have insight into whether users were cheating [5], and the principle is still being used in the current competition.

[redacted]OTHER CTFS

Several other CTF competitions aim to achieve similar goals to picoCTF, such as HSCTF ("High School Capture The Flag") which is a CTF run by high school students for other high school students. Unlike picoCTF, HSCTF does not focus exclusively on computer security, and its problems cover a wider range of general computer science concepts [4]. Cyber Security Awareness Week (CSAW) is a college student-run CTF that also has a Jeopardy-style CTF for high school students called "Red Team" (formerly High School Forensics). The qualification round is online but the finals are in person [3].

On the less technical side, CTF Unplugged is designed to teach students about the different challenges cybersecurity professionals face in the workplace without actually requiring them to use any of the technology that professionals do. This allows students with little to no background in cybersecurity to participate in a CTF, thereby lowering the barrier to entry [11]. For example, students participating in the project use data generated by Wireshark [12] without ever having to install or use the tool itself. Students enter "missions" that are made to teach specific skills by solving problems, with each problem broken up into smaller tasks [11].

3. DESIGN

In this section we describe our strategies for improving student engagement in picoCTF. These include problem development, the learning guides we developed as resources for students, and the classroom feature we added to the competition.

3.1 PROBLEM DEVELOPMENT

As in the 2017 picoCTF, there were six categories of problems within the 2018 competition: Web Exploitation, Forensics, Cryptography, Binary Exploitation, Reverse Engineering, and General Skills. The first five of these categories align with traditional CTF categories while General Skills contained some problems with an overarching goal that someone with no knowledge could learn enough to interact with later challenges without being intimidated by the setup of the problem. This included things like basic interactions with a linux terminal and connecting to remote servers.

The 2018 competition had 108 problems total, an increase from 71 problems in the 2017 competition. The problem development process took about eight months total and involved establishing the educational learning objectives, developing problems that spanned these learning objectives, and deploying and testing these problems. Problem development was a joint effort between two core developers on the picoCTF education team, under the advisement of the Faculty Education Director, along with a number of individual problem developers. These people include alumni, students from the Information Networking Institute, employees at CMU’s Software Engineering Institute, and students from CMU’s premier CTF team, Plaid Parliament of Pwning (PPP), the team with the most wins in the history of DEFCON’s CTF competition [10], as well as a few other developers not associated with CMU. The two core developers made the majority of the problems, particularly the easier ramp problems while developers from PPP focused on developing some of the more novel and difficult problems.

The learning objectives were established using Association for Computing Machinery’s (ACM’s) 2017 report on curriculum guidelines for cybersecurity education [1] as well as the picoCTF education team’s experience of which skills are needed to succeed in a CTF competition and which skills/knowledge are required to be a proficient practitioner in computer security. Problems were designed to reflect these learning objectives, with gradual increases in difficulty. The 2017 competition analysis showed sharp drop-off in student participation after they completed the first few easier introductory problems. We hypothesized that the introduction of more beginner and intermediate-level problems would keep students engaged longer in the competition and help them learn the concepts better. For example, the series of buffer overflow problems presented a number of different topics from the history of buffer overflow exploits. This demonstrated how patching one security vulnerability simply means that exploiters become more creative in finding ways to exploit a program. With this idea in mind, we led the students through the beginning of the Binary Exploitation category.

We broke this down to the most basic building blocks we could test in Buffer Overflow 0, which was just to show how improper coding could cause overwriting of data. By putting too many characters into an input the participant could modify the return address of the stack and cause the program to crash. Upon crashing, the flag was printed out. With that understanding, we then continued to our next objective developed in Buffer Overflow 1: showing how modification of the return address could then be exploited. By leaving an uncalled function stored in memory, students were able to change the return address from the current function using an overflow and receive the flag. This allowed us to start tying in more objectives, such as how stack is set up, and a bit about how the returns are utilized in an attack. In order to drill home the idea of the layout of the stack, we introduced Buffer Overflow 2. This problem required knowledge of stack layouts beyond the return address to ensure that proper arguments were also passed into the relevant function, giving a bit more reality to the situation. Rounding out this series was Buffer Overflow 3, which introduced the first idea of a specific security measures put in place to combat some of these attacks that had occurred in the earlier problems. Placing a four-byte canary on the stack and checking if it had changed later gave the problem further similarity to a real-world challenge.

