George Litvinov
Topics in SWE
Spring 2022

Topics in Software Engineering Paper:

Teaching Security – The Practical Way

Abstract

In the field of cybersecurity, theoretical knowledge is just as important as practical skill. While theory-centric fundamental security courses graduate students that can adapt to the changing technology landscape, and are able to generate on-paper solutions for the emerging security problems, they lack practical prowess that is so often required to understand these theoretical concepts on a deeper level as well as analyze and implement their own solutions. This paper is an attempt at gauging the student body's, as well as professor's opinion on combining the practical skill aspect of security Capture The Flag competitions into a traditional theory-centric fundamental cybersecurity course.

Introduction

Cybersecurity is a rapidly growing field. The U.S. Bureau of Labor Statistics estimates that the percent change in employment for Information Security workers will increase by 33% between 2020 and 2030 (US Bureau of Labor Statistics, US Department of Labor). This number is astonishing compared to the average 13% increase for all Computer occupations and average 8% increase for all occupations. Furthermore, there was a 30% increase in the cybersecurity workforce from 2020 to 2021 (ISC2). From this data it is clear that cybersecurity education is very sought after, while the field is becoming more and more competitive.

Cybersecurity is unique in the sense that it is adversarial. There is a constant cat and mouse game between the defenders and the attackers. Defenders try to create innovative defenses against the new and creative techniques the attackers use to bypass them. The consistent clash of minds makes the field highly competitive, innovative and practical while covering a wide range of Computer Science areas (Mirkovic and Peterson).

While security courses taught by universities serve the very important role of teaching theory to students in order to prepare them to apply this knowledge in designing security systems (Yurcik and Doss), the cybersecurity field is highly dependent on practical skill and the application of the theoretic concepts to arising situations (Bishop). Therefore teaching a theoretical as well as practical knowledge base in classrooms is important.

In this paper I am attempting to show that a basic cybersecurity course should be considered an important part of a Computer Science degree, and that a practical component is necessary for a foundational cybersecurity course. Furthermore, I will attempt to argue that the embedding of the competitive Capture The Flag (CTF) contest format into the curriculum of a foundational security course will increase both student material retention and enjoyment. Lastly, I will

George Litvinov
Topics in SWE
Spring 2022

describe why the CTF format is a perfect candidate to be the practical component of *COMS 4181 Security I* (Bellovin), which is Columbia University's foundational cybersecurity course.

Motivation

I am currently pursuing a Master's degree in Computer Science from Columbia University's Engineering School – the same school where I had received my Bachelor's degree which was also in Computer Science. I have been interested in cybersecurity since my sophomore year, and have explored the vast majority of the security curriculum the university has to offer. The biggest thing that stood out to me was that although I have received a strong theoretical foundation in security through coursework, the practical skills taught in the majority of these courses are insufficient.

One of the only courses that taught a strong practical foundation alongside theoretical components was *COMS 4186 Malware Analysis*. The course forced students out of their comfort zone week after week by pushing them to analyze live malware samples through a multitude of taught techniques. The highly practical component of the course is the reason it stands out to me as a highlight in my Columbia education.

Majority of all other courses at the university focused on developing a strong theoretical background as well as a problem-solving mindset. This way of teaching has its advantages as it allows students to adapt to new technologies and feel comfortable with any posed theoretical or system design problem. Although teaching a problem-solving mindset and theoretical background are important for someone that would want to continue their professional journey in security, the development of practical skill cannot be left behind. At Columbia, although we had learned that some cryptographic systems are insecure, we never explored how to actually break them beyond a theoretical explanation; we learned about memory safety, but never implemented a buffer overflow or ROP chain attack; we were introduced to the idea of SQL injections but didn't attempt performing one.

In order to gain practical skill, I had to go beyond what was taught in the classroom and find other resources. I had found that CTF competitions were a great way to gain practical prowess while also learning about new technologies. However, these events often pose a great barrier of entry and require a significant time investment in order to be competitive. It is not rare that newcomers, even if they have strong theoretical backgrounds, are only able to solve a couple out of tens of challenges.

My experience as a student and cybersecurity enthusiast in the past four years has motivated me to attempt to combine advantages of theory-centric coursework with the advantages of highly practical security competitions into one foundational security course that will leave students with a strong theoretical and practical background.

George Litvinov
Topics in SWE
Spring 2022

Background

Currently, Columbia University's foundational security course is *COMS 4181 Security I* (Bellovin). It is taught after the book *Thinking Security: Stopping Next Year's Hackers* (Bellovin), which does not have any practical exercises included in it. Rather, according to the Syllabus the students performed a semester-long project that was composed of a couple modules. The project requires a very large amount of non-security related programming from the students and a deep familiarity with C++. While such an approach has the benefit of shaping the students to be strong programmers, the related security concepts may get lost as lines of code written grows.

There are a number of courses at Columbia that have a practical-first approach while retaining the benefits of a theory-centric course – *COMS 3157 Advanced Programming* (Lee) and *CSEE 4868 System-on-Chip Platforms* (Carloni). The instructor of one course saw that students that had taken the practical-first *Advanced Programming* course performed better than students from previous semesters in an advanced course that requires a strong practical foundation, increasing the average GPA from 2.99 to 3.14 (Lee et al.). The other, focused on hands-on projects that would further the theoretical knowledge while providing a broad practical foundation on the topic of heterogeneous computing (Carloni et al.).

Instead of simply encompassing practical exercises that cover the theoretical concepts in a lab-like manner, the addition of gamification to courses has been shown to better active learning experiences in higher education (Murillo-Zamorano et al.). By introducing a CTF-like structure to a foundational cybersecurity course's practical components we can attempt to achieve exactly that. CTF, short for Capture The Flag, is a kind of cybersecurity competition that challenges individual contestants or teams to solve a variety of security-related tasks in hopes to score points. There are two types of these competitions – Jeopardy-style and Attack-Defense. Jeopardy-style CTF competitions present contestants with a variety of challenges they may solve, usually split into categories. The point-value depends on the difficulty level of the challenge, but also may depend on how many competitors or teams have solved them, decreasing the point value as more solves are registered. For Attack-Defense competitions each team is usually given access to multiple services that may have bugs. These services are the same across teams, and each team is responsible for patching their own services, while exploiting other teams'. A team earns points both for capturing specific generated strings from other teams, as well as for keeping their own services up and running. These competitions are often organized by private, school, government and company entities, such as DEFCON hosting the DEFCON CTF ("DEFCON"), the Department of Energy hosting the Cyberforce Competition ("Cyberforce"), and NYU hosting CSAW ("CSAW"). Attack-Defense competitions tend to have a very steep learning curve associated with them, while the difficulty of Jeopardy-style competitions depends on the posed challenges in it. This makes a Jeopardy-style CTF more suitable as a practical component for a theory-centric course.

There has been other work performed that is related to applying CTF competitions to a theory-centric security course. McDaniel created a CTF to introduce security topics to high

school students, but it is in no way an application of the concept of CTFs to a university-level security fundamentals course (McDaniel). However, the paper did share the mindset of transmitting security educational material through gamification.

