CyberChallenge.IT@Unige

Ethical Hacking for Young Talents

Gaspare Ferraro
ferraro@gaspa.re

Giovanni Lagorio
DIBRIS, University of Genoa, Italy
giovanni.lagorio@unige.it

Marina Ribaudo
DIBRIS, University of Genoa, Italy
marina.ribaudo@unige.it

ABSTRACT

We present our experience as educators of small groups of students interested in cybersecurity and ethical hacking. Since 2018 we have been involved in a national cybersecurity training program whose primary goal is bringing talented young students to this field and lessen the gap between the available workforce and what the market demands. The training model exploits gamification principles, and our students apply their knowledge and skills in online competitions, playing in virtual arenas. These provide lawful environments to experiment with cybersecurity vulnerabilities, attacks, and defenses freely and legally. In this paper we first describe the national program we are involved in, and then we detail the activities taken at our university, with a special emphasis on this year edition that, due to the COVID-19 restrictions, is currently running entirely online.

CCS CONCEPTS

• **Security and privacy** → *Human and societal aspects of security and privacy*; • **Applied computing** → *Education*.

KEYWORDS

cybersecurity education, gamification, capture the flag

ACM Reference Format:
Gaspare Ferraro, Giovanni Lagorio, and Marina Ribaudo. 2020. CyberChallenge.IT@Unige: Ethical Hacking for Young Talents. In *Adjunct Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '20 Adjunct), July 14–17, 2020, Genoa, Italy.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3386392.3399311

1 INTRODUCTION

Digital technologies are pervasive and deeply embedded in everyone's daily activities. We communicate, study, buy goods and services, manage money, organize free time, and play through computers or mobile devices constantly connected to the Internet. For this reason, we should be aware of the dangers of such an always-connected life, which has permeated society in the last decade. If ten years ago we had forced to wear a tracking device we would have started a revolution. Instead, we bought a smartphone.

As a consequence, we cannot ignore the risks of being hyper-connected anymore. In this context, *cybersecurity training* comes into play, and educators have the opportunity to introduce a broad spectrum of topics to learners with different backgrounds. Everyone, not only the young generation of students, should be aware of risks, such as continuous profiling and loss of personal data, they face daily by having Internet in their pockets. Attending cybersecurity awareness courses, to understand correct and safe online behavior and how they can be good digital citizens, could be a first step. However, for computer science and engineering students, more in-depth technical training is needed to complement these cybersecurity literacy modules. Since the future technical workforce consists of these people, they should know how to design and implement high quality and secure systems, how to protect digital assets, detect vulnerabilities, and so on. Furthermore, the labor market asks educators to work in this direction, since the demand for cybersecurity professionals is high, with increasing shortfall[1]. Fortunately, this need has been recognized by governments and academic institutions. Indeed, many countries have launched different cybersecurity training programs not only to raise awareness but also to form a competent workforce.

We are involved in one of these national projects, called CyberChallenge.IT[2], which represents the *best practice* for cybersecurity training of young talents, aged from 16 to 23, in Italy. In this paper, we introduce our experience as educators of the local class at our university. We started in 2018, and we are currently running the third edition of the course. This year, due to the COVID-19 restrictions, the training is entirely online, and we needed to set up an online ecosystem of software tools, in a very short time, to make the training possible.

The paper is organized as follows. We first introduce in Section 2 the CyberChallenge.IT project and Capture the Flag competitions to make the context clear. In Section 3, we briefly discuss some other experiences we are aware of, and then, in Section 4, we present our training path, also describing the software platforms we adopted when moving from a face-to-face to an online course. Finally, Section 5 concludes this work with some suggestions for educators interested in putting to practice a similar experience.

2 CYBERCHALLENGE.IT

CyberChallenge.IT (CC.IT in the following) is the leading Italian initiative for introducing young talents to the field of cybersecurity. Annually organized by the National Cybersecurity Laboratory[3], the project is currently running its fourth edition.

UMAP '20 Adjunct, July 14–17, 2020, Genoa, Italy

ACM ISBN 978-1-4503-7950-2/20/07…$15.00
https://doi.org/10.1145/3386392.3399311

[1] https://cybersecurityventures.com/jobs/
[2] https://cyberchallenge.it/
[3] https://cybersecnatlab.it/

Table 1: CyberChallenge.IT participants over the four editions and Italian CTF teams

Year	Nodes	Involved students									CTF teams
		Booked							Enrolled		
		Total	Gender		Origin						
			M	F	High Schools		Universities				
		#	#	#	#	%	#	%	#	%	
2017	1	683	603	80	57	8.3	626	91.7	20	2.9	27
2018	8	1 866	1 698	168	583	31.2	1 283	68.8	160	8.6	40
2019	18	3 203	2 830	373	1 341	41.9	1 862	58.1	360	11.2	60
2020	28	4 452	3 848	604	1 960	44.0	2 492	56.0	560	12.5	n.a.