While this may seem like a large number of problems devoted to a relatively small number of learning objectives, we believe this focus allowed students to develop a deeper understanding of the fundamentals crucial to solving more challenging problems. Once a student feels that they have accomplished something and have an understanding of the basics, we believe they are much more apt to continue on to the more challenging problems, which in this case evolved into return-oriented programming problems that involve running shellcode and return-to-libc attacks.

An additional example of the ramps we created was within the Reversing category, a category we see as being notoriously difficult to get students engaged and up to speed in. We started with problems simpler than what would be considered reversing in most CTFs, and tried to lead students through the basics of assembly programming using the style of CTF problems. Each of these assembly problems were a basic source code of an assembly function, and given an input (or set of inputs), the output had to be provided instead of the standard flag format.

Assembly 0 started by introducing the basics of the structure of assembly code and ensuring the participants understand how arguments are passed in. We then led them into Assembly 1 introducing the idea of if statements, and how they are expressed using branches and conditional actions. Assembly 2 continued the idea of branches into a simple loop, adding operations done repetitively, furthering the concept of branching and helping participants to recognize what a loop looks like in assembly. This then led to Assembly 3, which added complexity by requiring participants to track the order of the bytes' endianness. Our final explicitly labeled problem in this ramp was part of our transition that we designed to help people think differently about these types of problems. This problem had 79 possible characters that were assembled into the flag that was itself in 43 different parts. This complexity led participants to the necessity of compiling the program using an assembler; we wanted to demonstrate that in many situations trying to analyze code by hand is not feasible, and therefore using other tools are going to be necessary for all but the most motivated of solvers.

Throughout this set of problems we tried to lower some of the barriers to entry to a topic that can be intimidating to so many new students, and tried to incorporate some of the basic techniques that are necessary to build off of for more sophisticated problems. We incorporated the concept of ramps in each of the other categories, including progressions through the basics of interacting with command line, working through the history of cryptography, and levels of securing websites.

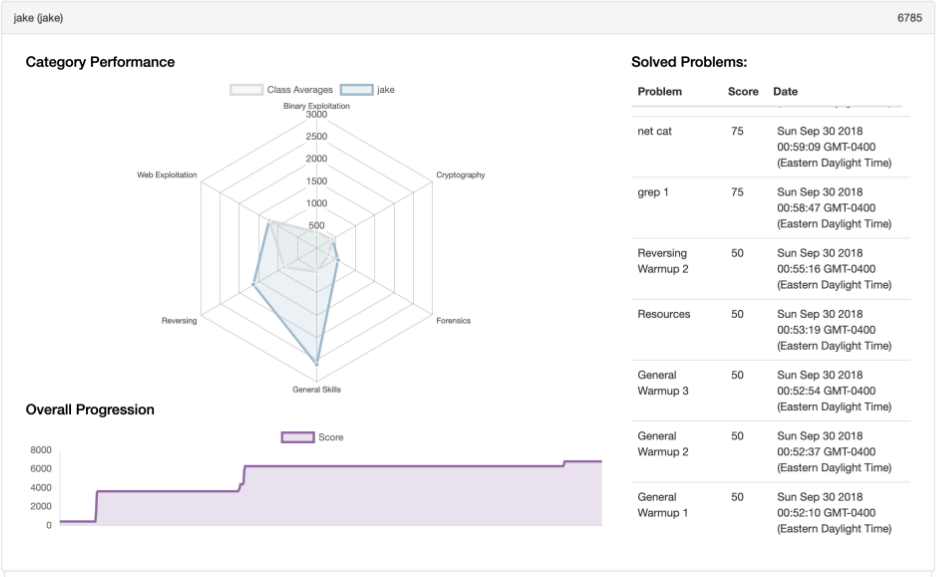
3.2 LEARNING GUIDES

To help beginners who might not have much experience with programming, we developed a series of learning guides [9]. These guides are not meant to be exhaustive guides or tutorials for solving problems; they simply introduce some high-level concepts and terms that students need to understand to begin participating and solving problems in the competition. The guides aim to lower the barriers to entry for students who want to participate in the competition but have never taken a programming/computer/security course. They also centralize information so that initial research is much easier for the students, and teachers have a starting point of materials for instruction. Our goal was to have these guides go hand in hand with some of our initial ramps to mitigate some of the frustration that may arise from not knowing enough about a subject to successfully research the topics. Once students had the basic terminology and fundamentals, we left it up to them to continue researching to have success with the more challenging problems. These guides were split by overarching category and posted on our website, with one introductory problem requiring participants to navigate to the resource list webpage to find a flag. The General Skills learning guide included topics such as using the command prompt, using SSH, binary numbers, and little/big endianness, which are all concepts that must be used to complete not only early problems but are also essential to access and understand later problems.