Another related publication described the concept of a CCTF – Class Capture The Flag – competitions. In that paper, the competition was described to be a series of beginner friendly team-vs-team playoff-style Attack Defense events held throughout the semester (Mirkovic et al.). In this paper, I would like to argue that a Jeopardy-style CTF challenge would be a great practical addition to a theory-centric course, rather than one or multiple Attack-Defense competitions. That is due to the live nature of Attack-Defense style competitions. Usually, one gets only a small amount of downtime between competition “ticks”, which is when the connection to all other teams’ machines is enabled. Hosting one live event where students may team up and solve challenges would be great to promote the competitive nature of CTF events is an idea similar to the way *Advanced Programming* hosts a hackathon during the semester (Lee et al.). However, due to students having busy and different schedules, multiple live events may be difficult to organize throughout the semester.

More related work has been done in the creation of Web Labs for the purposes of teaching fundamental cybersecurity concepts (Schweitzer and Boleng). The US Airforce Academy has explored the usage of at-home labs that are accessible through the internet as an additional source to enroot security concepts in their students. They have found that these easily accessible and guiding labs promote an enjoyable environment, and in some cases make the students more motivated to learn the material.

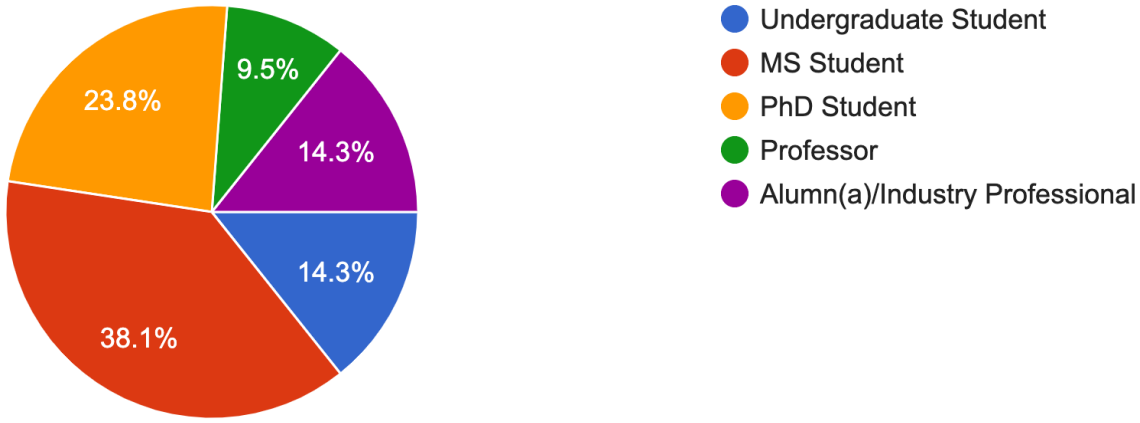
Survey and Results

The questions that this paper is attempting to answer are the following:

1. Should a fundamental security course be part of a standard Computer Science curriculum?
2. How important is a practical component for a foundational security course?
3. Should a course-wide CTF event be added to the foundational security course?
4. Are Jeopardy-style CTF challenges a good fit to be the practical component of a foundational security course?

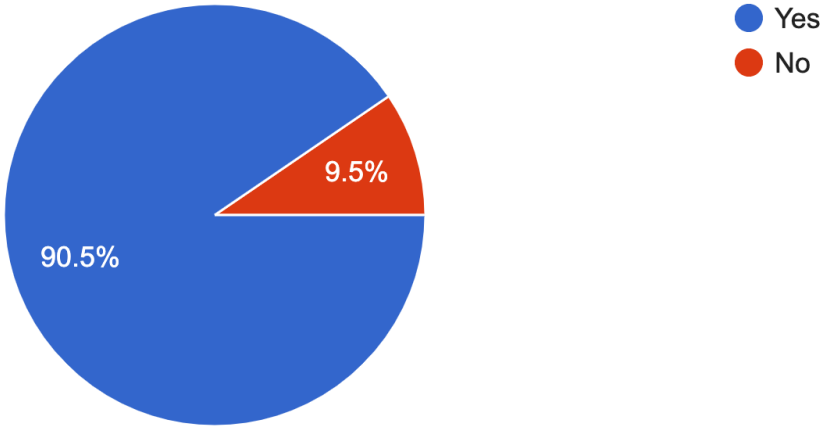
The first three questions will be discussed in this section, while the last will be addressed in the next section.To answer the posed questions a survey was conducted that was sent out to the student body, alumni and professors. All of the questions in the survey can be referenced in Appendix C. It should be mentioned that only 21 individuals responded to the survey, somewhat narrowing down the scope of the results, and potentially swaying the outcome as each vote carried 4.76% weight with it.
The survey resulted in the following demographic breakdown:

George Litvinov
Topics in SWE
Spring 2022



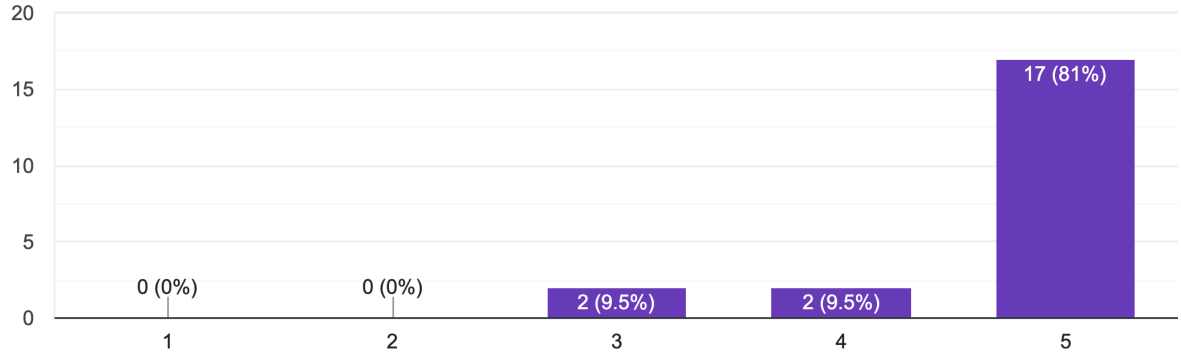
Out of the responders, 71.4% of individuals have taken a security course, while 28.6% of individuals have not. However, this data is swayed by the Professors teaching security courses at Columbia having answered “No” to this question.