Its target is young people aged 16-23, and the 2020 edition has interested more than 4 400 of the best students who live and study in Italy. The project aims at creating, and then continually growing, a community of young *cyber-defenders*, that is, *ethical hackers*, to form the workforce of the years to come. Ethical hackers are skilled security experts who specialize in penetration testing and methodologies that ensure the security of organizations' information systems. The term *cyber-defender* puts particular emphasis on the fact that these skills should always be used to *defend* systems, legitimately and ethically.

Ethical hackers possess diversified technical skills, and, according to [2], standard computer science and computer engineering curricula lacked several topics for building such expertise. This shortage, in turn, led to unrealistic teaching environments creating false expectations in students when, after graduation, they joined professional fields, where security was vital to many companies.

The CC.IT project tries to fill this gap. It offers training opportunities to stimulate interest in STEM disciplines and, in particular, in information and computer security, keeping in mind also what happens outside universities. Participants have the opportunity to get in direct contact with IT companies working in the field, which actively contribute to their orientation and professional training.

Higher education institutions organize the training, and Table 1 shows the numbers of students and training nodes involved in the four editions of the project.[4] After the first pilot, organized in 2017 by professors Roberto Baldoni and Camil Demetrescu at Sapienza, the University of Rome, the number of training nodes has grown significantly, involving this year 27 Italian universities and 1 military academy.

The last row in the table shows the data of the 2020 edition: 4 452 students enrolled (86.5% male, 13.5% female), 44% of them coming from high schools, 56% coming from universities distributed throughout the country.

Registered students can train at their best for the admission, thanks to a custom in-house software platform[5] developed to meet the requirements of the CC.IT project. The project does not assume any prior cybersecurity knowledge, and participants are selected only on their logic, problem-solving, and programming abilities. Admission tests are both online, first, and then on-site, for those who passed the so-called, *pre-selection phase*. At the end of the admission phase, the top-ranking students join classes of 20 members, one for each training node, for a total of 560 learners in 2020, and start their training path, which lasts for 3 months. The course is organized in 12 weeks of training, with 2 hours of theory and 4 hours of hands-on per week, for a total of 72 hours. Notice that students add all these hours to the official courses, at their school or university.

Most of the training consists of technical cybersecurity topics, but without ever forgetting legal and ethical aspects. Moreover, sponsor companies organize seminars to share experiences and discuss real test cases.

Hands-on activities are the core of the training, and these guide students step-by-step in solving *Capture the Flag* challenges of increasing complexity. Capture The Flag competitions (simply CTFs in the following) are a special kind of information security competitions, which have been around for many years. They are regarded as an excellent means to acquire deeply technical concepts in a fun, non-traditional, learning environment. Many different types of CTFs exist, among them *Jeopardy* and *Attack/Defence* are those interesting for CC.IT.

- Jeopardy CTFs involve multiple categories of challenges, each of which contains vulnerabilities. Participants, often grouped into teams, must exploit these vulnerabilities to find hidden *flags*, that is, (unpredictable) strings in a given format. The knowledge of a flag proves that the corresponding challenge, to be precise, one or more of its vulnerabilities, has been successfully exploited. Participants (teams) do not directly attack each other. They enroll in online platforms, where they find the challenges and submit their flags to gain points. Competitions in this format allow students to think *adversarially*, i.e., to think as an attacker would, and this form of gamification motivates them to learn by doing.
- In Attack/Defense CTFs, teams run an identical machine, or a small network, injected with vulnerable services. In this case, the goal of each team is to find and exploit the vulnerabilities in opponent' machines, while fixing or mitigating flaws in their own. Compromising a machine enables a team to acquire hidden flags. Note that, differently from Jeopardy CTFs, in this case, flags change during the event because a *scoring bot* service updates them regularly, and teams lose points if their services are not up when the scoring bot contacts them. That is, *availability* of services or SLA (service level agreement) plays an essential role in calculating the final score.

[4]Data extracted from: https://cyberchallenge.it/assets/CCIT20-sito.pdf.
[5]https://training.cyberchallenge.it/

> Usually, teams have a couple of hours to understand the playing scenario before the competition really starts. Although Attack/Defense CTFs are more demanding to play, they allow participants to gain experience with both *offensive* and *defensive* related skills.

In all types of competitions, there is also a follow-up phase dedicated to the publication of *write-ups*. Write-ups are short descriptions of how a challenge could be solved, usually written by those who solved it during the contest.

From an educational point of view, this is extremely useful since, on the one hand, it allows participants to arrange and summarize the steps towards their solutions and, on the other hand, it allows to compare different techniques, chosen by different people, to face the same problem. Write-ups are even more useful for those who did not succeed in solving some exercises since they can, a-posteriori, find hints valuable for future competitions.

The training process of CC.IT ends with two final competitions. The former is a Jeopardy-style CTF, run concurrently by all the attendees at their training nodes. The result of such a CTF helps select the four-member teams, from the 20-people classes of each node, that compete in the following national competition. The latter is the Italian CTF championship in Cybersecurity, an Attack/Defense-style CTF, organized each year in a different location. During this event, all the teams, their instructors, and the sponsors meet to celebrate and conclude the entire project.