3.3 CLASSROOMS

This year, in addition to the learning guides we wrote, in order to encourage more participation in a structured educational setting, teachers were able to create a "classroom" which had an internal scoreboard. Since we are an education-focused competition, we wanted to provide more control and insight to the participating teachers. This feature was added to allow teachers to create smaller competitions within their classes, as a subset of the national/international scoreboards to allow students to compete against one another in addition to the world wide competition. They were able to create these subsets on a per user or per team basis, aligning with the rest of the competition.

This feature was designed to encourage students to remain engaged in the competitions and try to compare themselves and compete with each other, as students in their“classrooms” may have had more similar backgrounds compared to everyone in the entire competition. In the past, teachers were only able to view the public scoreboard, and since the displayed names were often nicknames, this created a real barrier toward trying to utilize the scoreboard in an education focused manner. This year, teachers could gain much greater insight into each student’s scores by monitoring the progress of each student, gaining insight into every problem participants have solved, performance in each category, performance overall, as well as view comparisons between students and data regarding how their class’ performance compares with that of other classrooms (see Figure 1).



Solved Problems:

Problem	Score	Date
net cat	75	Sun Sep 30 2018 00:59:09 GMT-0400 (Eastern Daylight Time)
grep 1	75	Sun Sep 30 2018 00:58:47 GMT-0400 (Eastern Daylight Time)
Reversing Warmup 2	50	Sun Sep 30 2018 00:55:16 GMT-0400 (Eastern Daylight Time)
Resources	50	Sun Sep 30 2018 00:53:19 GMT-0400 (Eastern Daylight Time)
General Warmup 3	50	Sun Sep 30 2018 00:52:54 GMT-0400 (Eastern Daylight Time)
General Warmup 2	50	Sun Sep 30 2018 00:52:37 GMT-0400 (Eastern Daylight Time)
General Warmup 1	50	Sun Sep 30 2018 00:52:10 GMT-0400 (Eastern Daylight Time)

Figure 1: Example classroom view of one student's progress

4. EVALUATION

4.1 2017/2018 PROBLEM SOLVES

Our objective was to improve student engagement in the competition by increasing the number of problems students solved. We defined student engagement as the percentage of students remaining in the competition as a function of percentage of problems solved. If our changes were effective, we would expect there to be a statistically significant increase in the percentage of students playing later in the CTF.