An overwhelming majority of responders think that a fundamental security course should be part of a general Computer Science curriculum:



The survey queried what individuals have enjoyed and not enjoyed in the security courses they have taken. All answers to these questions can be seen in Appendix E, however the most frequent answers related to practical components in courses. Majority of the students responded saying that they enjoyed security courses with a practical component, while they did not enjoy courses that did not have a well-structured practical component. These responses were also reflected in the answers to the question of whether practical knowledge is just as important as theoretical knowledge in the cybersecurity field (1 being Strongly Disagree, 5 being Strongly Agree):

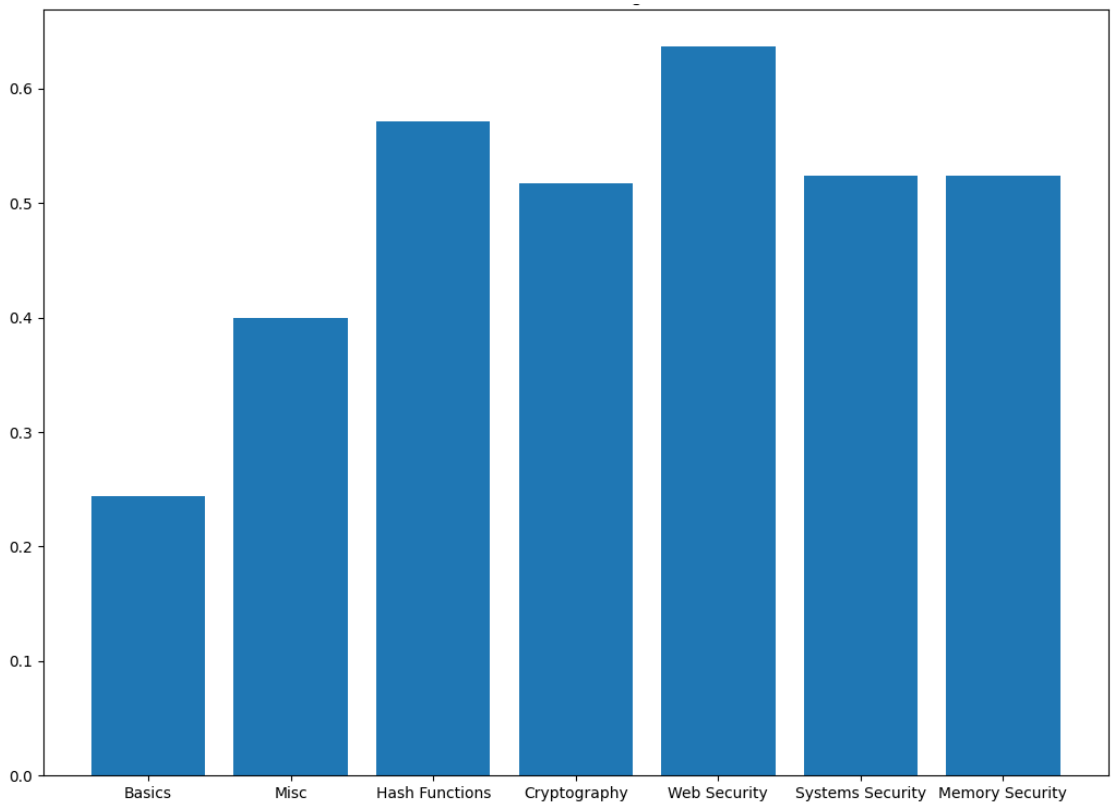
George Litvinov
Topics in SWE
Spring 2022



From this data, we can clearly gather that all demographics seem to agree on the fact that a practical component is very important for a fundamental security course.

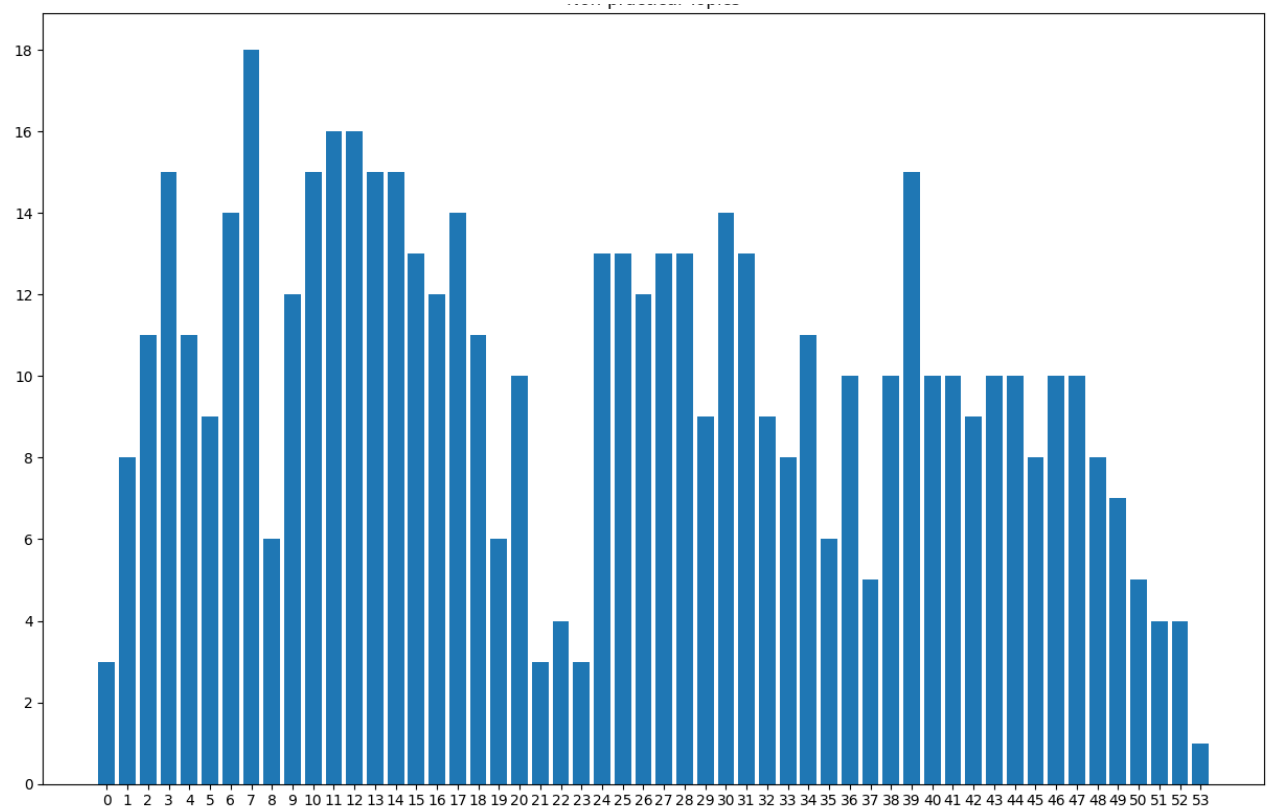
In the survey, the responders were also asked to identify what topics they found to be important material for a fundamental security course. The following graph is the breakdown of categories and topics, respectively, that should be covered in a fundamental security course (all topic-category and topic-id ties can be referenced in Appendix A):

The graph shows the average importance of the topic in that category, 1 being highest.



George Litvinov
Topics in SWE
Spring 2022

Topical breakdown by count of votes.