As already said, the project started four years ago, and it offered training opportunities to many young students, thus promoting cybersecurity training at the national level, and posing an initial seed to help the country in forming the future generation of professionals in the field. Moreover, we can observe that, in recent years, more and more strongly motivated students joined CTF teams and take part in the many online events advertised on CTFtime[6], as witnessed by the increasing number of Italian teams enrolled to the platform (see the last column of Table 1). We like to think that CC.IT contributed, at least partially, to the growth of these numbers: from the 4-6 teams of the initial years (2011-2014), the 10 and 13 teams of 2015 and 2016, respectively, we can observe that, in 2019, 60 teams played at least one online competition. This number (60) is 6 times the number of teams recorded in 2015.

3 RELATED WORK

The USENIX Workshops on Advances in Security Education are an important venue to share educational experiences in the field of cybersecurity. Indeed, some workshops' papers introduce several case-studies on educational activities that allow students to understand critical concepts of cybersecurity threats and to improve their skills and abilities to prevent cyber attacks. In many cases, gamification methodologies and techniques are selected to present cybersecurity scenarios, asking students to find possible solutions.

For instance, [6] reports on a 10-week experience during which students played an Alternate Reality Game, presented with a realistic narrative: *"The daughter of a student expelled 20 years ago is back to her father's campus to avenge him, and her initial point of attack is the website of a security course…"*. A goal of the experience was to understand key concepts of cybersecurity threats and to improve students' skills and abilities to prevent cyber attacks. Results show that after the course, students positively changed their perception of the cybersecurity profession, in terms of understanding the tasks and problems that need to be solved.

Also, the paper [3] proposes an experiment based on gamification. During an 11-week cybersecurity course, students played the role of newly hired IT security employees in charge of different tasks, presented as CTF-like exercises. Each exercise offers the chance to choose different options for advancing into the plot of the game. Depending on what the students decide, the plot evolves, and changes accordingly. Authors state that those students who actively followed the narration offered by the game scored better, as opposed to those who ignored the suggestions.

Švábenský et al. present in [10] the results of two interrelated undergraduate courses. The first one focuses on the basics of offensive cybersecurity, and the second one, more advanced, requires to design a gamified tutorial. The learner has to secure a service against automatic attacks running in the interactive virtual environment of their cyber range. The experiment is publicly presented to peers during the University Open Day, and the students gain an authentic experience of working with a real audience to which they have to present their projects. Students rated the courses positively since they exercised adversary thinking in real-world settings.

The same research group published in [11] the results of an extensive survey in which they examined 71 papers on cybersecurity education published at the ACM SIGCSE and ACM ITiCSE conferences. They identify the most common topics covered in the papers, among them, secure programming, network security and monitoring, cyber attacks, malware, hacking, exploitation, and cryptography. They also identify the most prominent target group for teaching interventions, which are university students (undergraduates or graduates) followed by instructors and educational managers, K-12 students (middle or high school). The paper also analyses the most common teaching and evaluation methods employed to check students' performance, the sample sizes, and the availability of public datasets. Its extensive bibliography is a good starting point to read about cybersecurity education experiences.

The majority of the examples we found in the literature describe official university courses that provide a mix of theory and hands-on activities. CC.IT is different, since the training is not included in any official course, and students have different ages and backgrounds. Their goal is to learn first, but also to enjoy and possibly to obtain a satisfactory result to enter the top positions at the Jeopardy CTF and be selected for the Attack/Defence competition.

In 2014, ENISA[7] started to organize a yearly European Cyber Security Challenge[8] (ECSC), which is an initiative that aims at enhancing cybersecurity talent across Europe and connecting high potentials with industry-leading organizations. ECSC is the occasion to meet the national teams of several countries, competing against each other to establish which country has the best cyber talents.

Each year, the best CC.IT students join the Italian national team, and all the other nations form their teams as well, by organizing

[6] This is the starting point for any CTF player: CTFs are weekly announced on the website, and all teams are registered there, see https://ctftime.org/

[7] European Union Agency for Cybersecurity, https://www.enisa.europa.eu/

[8] https://europeancybersecuritychallenge.eu/

events, hacking camps, or CTF games. To the best of our knowledge, as also written in [12], Italy has the most systematic and comprehensive model for selecting and training the national team. Some information on the selection processes of the other nations can be found by following the links available on the corresponding region on the interactive map of the ECSC website.

4 TRAINING EXPERIENCE

In this section, we introduce our training experience. We briefly describe the on-site admission phase and show some data on the students involved in our university (Section 4.1), how we organized the teaching in the 2020 edition (Section 4.2), and some results of the previous editions (Section 4.3).

4.1 Admission

The on-site admission phase does not assume any prior cybersecurity knowledge. It consists of (1) one quiz with multiple choice questions on logic and problem solving, and (2) a programming test reserved to those students who ranked in the top-k positions[9] at the quiz. The programming test lasts for three hours, with three proposed exercises, at different levels of difficulty.