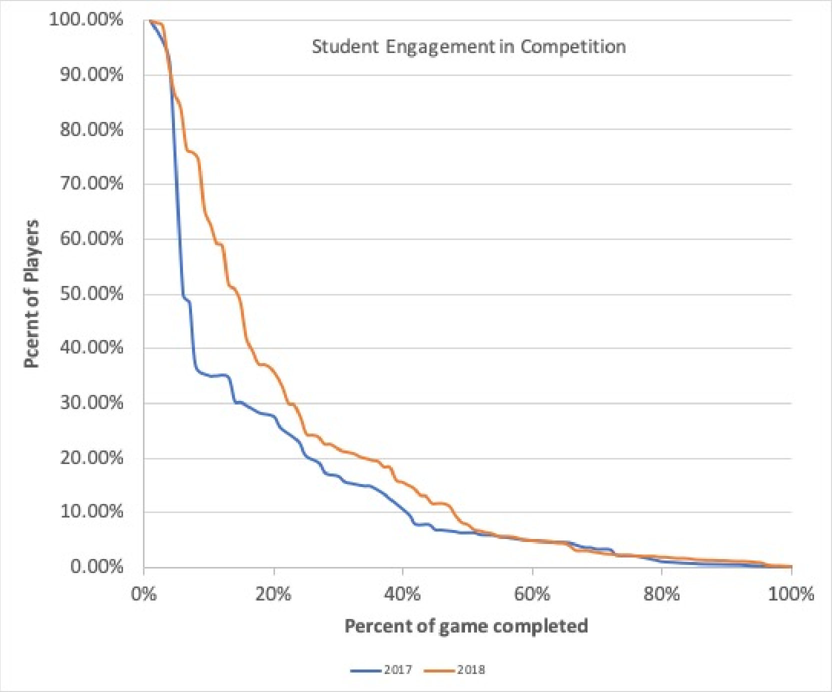


Figure 2: Student Engagement in the overall competition

Figure 2 shows that picoCTF 2018 had a higher percentage of students engaged further in the competition than 2017. A Mood's median test, a useful test when testing the equality of medians from two or more non-parametric data populations [2], showed that this difference is statistically significant ($p < 0.046$). Student engagement increased significantly. This is the case even after an increase of the number of problems which were possible (going from 71 to 108), giving validity to our hypothesis regarding a focus on the gradual increase in difficulty improves engagement in the game.

4.2 STUDENTS PERFORMANCE IN CLASSROOM FEATURE

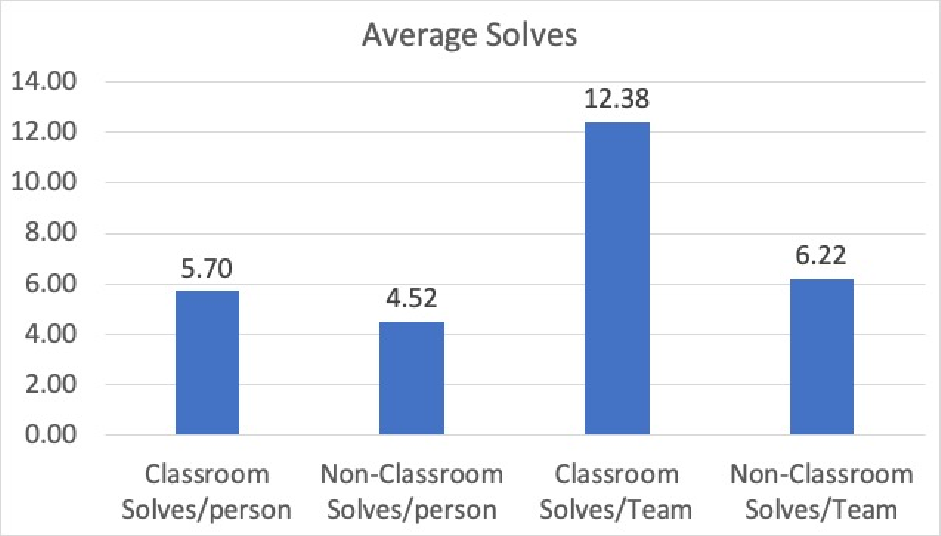


Figure 3: Classroom impact on Solves

As seen in Figure 3, students who competed in classrooms outperformed students who were not in classrooms (Mood's median, $p < 0.001$). This also demonstrates that being involved in the competition as a team positively impacts on the total number of solves. There may be several reasons for this impact, such as incentives from instructors (such as a grade or extra credit), or increased competitiveness due to the internal nature of the scoreboard. Another reason for increased student performance could be that teachers who used the classroom feature are overall more invested in their students' participation in the competition and likely to offer more active help during the competition. Regardless of the reason, these results indicate that a good way to increase engagement from middle & high school students is to develop our relationships with teachers and making tools to enhance their use of the platform. We hope to continue increasing the number of teachers involved in exposing their students to competitions like this one, and take advantage of the tools designed for them.

It can also be shown that when competing as a member of a team, there is an increase in the scores achieved. While we are unsure of a specific root cause of this, and it may be inflated slightly by having multiple people to work on different problems, we also theorize that this increase is due to the increased interaction and understand that comes with working together on a team.