The following were the top 10 selected topics to be covered:

ID	Topic	Category	Count
7	Basic Asymmetric Encryption (RSA/Diffie-Hellman/Digital Signatures)	Cryptography	18
12	Public Key Infrastructure	Web Security	16
11	TLS	Web Security	16
14	SQL injections	Web Security	15
3	MD5/SHA/others	Hash Functions	15
13	XSS	Web Security	15
39	VPN	Web Security	15
10	Network Stack	Web Security	15
30	Firewalls	Web Security	14
6	Symmetric Encryption	Cryptography	14

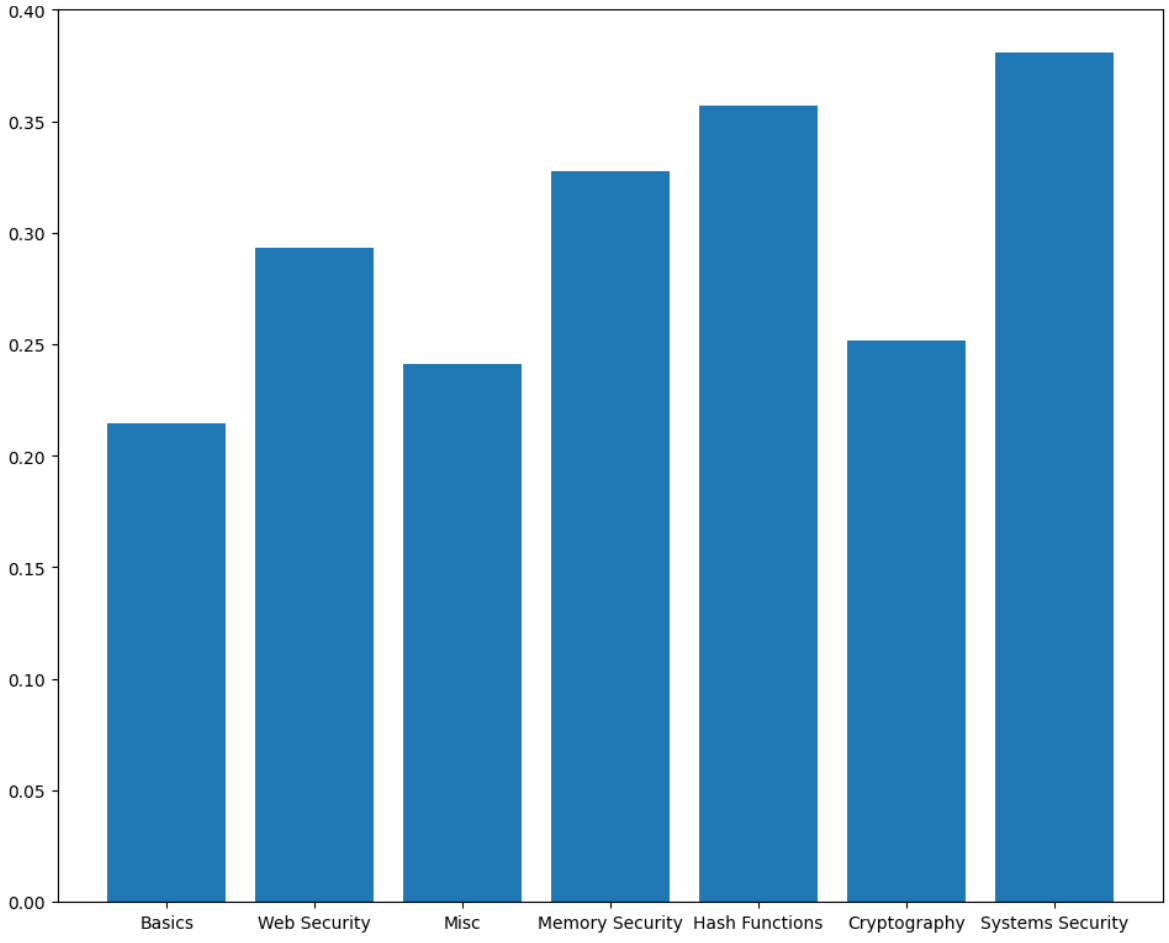
It was interesting to see that there was no major importance difference between the five categories of Hash Functions, Cryptography, Web Security, System Security and Memory Security. The Miscellaneous category may have been weighed down by some of the topics in it, such as Election Security only receiving 4 votes. However, topics such as Ethics and

George Litvinov
Topics in SWE
Spring 2022

Phishing/Social Engineering in the same category could be found in the top 20 topics out of the 54 total.

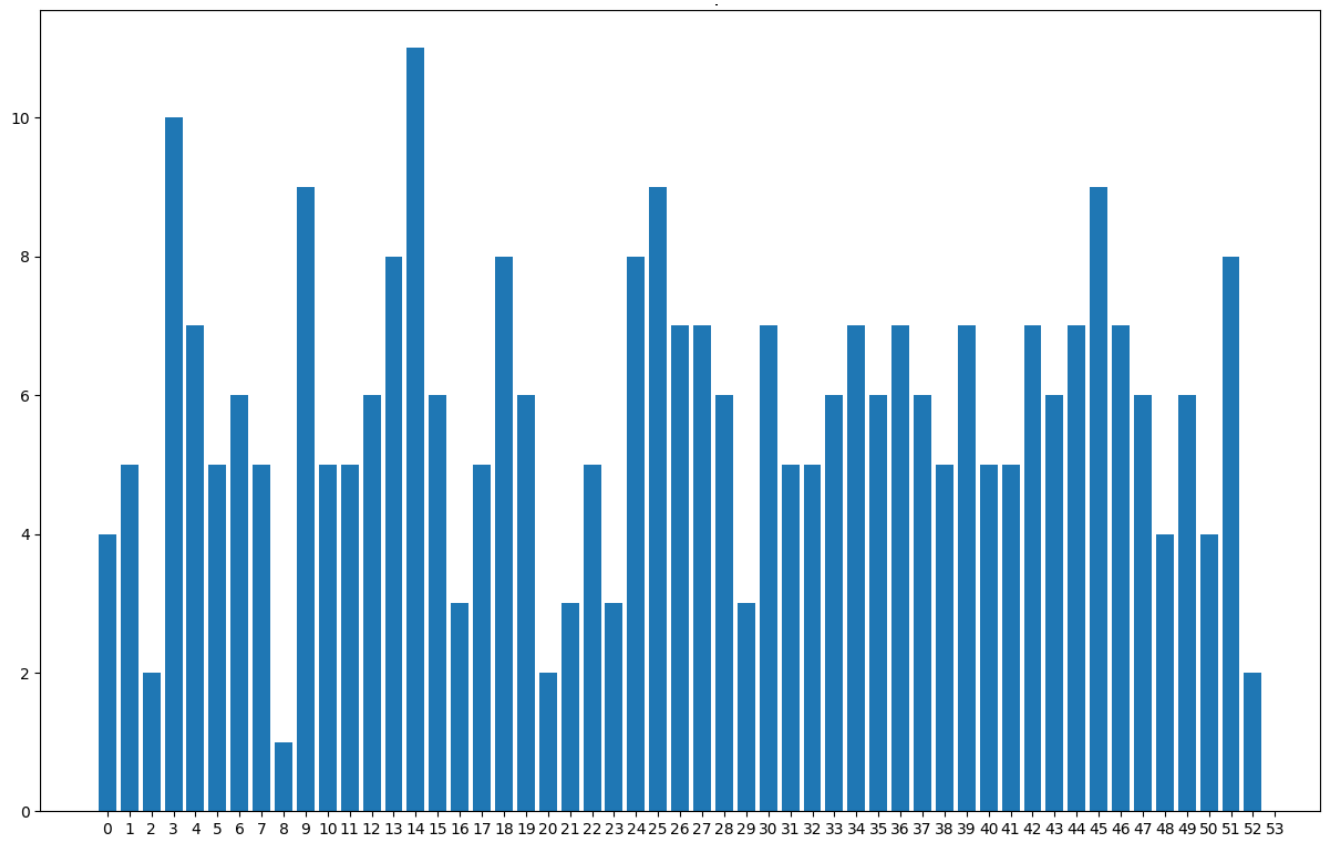
Next, responders were asked to identify topics that would be considered important practical components of a fundamental security course, again by category and by topic (data can be referenced in Appendix A):

Similarly to before, the graph above is the average importance of a topic in a category, 1 being highest).