To be able to manage the high number of participants, the 2020 on-site admission phase used a customized version of the software CMS [5], the acronym of Contest Management System[10], a popular open-source grading system often used for competitive programming contests developed by the team organizing the Italian Olympiad in Informatics[11]. CMS provides a web-based interface to submit students' solutions, which are graded as they are submitted without human intervention. The languages supported in the admission phase are C/C++, Java, and Python.

The automatic evaluation of the solutions is a real improvement, to what happened during previous editions, since it relieved the instructors from manual evaluation, which required days in past years, and is no longer sustainable with the current number of participants. On the other hand, manual evaluation permits to take into account some aspects, like code clarity and good ideas poorly implemented, that cannot be automatically evaluated.

This year, each exercise had a score from 0 to 100, depending on various factors, e.g., the number of passed tests or execution time. The results of the programming phase and the quiz are averaged to form a final score. Students are then ranked, and the top-20 in each node form the corresponding class.

Table 2 shows the number of students who applied for CC.IT at our university. We can observe a decrease in the total number of registered participants from 2018 to 2020, and we motivate this trend with the fact that the training is objectively demanding: after the novelty of the first year, now students know this, therefore only motivated students enroll. For the same reason, we also observe fewer young students from high schools (from 65% of 2018 to 40% of 2020).

The percentage of female students (8% in 2018, 4% in 2019, 11% in 2020) does not show any trend. However, it highlights the same gender problem Europe faces with STEM disciplines[7], confirmed in the context of cybersecurity. Finally, in 2019 our class was formed by 16 students only (see column Enrolled in the table) because, on the day of the on-site admission (which is the same for all training nodes), there was a forecast alert, so only 35 students could reach the university.

Table 3 shows some details about the classes formed after the admission. Few female students reached top positions, only 2 in 2018 and 2020. We never attracted students aged 19; this is probably explained by the fact that students at such an age are in the last year of high school, and they need to prepare for their final exam (whose dates overlaps the ones of CC.IT finals). The average age of this year's cohort is higher than the previous two since only two students from high school entered the top-20 positions.

We end this section by briefly describing the three programming exercises of the 2020 edition, to give an idea of the skills that are verified before starting the training period:

- **Postfix**, the first and easiest exercise, asks students to write a program that, given a postfix expression, evaluates it and returns the result.
- **MaxOfMin**, the second exercise, of medium difficulty, tests the knowledge of array manipulation with the following problem: given an array of n integer numbers, find, for each size between 1 and n, the maximum of the minimum's of every contiguous sub-sequence in the array.
- **Polynomials**, the third exercise, poses a mathematical question: given an integer coefficient polynomial $p(x)$ such that $p(2) \neq 0$, count in how many ways it is possible to change a coefficient, in a given numeric range, to obtain $p(2) = 0$.

The solutions of the proposed programming exercises are relatively easy to code, as they do not need any particular programming trick. The first exercise, *Postfix*, requires a basic knowledge of stack abstract data type. The second one, *MaxOfMin*, has a lot of different solutions, and it tests students' algorithmic skills. Finally, the third one, *Polynomials*, needs some basic math knowledge or can be solved by (partially) using a brute-force method, and both these skills are instrumental in solving cryptography challenges.

4.2 Teaching

After the admission phase, each node starts teaching independently. That is, while the contents are the same among all nodes, timetables typically differ. As already mentioned in Section 2, CC.IT lectures and hand-on activities span over 12 weeks, for 6 hours per week. It is an intensive period of hard work during which instructors also encourage team building since collaboration is essential to solving complex tasks and challenges.

The current edition started in March 2020, during the lockdown due to the COVID-19 emergency. Therefore, after the first moment of loss, we organized ourselves to switch to distance learning. The first step was the selection of the appropriate software platforms for organizing lectures and hands-on activities.

After a bit of experimentation with some other tools, we decided to settle with Microsoft Teams[12], among Wooclap[13] and Telegram[14].

[9] The number k varies from one node to another, depending on the number of computers available in the labs. We had $k = 50$.
[10] https://github.com/cms-dev/cms
[11] https://olimpiadi-informatica.it/
[12] https://products.office.com/en-US/microsoft-teams/group-chat-software
[13] https://www.wooclap.com/
[14] https://telegram.org/

Table 2: Unige participants over the three editions of CyberChallenge.IT

Year	Involved students								
	Booked							Enrolled	
	Total	Gender		Origin					
		M	F	High Schools		Universities			
	#	#	#	#	%	#	%	#	%
2018	154	142	12	100	64.94	51	35.06	20	12.98
2019	94	90	4	53	56.38	40	43.61	16	17.02
2020	82	73	9	33	40.24	46	59.75	20	24.39

Table 3: Statistics of the admitted students

		Sex		Origin		Age distribution								
Year	#	M	F	School	University	16	17	18	19	20	21	22	23	Avg
2018	20	18	2	7	13	0	2	5	0	3	4	6	1	20
2019	16	16	0	7	9	4	2	1	0	2	5	2	0	19
2020	20	18	2	2	18	1	1	0	0	5	4	4	5	21

Microsoft Teams is a chat-based collaboration tool that provides remote participants with the ability to work together and share information via a shared space. Our university provides free access to the platform to all faculties and students. It allows audio/video communication, document sharing, and above all, screen sharing. This last requirement is a must for us to show the students how to use the various tools and approach the solving of cybersecurity challenges. Lectures are streamed and recorded, and we use Wooclap's instant polls to get real-time feedback from students. In addition to Teams and Wooclap, used during lectures, we also set up two Telegram channels, one for quick communication of official announces (basically "read-only" for students) and an unofficial one, for general discussion and team building.