5. FUTURE WORK

While we see great strides being made in bringing cyber security education to middle and high school students, there are always opportunities for improvement. The concept of ramps is something that is not new, but we have added to the body of research around this topic. In the future, we intend to continue to focus on improving these ramps. Trying to bring students up to speed to an evolving field means that this effort can never be truly static, and a sharp drop off in solves in multiple categories shows that there is certainly still significant room for improvement. Providing additional resources and capabilities to teachers through both framework features and resources that they can use to help their students succeed in the competition will also continue to be a focus of ours going forward. Further study is necessary into the demographics of the students who are involved in the competition, whether that be socio-economic, gender or otherwise. Ensuring the competition is as inclusive and accessible as possible will continue to be a priority.

6. CONCLUSION

CTFs present one method for the cybersecurity industry to engage middle/high school students to expose them to the field and cultivate interest in pursuing a career in the field in the future. However, students may become intimidated or frustrated with sharp increases in difficulty of problems while participating in CTFs. We outline a strategy that combines learning objectives and problem development to create problems that gradually expose students to concepts with the goal of increasing student engagement and reducing dropout rates. In picoCTF 2018, the addition of more introductory and intermediate-level problems to each category, with gradual increases in difficulty, proved to be an effective strategy in improving the percentage of problems students completed in the picoCTF 2018 competition when compared to the 2017 competition. The use of picoCTF's new classroom feature also correlated with a high number of problems solved by students. While this correlation may be due to additional factors (incentives, overall teacher engagement, etc), the features is unique among CTFs we have encountered, and it is simple to adopt to other CTF frameworks.

7. ACKNOWLEDGMENTS

We would like to this the following people for making this paper possible: Maverick Woo, Megan Kearns and Ivan Liang. We would also like to thank each of our sponsors for making picoCTF possible: Ryerson University, Boeing, Amazon Web Services, Eaton, Carnegie Mellon Information Networking Institute, and Siemens.

REFERENCES

[1] ACM, IEEE, AIS, IFIP. Cybersecurity Curricula 2017. 2017. Curriculum guidelines for post-secondary degree programs in cybersecurity. Available from https://www.acm.org/binaries/content/assets/education/curricula-recommendations/csec2017.pdf.

[2] Charles Zaiontz. 2019. Mood's Median Test (for two samples) | Real Statistics Using Excel. Retrieved April 28, 2019 from https://www.real-statistics.com/non-parametric-tests/moods-median-test-two-samples/

[3] Cyber Security Awareness Week. CSAW :: Home. Retrieved from https://csaw.engineering.nyu.edu/.

[4] HSCTF - The First CTF By High Schoolers For High Schoolers. (March 2019). Retrieved from https://hsctf.com/.

[5] Jonathan Burket, Peter Chapman, Tim Becker, Christopher Ganas, and David Brumley. 2015. Automatic Problem Generation for Capture-the-Flag Competitions. In *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*. USENIX Association, Washington, DC, USA.

[6] Kaiyang Zhang, Shihao Dong, Guoliang Zhu, Danielle Corporon, Tim McMullan and Salvador Barrera. 2013. picoCTF 2013 - Toaster Wars: When interactive storytelling game meets the largest computer security competition. In *IEEE International Games Innovation Conference (IGIC 2013)*, Vancouver, BC, pp. 293-299. doi: 10.1109/IGIC.2013.6659158

[7] Martin Carlisle, Michael Chiaramonte, and David Caswell. 2015. Using ctfs for an undergraduate cyber education. In *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*. 2015.

[8] Peter Chapman, Jonathan Burket, and David Brumley. 2014. PicoCTF : A Game-Based Computer Security Competition for High School Students. In *USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 2014)*. USENIX Association, San Diego, CA, USA.

[9] picoCTF. 2018. picoCTF - CMU Cybersecurity Competition - Resources. Retrieved from https://picoctf.com/resources.

[10] Staff Writer. 2017. PPP: The strongest team in DefCon history - College of Engineering at Carnegie Mellon University (Fall 2017). Retrieved April 22, 2019 from https://engineering.cmu.edu/news-events/magazine/fall-2017/ppp-defcon.html.

[11] Vitaly Ford, Ambareen Siraj, Ada Haynes, and Eric Brown. 2017. Capture the Flag Unplugged: an Offline Cyber Competition. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (SIGCSE '17). ACM, New York, NY, USA, 225-230. DOI: https://doi.org/10.1145/3017680.3017783.

[12] Wireshark Foundation. Wireshark · Go Deep. Retrieved from https://www.wireshark.org/.