George Litvinov
Topics in SWE
Spring 2022

Topical breakdown:



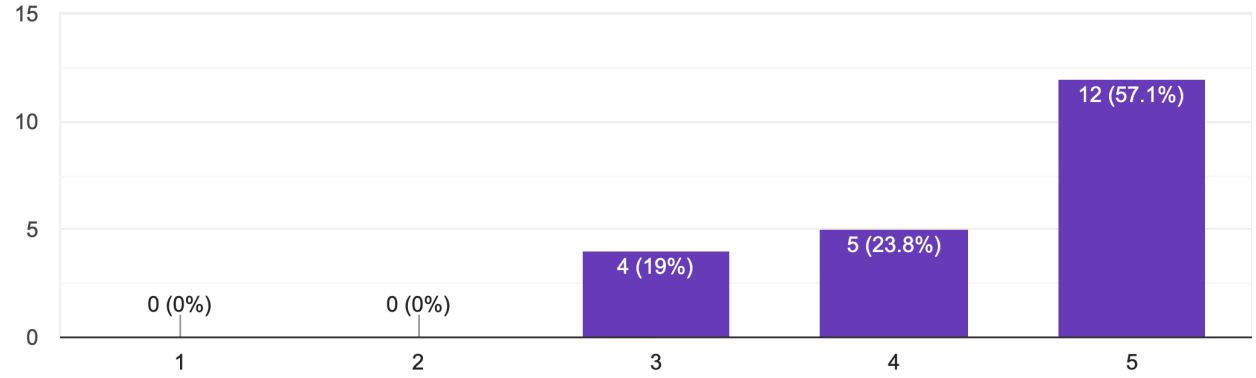
The following topics were voted as the top 10 practical components for a security course:

ID	Topic	Category	Practical Count
14	SQL injections	Web Security	11
3	MD5/SHA/others	Hash Functions	10
25	Buffer Overflows	Memory Security	9
9	Usage (how to store passwords/etc.)	Cryptography	9
45	Basics of Reverse Engineering (Malware)	Misc	9
13	XSS	Web Security	8
24	Basics of Memory	Memory Security	8
18	Permissions	Systems Security	8
51	Programming in Python	Basics	8
39	VPN	Web Security	7

Interestingly, less votes were made in total for practical components. Additionally, in the practical category breakdown, Cryptography is deemed to be less important than previous non-practical

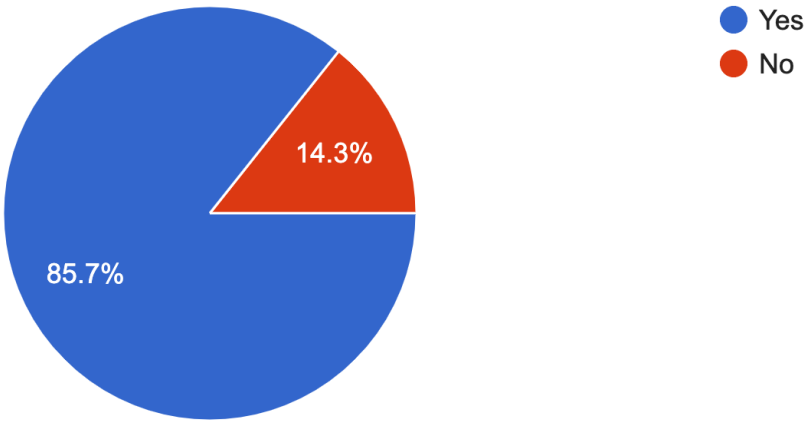
votes. Additionally, Programming in Python was deemed important for the practical component along with Malware Reverse Engineering.

The responders also agreed that gamification would allow for better retainment of material for such as course:



Knowing the topical and categorical breakdown, along with the education expertise of professors at Columbia University, the fundamental security course can be tweaked to create a better student experience. This categorical and topical breakdown can also be kept in mind during the design of the practical challenges for the course.

Moving onto hosting a course-wide CTF event. An overwhelming majority of responders responded that it would be beneficial for students if a fundamental security course hosted a CTF competition where students could form teams and compete against each other:



Additionally, the data showed that such an event would promote students to learn more in the course, as well as make the course more fun (Appendix D). Other research has shown that points and a leader board were the most effective game mechanisms for educational purposes

(Dicheva et al.), thus making a CTF competition the perfect candidate for the gamification of a fundamental security course.

Proposed Course Practical Components

I would like to propose to tweak the current theory-centric approach of the security course, by implementing a CTF-like practical component to it to promote practical skill growth. As seen in other courses, this may help students to perform better in future, more difficult courses (Lee et al.), but also grow a skill set that may be applied after graduation from the university. The course coverage may be slightly changed accounting for the voted topics in the survey, however the professorial expertise takes precedence in this case.

A Jeopardy-style CTF structure is a great candidate for the practical component of a fundamental security course, as its challenges are topic-specific, can be rolled out in batches, can be easily updated, can prevent cheating, build real-world practical skill and are a gamification of security.

There are plenty of open-source CTF frameworks that can be used to set up the practical component as well as host the course-wide CTF event. (“picoCTF”) (ZANNI) (“RTB-CTF-Framework”) These frameworks allow for students to register individual or team accounts, scoreboard tracking and time-based rollouts of exercises. Additionally, they allow for easy writing of challenges, flag (solution) randomization as an anti-cheating mechanism, and submission tracking. The flag randomization allows for each student, or each team to have a different flag from another, thus preventing the sharing of solutions. Additionally, the submission of solution code would allow checking for plagiarism with methods such as used in the *Advanced Programming* course (Lee). Such a practical exercise structure would also allow for the course to adapt to the ever-evolving field of cybersecurity, as challenges can be very easily added to the CTF platform.

Furthermore, such a structure also allows for students to be introduced to a wide range of security topics, which is the point of a fundamental security course. A grading structure where a student must solve 1-2 high point, difficult challenges in a topic that interests them, would allow for so leeway for students to explore certain topics in a deeper manner. This specialized knowledge would then be a great asset for teams during the course-wide CTF competition where students would compete in teams against each other.

A Jeopardy-style CTF structure is a better candidate than an Attack-Defense structure as presented by Mircovic et. al, as it is easier to set up and maintain, while allowing for higher modularity in terms of challenges. It is also easier to run throughout a longer period of time.

Such a structure for the practical component of the course is also adaptable for a virtual learning experience. Majority of CTFs are held online rather than on-site and a simple communication method such as a Discord channel may replace the in-person team competition.

The last consideration that needs to be made is ethics. Afterall, these security skills can be applied in illegal activities. Teaching these practical methodologies may allure negative influences on students, however it is also an important aspect of understanding cybersecurity and an adversarial mindset that may help in a student's professional development. Pike showed that the involvement of students with peer groups, clubs and competitions highly reduces the risk of illegal activity (Pike). One such peer group on Columbia's campus is CUCyber ("CUCyber"), a security group that involves itself in building a friendly community, competing in CTF events, professional networking, and funding of independent research.