All teaching material (e.g., slides, video lessons, and challenges) are weekly published on a CTFd[15] instance, expressly customized for the project. CTFd is one of the most popular framework to run CTF competitions: it allows organizers to publish their challenges, and participants to solve them, submit the associated flags, get the corresponding scores, and check the scoreboard. In order to manage the different classes of CC.IT, a custom permission management has been implemented. So, each participant can see only the scoreboard of its local class, and instructors can access reserved material, such as the challenge write-ups, which are not visible to students.

A typical week consists of a 2-hour online lecture that we deliver on a dedicated channel on Microsoft Teams, to introduce a new topic, typically. This lecture is then followed, on another day, by a 4-hour online hands-on training. During these hands-on sessions, useful tools are introduced and demoed. Then, some CTF-style challenges are first presented, and then tackled together. Microsoft Teams allows the course's owners to open multiple channels, in which students can meet to work in small groups, sharing their screens and via text/voice chats. Members of the local CTF team, Zenhack[16], support the students by giving help, and sharing tips, via 1-1 calls or in separate Teams channels. Facilitators take actions on-demand, that is, when someone asks explicitly for help, or when they see someone stuck. In both cases, they explain how to approach the problem, without giving the full solution. We like to notice that many of them were, in turn, CC.IT students in one of the previous editions, and they now continue to study and practice cybersecurity with us. This fact is probably one of the most concrete signs that we are on the right track in building a community.

The first week of training is introductory, and its goal is twofold. On the one hand, it presents the CC.IT project and CTF competitions in general. On the other hand, it presents some legal and ethical issues related to cybersecurity, privacy, and data protection, outlining the "limits" a cyber-defender should never trespass.

Some trivial challenges are also proposed, to give students a taste of what to expect, and to test flag submission on the platform. These are relatively easy exercises, called *warm-up*, and come with no explanations. Students are encouraged to search on the Internet the necessary hints to understand the problem and find a solution. Indeed, they must understand from the very beginning that it is not possible to know everything, and learning how to search for information is an essential skill on its own. Online materials, papers, software documentation, and likewise are vital to fill technical or theoretical gaps.

Warm-up exercises cover some basic encoding/decoding techniques, for example, Base64[17], image manipulation to find hidden text, and some web basics. In one challenge, students need to change the value of a cookie released by a web page to bypass an authentication check and get the flag back. This shows how information stored on the client can be easily manipulated and cannot be used for security checks, unless the server can validate their integrity (for instance, by using a digital signature scheme). In another challenge, they need to inspect the `robots.txt` file to find a document that web crawlers cannot index, which contains another flag. Online websites, such as the *Bandit* wargame[18] are also suggested. Bandit

[15] https://ctfd.io/
[16] https://zenhack.it/
[17] https://en.wikipedia.org/wiki/Base64
[18] https://overthewire.org/wargames/bandit/

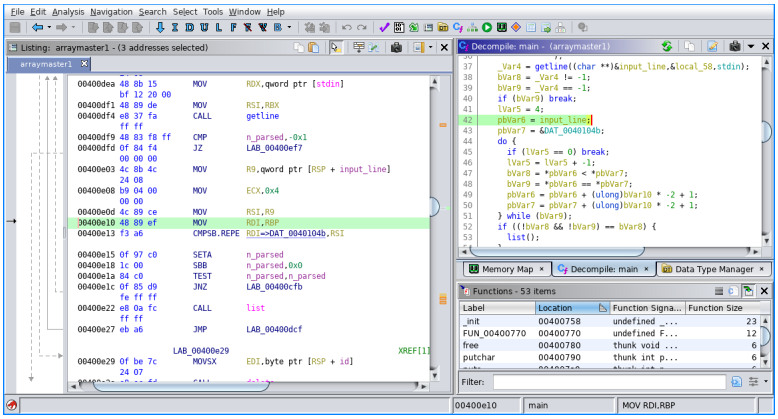

Figure 1: Ghidra GUI

provides exercises for beginners, which can be accessed sequentially: the solution of each level reveals the password to access the next one. By solving these exercises, students acquire basic knowledge on shell commands, which are fundamental to tackle more complex exercises.

After the warm-up phase, the technical lessons begin, and *the going gets tough*, as the saying goes. To make things more concrete, we now detail the topics of the weeks dedicated to software security.