Conclusion

Through literature review as well as the collected survey results it is clear that a fundamental security course should be part of every computer science curriculum and development of practical security skills should be a big part of that course. Furthermore, the addition of gamification to the curriculum of such a course would allow students to remain more engaged, retain more information and have more fun along the way. There is already a gamified event framework where cybersecurity enthusiasts can hone their practical skill which would be the ideal addition to a theory-centric security foundations course. Jeopardy-style CTF challenges are topical, adaptable to future developments in the technology landscape as well as for virtual-only learning, and give the instructor the ability to implement anti-cheat mechanisms all while gamifying the development of practical security skills.

While performing research for this paper I was surprised to not have found any schools adopting this methodology to their teaching. While there have been attempts in implementing Attack-Defense CTFs into a security course (Mirkovic and Peterson), and there is a record of a high-school level introductory class using CTFs as its main component (McDaniel), there are no implementations of Jeopardy-style CTFs into a foundational, theory-centric university security course. I was also very surprised in how similar this methodology was to *Advanced Programming*'s approach to teaching systems programming, including the course-wide collaborative event. (Lee et al.)

I believe that the addition of a Jeopardy-style CTF methodology to the current offering of *Security I* (Bellovin) as the practical component will produce students with both strong theoretical as well as practical backgrounds that will benefit them in future harder courses, and their professional development.

George Litvinov
Topics in SWE
Spring 2022

Works Cited

"abs0lut3pwn4g3/RTB-CTF-Framework: A fast, efficient and lightweight (~100 KB) Capture The Flag framework inspired by the HackTheBox platform. Built with Flask." *GitHub*, https://github.com/abs0lut3pwn4g3/RTB-CTF-Framework. Accessed 9 March 2022. [1]

Bellovin, Steven. "COMS W4181: Computer Security I." *Computer Science, Columbia University*, 15 December 2020, https://www.cs.columbia.edu/~smb/classes/f20/. Accessed 9 March 2022. [2]

Bellovin, Steven M. *Thinking Security: Stopping Next Year's Hackers*. Addison-Wesley, 2016. [3]

Bishop, Matt. "Learning and Experience in Computer Security Education." *eScholarship*, 1 September 2012, https://escholarship.org/content/qt98g0p8f4/qt98g0p8f4.pdf?t=onr5mp. Accessed 8 March 2022. [4]

Carloni, Luca. "CSEE W4868 - System-on-Chip Platforms." *Computer Science, Columbia University*, http://www.cs.columbia.edu/~cseesoc/. Accessed 9 March 2022. [5]

Carloni, Luca P., et al. "Teaching Heterogeneous Computing with System-Level Design Methods." *Proceedings of the Workshop on Computer Architecture Education*, no. WCAE'19, June 2019, pp. 1-8. *ACM Digital Library*, https://dl.acm.org/doi/abs/10.1145/3338698.3338893. [6]

"CSAW." *NYU CSAW*, https://www.csaw.io/. Accessed 8 March 2022. [7]

"CUCyber." *CUCyber @ Columbia*, http://cucyber.cs.columbia.edu/. Accessed 9 March 2022. [8]

"Cyberforce." *Department of Energy's CyberForce® Program*, https://cyberforcecompetition.com/. Accessed 8 March 2022. [9]

"DEFCON." *Defcon.org*, https://defcon.org/. Accessed 8 March 2022. [10]

Dicheva, Dariana, et al. "Gamification in Education: A Systematic Mapping Study." *Educational Technology & Society*, vol. 18, no. 3, 2015, pp. 75-88, https://www-jstor-org.ezproxy.cul.columbia.edu/stable/jeductechsoci.18.3.75?pq-origsite=summon&seq=1#metadata_info_tab_contents. [11]

"Information Security Analysts : Occupational Outlook Handbook : US Bureau of Labor Statistics." *Bureau of Labor Statistics*, 12 January 2022, https://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm#tab-1. Accessed 7 March 2022. [12]

Lee, Jae, et al. *Follow the River and You Will Find the C*. *Columbia University*, http://www.cs.columbia.edu/~jae/papers/3157-paper-v2.2-camera-final.pdf. [13]

Lee, Jea W. "COMS 3157 Advanced Programming." *COMS W3157 Advanced Programming*, http://www.cs.columbia.edu/~jae/3157/. Accessed 9 March 2022. [14]

McDaniel, Lucas. "Capture the Flag as Cyber Security Introduction." *49th Hawaii International Conference on System Sciences (HICSS)*, 2016, https://ieeexplore.ieee.org/abstract/document/7427865. [15]

Mirkovic, Jelena, and Peter Peterson. "Class Capture-the-Flag Exercises." *3GSE'14*, 2014,

https://www.usenix.org/conference/3gse14/summit-program/presentation/mirkovic. [16]

Murillo-Zamorano, Luis R., et al. “Gamification and active learning in higher education: is it possible to match digital society, academia and students' interests?” *International Journal of Educational Technology in Higher Education*, vol. 18, 2021, https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-021-00249-y. [17]

“picoCTF/picoCTF: The platform used to run picoCTF 2019.” *GitHub*, https://github.com/picoCTF/picoCTF. Accessed 9 March 2022. [18]

Pike, Ronald E. *The “Ethics” of Teaching Ethical Hacking*. https://scholarworks.lib.csusb.edu/jitim/vol22/iss4/4/. [19]

“A Resilient Cybersecurity Profession Charts the Path Forward.” *ISC2*, https://www.isc2.org/-/media/ISC2/Research/2021/ISC2-Cybersecurity-Workforce-Study-2021.ashx. Accessed 7 March 2022. [20]

Schweitzer, Dino, and Jeff Boleng. *Designing Web Labs For Teaching Security Concepts*. 2009, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.463.8572&rep=rep1&type=pdf. [21]

Yurcik, William, and David Doss. *Different Approaches in the Teaching of Information Systems Security*. 2001. *ISECON*, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.9739&rep=rep1&type=pdf. [22]

ZANNI, Alexandre. “Orange-Cyberdefense/ctf-party: A library to enhance and speed up script/exploit writing for CTF players.” *GitHub*, https://github.com/Orange-Cyberdefense/ctf-party. Accessed 9 March 2022. [23]

George Litvinov
Topics in SWE
Spring 2022

Appendix A: Topic Data Table

ID	Topic	Category	Practical Freq.	Theoretical Freq.
0	Programming in C	Basics	4	3
1	Introduction to Assembly	Basics	5	8
2	Adversarial Mindset	Misc	2	11
3	MD5/SHA/others	Hash Functions	10	15
4	AES CBC vs ECB	Cryptography	7	11
5	Rainbow Tables and Hash Cracking	Hash Functions	5	9
6	Symmetric Encryption	Cryptography	6	14
7	Basic Asymmetric Encryption (RSA/Diffie-Hellman/Digital Signatures)	Cryptography	5	18
8	Advanced topics s.a. ECDAA/Random Number Generators	Cryptography	1	6
9	Usage (how to store passwords/etc.)	Cryptography	9	12
10	Network Stack	Web Security	5	15
11	TLS	Web Security	5	16
12	Public Key Infrastructure	Web Security	6	16
13	XSS	Web Security	8	15
14	SQL injections	Web Security	11	15
15	Proxies and MITM Attacks	Web Security	6	13
16	Ethics	Misc	3	12
17	Social Engineering & Phishing	Misc	5	14
18	Permissions	Sys. Security	8	11
19	Basics of Linux Systems and Commandline	Basics	6	6
20	Basics of Networks	Basics	2	10
21	Basics of Windows Systems and Commandline	Basics	3	3