4.2.1 Software security syllabus. The purpose of introducing software security is to make students aware that poorly written programs can be exploited for malicious purposes, making them act in unintended ways. Software security is a vast topic, of course, so ours is just an overview, which spans over three weeks.

During the first week, we briefly discuss the compilation-linking process, explaining the ELF file format and dissecting some simple examples using tools like hex-viewers and binary parsers; for instance, *Katai Struct*[19] is particularly flexible. Then, we discuss platforms and *ABI*s, Application Binary Interfaces, focusing on calling conventions in the x86/x64 world. The role of an operating system, and its *syscall* interface is recalled. We find particularly helpful Godbolt's *Compiler Explorer*[20] to show how various language constructs get translated into machine code, with various level of optimizations.

In order to start reverse engineer some program, we introduce some static and dynamic analyses. For static analysis we mainly used *Ghidra*[21], a software reverse engineering suite of tools developed by NSA. Ghidra offers a variety of tools, in particular a *decompiler*, that is a great help in introducing students to reverse engineering of binaries, without requiring too much familiarity with the x86/x64 instruction set, even though we cannot dismiss that subject altogether. Indeed, for historical reasons, x86/x64 instruction set is very complex, and so is its disassembly [1, 4]. Figure 1 shows the main GUI of Ghidra; on the left, we can see the x86 assembler listing, on the right the corresponding decompiled function and the list of functions. This example screenshot covers only the main features; Ghidra, and its GUI, are vast and incredibly customizable.

For dynamic analysis, debugging in particular, we mainly use *GDB*[22], the GNU Project debugger, enhanced by *GEF*[23], the GDB Enhanced Features for exploit developers and reversers.

During the second week of software security, we introduce memory corruption attacks; for a thorough survey see [9].

We analyze stack buffer overflows in-depth, and then we develop some exploits in a '90 settings, that is, disabling all modern mitigations (ASLR, NX, and stack canaries), that are treated during the third week. To develop the exploits, we leverage on Python and, in particular, the pwntools framework[24]. Seeing how the control-flow of a program can be "easily" hijacked is both surprising and very instructive for students. In the last settlement of software security, we discuss the various forms of mitigation that are currently in place, in modern compilers and operating systems, and how they can be bypassed under certain circumstances. For instance, we discuss ROP [8] and the use of *Ropper*[25] to automatically find gadgets and generate ROP-chains.

We must notice that the technical program is broad, and one of the main challenges for educators is not to lose anyone behind: the path is demanding, but it is important to follow it all together, regardless of the age and the previous background. That is, forming a cohesive team is more important than reaching the end of training with a small bunch of technically savvy "survivors".

4.3 Results

This section shows some results of our classes during the three editions of CC.IT. The data of this year are only partial since, at the time of writing, the project is still running: lectures will finish at the end of May, and the Jeopardy CTF will be played online on June 8, 2020.

Table 4 compares the ranking obtained by the students after the admission phase, and after the training. For each edition, the row labeled A represents the position of each student after the admission, and the row labeled J represents the position after the Jeopardy CTF. The row labeled Δ shows the distance between these two numbers:

[19] https://kaitai.io/
[20] https://gcc.godbolt.org/
[21] https://ghidra-sre.org/
[22] https://www.gnu.org/software/gdb/
[23] https://github.com/hugsy/gef
[24] https://github.com/Gallopsled/pwntools
[25] https://github.com/sashs/Ropper

Table 4: Admission (A) vs Jeopardy CTF (J) ranking comparison

2018	A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	J	**4**	**2**	10	x	7	**1**	15	9	**3**	13	6	12	19	14	11	17	5	18	8	16
	Δ	-3	0	-7	x	-2	**+5**	-8	-1	**+6**	-3	**+5**	0	-6	0	-4	-1	**+12**	0	**+11**	**+4**
2019	A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16				
	J	**2**	x	**4**	**1**	5	**3**	14	15	9	10	8	13	12	7	6	11				
	Δ	-1	x	-1	**+3**	0	**+3**	-7	-7	0	0	**+3**	-1	**+1**	**+7**	**+9**	**+5**				
2020	A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	J*	6	9	16	7	2	17	1	5	14	11	3	15	13	18	8	10	19	12	20	4
	Δ*	-5	-7	-13	-3	**+3**	-11	**+6**	**+3**	-5	-1	**+8**	-3	0	-4	**+7**	**+6**	-2	**+6**	-1	**+16**