George Litvinov
Topics in SWE
Spring 2022

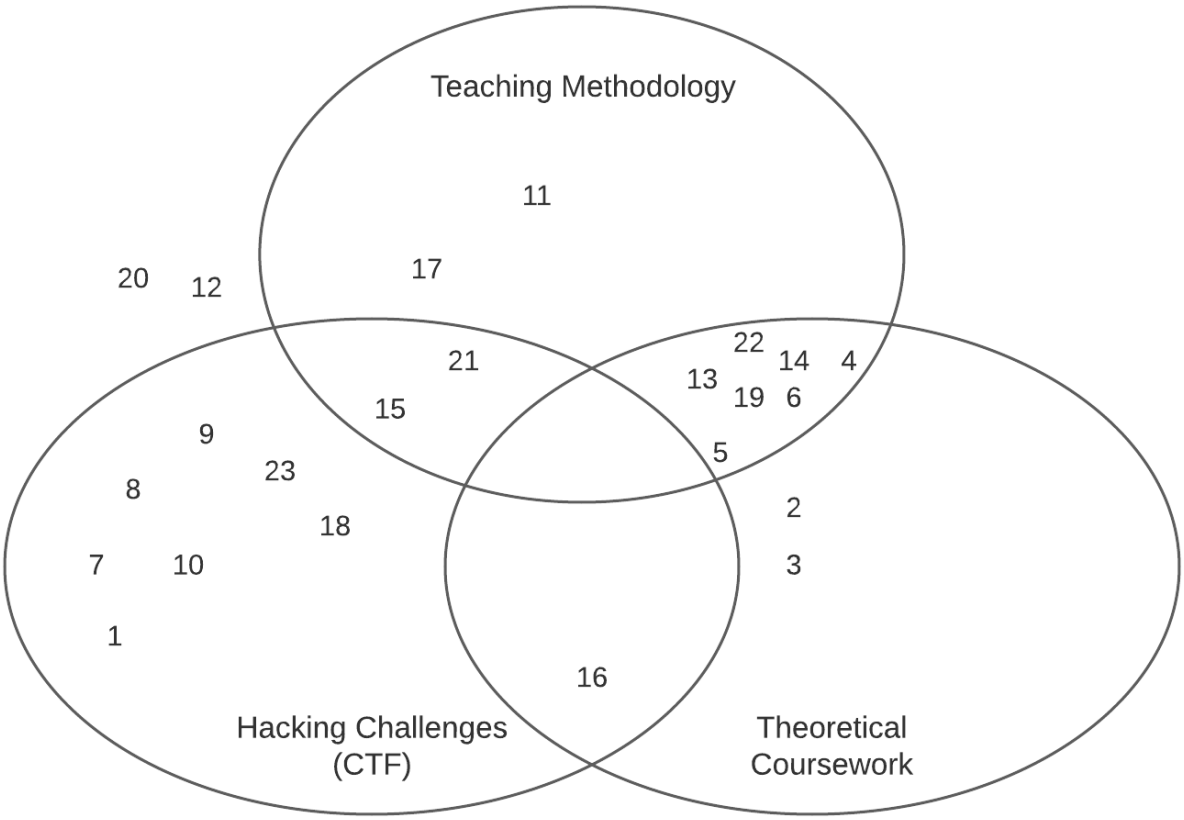
22	Basics of Bash	Basics	5	4
23	Basics of Powershell	Basics	3	3
24	Basics of Memory	Mem. Security	8	13
25	Buffer Overflows	Mem. Security	9	13
26	ROP Chain Attacks	Mem. Security	7	12
27	Heap Overflows	Mem. Security	7	13
28	TCP/IP	Web Security	6	13
29	Routing and BGP	Web Security	3	9
30	Firewalls	Web Security	7	14
31	Denial Of Service	Web Security	5	13
32	Secure Network/System Architecture Design (Enterprise networks/IoT Devices/etc.)	Misc	5	9
33	Logging	Misc	6	8
34	Sandboxing	Misc	7	11
35	Metasploit	Misc	6	6
36	Introduction to Capture The Flag (hacking competitions)	Misc	7	10
37	Bug Bounties	Misc	6	5
38	How to Design protocols (designing cryptographic systems)	Cryptography	5	10
39	VPN	Web Security	7	15
40	Tor	Web Security	5	10
41	Personal Best Practices	Misc	5	10
42	Canaries	Mem. Security	7	9
43	Use after Free	Mem. Security	6	10
44	Fuzzing	Mem. Security	7	10

George Litvinov
Topics in SWE
Spring 2022

45	Basics of Reverse Engineering (Malware)	Misc	9	8
46	Intrusion Detection	Misc	7	10
47	Web Server Architecture	Web Security	6	10
48	ASLR	Mem. Security	4	8
49	Digital Forensics	Misc	6	7
50	Application in the Blockchain	Cryptography	4	5
51	Programming in Python	Basics	8	4
52	Election Security	Misc	2	4
53	Authentication and authorization	Misc	0	1

George Litvinov
Topics in SWE
Spring 2022

Appendix B – Venn Diagram



Numbers reference the [#] next to the works cited. Plenty of the sources have changed, since some of them had no new information for me that I could include into the paper. Also, as per the request from the project update, I have added the course websites for the courses that I have cited. The two resources outside of the venn diagram were used in the introduction to support that the security field is growing, thus making security education very important. Resource [16] seems to be the most tangential paper to what I have proposed in this paper, however I am attempting to show that Jeopardy-style CTFs are better suited for security education than Attack-Defense style. I also would like to note that the paper-number association is different between the progress report and this paper, since the former was more categorically ordered, while the latter is alphabetically ordered.

George Litvinov
Topics in SWE
Spring 2022

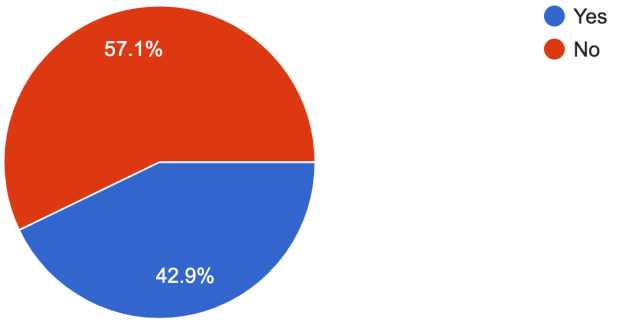
Appendix C – List of all Survey Questions