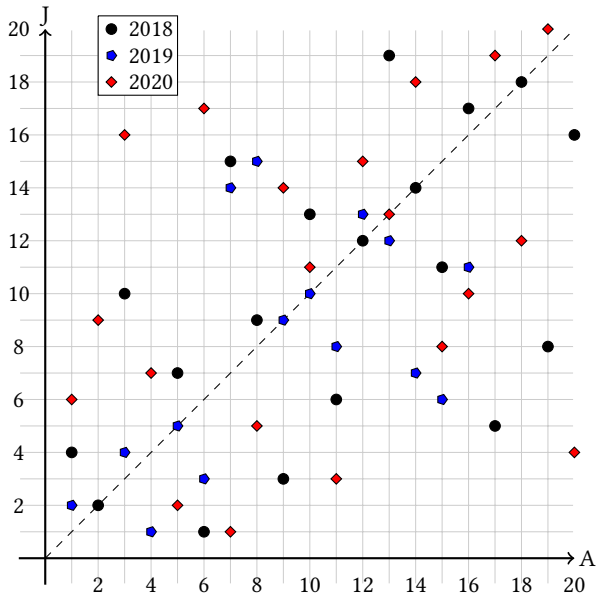

Figure 2: Admission vs Jeopardy CTF ranking comparison

positive numbers represent an improved position compared to the admission, negative numbers represent a worsening of the position. Students who maintained their positions have an associated value equal to 0; students who did not attend the Jeopardy CTF have an associated value equal to x. For the edition 2020, the current ranking (rows labeled $J*$ and $\Delta*$) is taken from the scoreboard of the CTFd platform used during the training, and therefore it is only partial and might change after the final.

Figure 2 shows, graphically, the same numbers, using different symbols for the different editions: black circles for 2018, blue pentagons for 2019, and red diamonds for 2020. Elements on the main diagonal $A = J$ represent students who did not change their positions; below the diagonal, we have those students who improved their performance after the training ($J < A$), and above the diagonal, those who did not perform so well ($J > A$).

The team who played the national competition for our university in 2018 (black circles with $J \leq 4$) consisted of two students, who already ranked in the top-4 positions after the admission phase, and two other students who jumped forward at the Jeopardy CTF. In 2019 (blue pentagons with $J \leq 4$) the situation was more stable, with fewer jumps forwards (see also boldface numbers in the rows labeled J in Table 4). No other trend can be observed: few students kept their position after the training, and we think this might be explained by the fact that programming and ethical hacking skills are very different.

CC.IT, building upon logic and programming skills, covers a broader content, which may not be congenial to all admitted students. Programming ability is a necessary but not sufficient condition to become a good ethical hacker and a CTF player, since other skills, for instance, lateral thinking, are needed.

The two teams, who played the national Attack/Defence competitions for our university, consisted of university students only. None of the youngest students, coming from high school, ranked in the top-4 positions at the Jeopardy CTF. This can be explained by observing the smaller number of students aged 16-19 admitted to CC.IT, and also by the fact that coding and programming are subjects taught at school, while other subjects, for instance, operating systems and networks, are sometimes totally new to them. It is impossible to become cybersecurity "experts" in one semester, and the project aims at *attracting* young people to this field and motivating them to continue their studies in this direction to build their professional profile. In order not to demoralize younger students, starting from the 2020 edition, it was decided that it is possible to enroll twice to the project, when students change their "role", i.e., when they finish high school and enter university.

5 CONCLUSION

We presented our experience as educators in an Italian national project, CC.IT, aiming at promoting cybersecurity among young talents. We are currently in its third edition, and the activities this year got more complicated because of the COVID-19 emergency. However, thanks to the adoption of appropriate tools for distance learning, both lectures and hands-on activities are up and running.

Every year, during CC.IT training, considerable effort is spent in trying to promote team building: the challenges, students have to solve, require diversified skills, which are difficult to find in a single individual. Working in a group, following different solution strategies, listening to others' opinions is sometimes the only way to solve a complicated task. In the current edition, this critical part of the educational path is more difficult to achieve due to the forced distance between the participants. However, we noticed constant participation in the online lectures and, after the first two weeks,

we also observed more interaction among the instructors and the class, and among the students themselves, even though none of us ever met in person.

Moreover, some of the more motivated students, once they acquired some initial skills, enrolled in online CTFs, specifically organized for beginners. This active approach allowed them to keep learning beyond the hours provided by the project. These episodes are a point in favor of the activity we are carrying out: students are motivated, they appreciate the covered topics, and they are committed despite the objective difficulties of this period. And it is also a source of satisfaction for us, who dedicate many hours to CC.IT, outside our institutional teaching load.

Of course, the project also has some weaknesses, at the organizational and educational level. The high number of training nodes, and registered students, highlighted some scalability problems, which have been partially addressed.

The software platform used to automate the scoring of submissions during the admission phase was a great addition, but it needs some improvements. For example, there was no possibility of withdrawing early from the test. Given the distributed nature of the admission phase, which is like an exam, but with thousands of students spread in dozens of labs around the country, it is essential to let each node manage some issues locally, as the entrances and exits from the labs. This functionality, which is implemented for the next edition, was missing, and a large number of messages was exchanged between universities and the central organization.

A lot of effort has gone in preparing the syllabus and support materials (slides and accompanying videos, plus challenges and their detailed write-ups) for the CC.IT project. However, notwithstanding this enormous work, each node has its peculiarities, strengths, and weaknesses. For instance, some local instructor might be an expert of, say, web security, but not so skilled in cryptography. And, while the supporting material helps, it takes dedication and a lot of work to prepare engaging and technically adequate lectures on such disparate topics. Furthermore, we cover a lot of ground in (relatively) few weeks. It is challenging to find the right amount of theory versus practice. On the one hand, with too much theory then students can not tackle any practical challenge. On the other hand, you cannot understand how to use the tools without a reasonable background effectively. So, choosing the specific topic to deepen is a work still in progress and, in some sense, a moving target, which depends on the class and its varied background.

This diversity among topics is another reason why creating a local community pays back in the long run. The nodes that have an active CTF team, in our case *ZenHack*, can count on its members for supporting the new students and explain some rather specific topic or technique. Their experience is invaluable, and it makes the difference in bridging the gap between theory and practice.

As we discussed in Section 4.3, selecting the students with the right mindset and skills is very though. While some programming abilities are necessary, excelling in programming challenges does not imply excelling in cybersecurity ones. There are some apparent overlaps, but the disciplines have their peculiarities and, without passion, one cannot excel in any of them. So, in these years, we witnessed some programming talents fail miserably on CTF challenges, and, vice versa, some mediocre programmers excel in solving cybersecurity problems.

Cybersecurity is currently a hot topic, and young students are easily attracted and they *think* they love it, maybe thanks to videogames and TV series, but many of them give up when the technical part gets tough. Unfortunately, we do not have found a perfect recipe yet, and may only caution in trying to be too selective on the (a-posteriori) wrong skillsets.

We conclude this paper with a personal suggestion for those interested in following a similar experience. The project is exciting but - especially in the first year of adoption - it is also very demanding. The topic selection is rather broad, and it is tough for a single educator to cover all of them satisfactorily. Group collaboration is also needed from the teachers' perspective, and things become easier only in the following years if a community of young talents grows and helps the new entries, year after year. Without this help, it can become very frustrating and, honestly, pointless.

6 ACKNOWLEDGMENTS

The training activities would not have been possible without the cooperation of the ZenHack CTF team. We thank its members for their support and enthusiasm.

REFERENCES

[1] Dennis Andriesse, Xi Chen, Victor van der Veen, Asia Slowinska, and Herbert Bos. 2016. An In-Depth Analysis of Disassembly on Full-Scale x86/x64 Binaries. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 583–600.

[2] S. Bratus. 2007. What Hackers Learn that the Rest of Us Don't: Notes on Hacker Curriculum. *IEEE Security Privacy* 5, 4 (July 2007), 72–75. https://doi.org/10.1109/MSP.2007.101

[3] Tom Chothia, Sam Holdcroft, Andreea-Ina Radu, and Richard J. Thomas. 2017. Jail, Hero or Drug Lord? Turning a Cyber Security Course Into an 11 Week Choose Your Own Adventure Story. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*. USENIX Association, Vancouver, BC.

[4] C. Jämthagen, P. Lantz, and M. Hell. 2013. A new instruction overlapping technique for anti-disassembly and obfuscation of x86 binaries. In *2013 Workshop on Anti-malware Testing Research*. IEEE, Montreal, QC, Canada, 1–9.

[5] S. Maggiolo, G. Mascellani, and L. Wehrstedt. 2014. CMS: A growing grading system. In *Olympiads in Informatics*, Vol. 8. Vilnius University, 123–131.

[6] John R. Morelock and Zachary Peterson. 2018. Authenticity, Ethicality, and Motivation: A Formal Evaluation of a 10-week Computer Security Alternate Reality Game for CS Undergraduates. In *2018 USENIX Workshop on Advances in Security Education (ASE 18)*. USENIX Association, Baltimore, MD.

[7] Patricia C. Salinas and Claudia Bagni. 2017. Gender Equality from a European Perspective: Myth and Reality. *Neuron* 96, 4 (2017), 721–729. https://doi.org/10.1016/j.neuron.2017.10.002

[8] Hovav Shacham. 2007. The Geometry of Innocent Flesh on the Bone: Return-into-Libc without Function Calls (on the X86). In *Proceedings of the 14th ACM Conference on Computer and Communications Security* (Alexandria, Virginia, USA) *(CCS '07)*. Association for Computing Machinery, New York, NY, USA, 552–561. https://doi.org/10.1145/1315245.1315313

[9] Laszlo Szekeres, Mathias Payer, Tao Wei, and Dawn Song. 2013. Eternal War in Memory. *IEEE Security & Privacy* 12, 48–62. https://doi.org/10.1109/SP.2013.13

[10] Valdemar Švábenský, Jan Vykopal, Milan Cermak, and Martin Laštovička. 2018. Enhancing Cybersecurity Skills by Creating Serious Games. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) *(ITiCSE 2018)*. Association for Computing Machinery, New York, NY, USA, 194–199. https://doi.org/10.1145/3197091.3197123

[11] Valdemar Švábenský, Jan Vykopal, and Pavel Čeleda. 2020. What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, USA, 2–8. https://doi.org/10.1145/3328778.3366816

[12] Muhammad Mudassar Yamin, Basel Katt, and Espen Torseth. 2020. Review of Training and Selection Processes for European Cyber Security Challenge 2019. Internal report.