1. What is your name?
2. What is your email address?
3. Are you an undergraduate student, MS student, PhD student, Professor or Alumn(a)/Industry Professional?
4. Have you taken a security course before?
5. Please name the security courses that you have taken.
6. Please describe what you enjoyed about them? (Appendix E)
7. Please describe what you did not enjoy about them? (Appendix E)
8. Have you taught or assisted a security course? (Appendix D)
9. Do you think that a fundamental security course should be taught in a general Computer Science curriculum?
10. What should be a student's knowledge base before taking a security course? (Introductory, Advanced or Experienced)
11. Out of the following list, please check topics that are in your opinion important for a foundational security course. (Appendix A)
12. Given the same list as in the previous question, please indicate what topics should have a practical component.
13. Have any topics been missed? (Added to data if missed)
14. Do you think it would be beneficial to the students if the course hosted a Capture The Flag competition where students could compete in groups against other peers? (similarly to some courses hosting hackathons)
15. If yes, should this competition be required to attend? (Appendix D)
16. Agree or Disagree, gamification in education allows for better retainment of material.
17. Agree or Disagree, competition (such as a Hackathon or CTF) in a course will push students to learn more from a course.
18. Agree or Disagree, competition (such as a Hackathon or CTF) in a course will make the course more fun.
19. Agree or Disagree, practical knowledge is just as important as Theoretical Knowledge in the Cybersecurity Field.

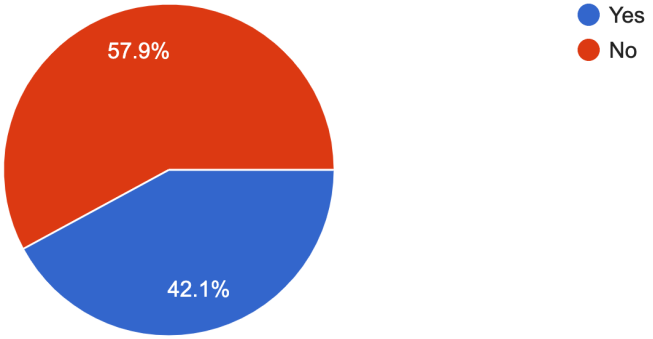
George Litvinov
Topics in SWE
Spring 2022

Appendix D – Misc. Graphs from the Survey

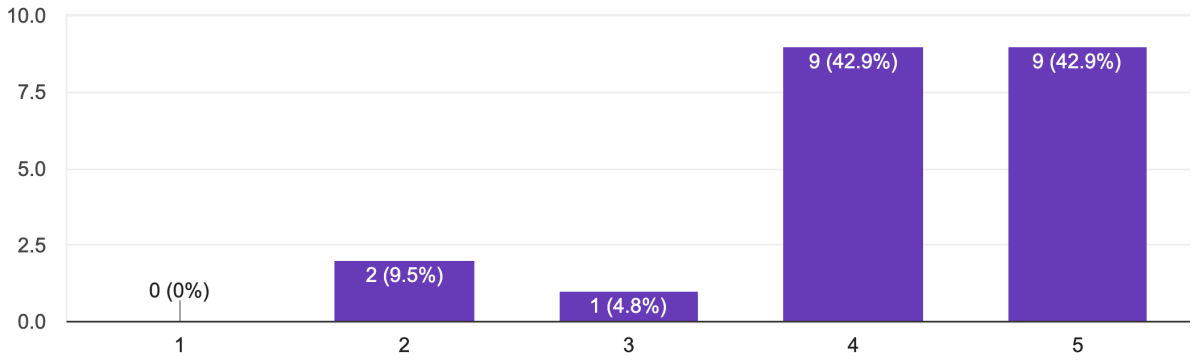
Answers to whether the participants of the survey have taught or assisted a course:



Answers to whether a course CTF competition should have required attendance:

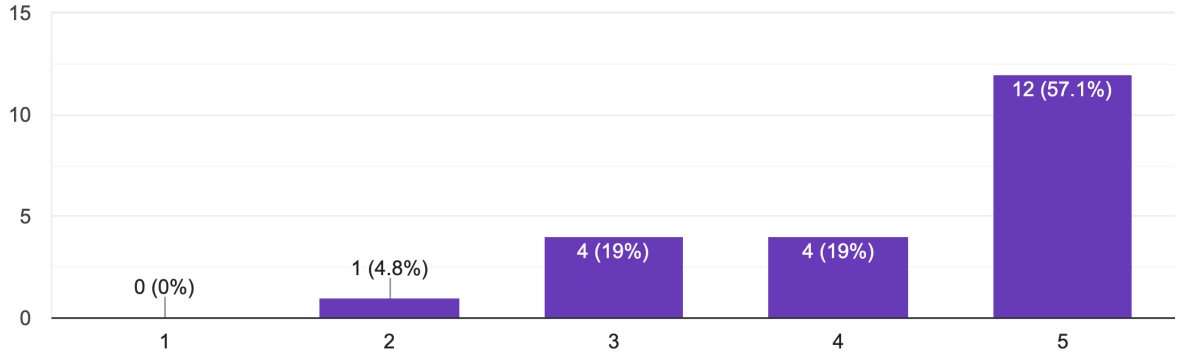


CTF competition would allow students to learn more
(1 strongly disagree, 5 strongly agree):



George Litvinov
Topics in SWE
Spring 2022

CTF competition would make the course more fun
(1 strongly disagree, 5 strongly agree):



George Litvinov
Topics in SWE
Spring 2022

Appendix E – Text Answers to Survey questions:

If you have taken a security course, what have you enjoyed from it:

- A different perspective to topics studied in computer science
- Providing an extra layer of a security mindset when designing systems
- Exposure to a significant specialization in the field

- Topics, learning the basics
- Practical teaching of skills
- Strong theoretical background

- Able to do real attacks on toy applications

- Getting to know the OS and the low level API better
- Getting to learn how malicious actors exploit well intentioned API and UX
- Getting better at working with Assembly too

- Interesting context
- Something I can not understand on my own
- Well-constructed learning path

- The hands-on experience in the malware analysis class
- The real-world incidents lecture at the end of the malware analysis class
- The high-level overview of different security concepts/topics in the security architecture class

- Writing exploits and understanding how to defend against them
- Learning what good practices for secure systems are

- Enjoyed learning the math behind the algorithms in cryptography topics
- Enjoyed learning the types of attacks that are possible

- Programming exercises
- Good TA's
- Spirit of Competition

- Hands-on experience

- I enjoy classes that are hands on and interactive with realistic environments

- Learning about the idea of how to decrypt and encrypt

George Litvinov
Topics in SWE
Spring 2022

If you have taken a security course, what did you not enjoy from it:

- I would have preferred to have a more hands on experience in some of the courses

- Spending most of the time on theory can lead to lack of the industry practices

- The variety of security courses could be more.

- Disorganized homeworks
- Old syllabus
- Not enough practical exercises

- It's a little bit hard to get up early for the morning course.
- Maybe more recitation class on problem examples.

- I disliked the written (essay-stile) exams. I believe security-class evaluation should be more practical (or at least the practical part, such as projects should have higher grading percentage).
- I wished there had been more real-world examples/incidents to further demonstrate the practical impact of the studied topics.

- Too easy; I wished there were more challenges.

- Bad Lectures
- Disinterested Professors
- Practice problems not guaranteed to be reflected in real world

- Tutoring style,

- I dislike classes that are slide-centric, lack interactive content, focus on memorization, are oriented to certifications, have stale content, or focus on boring IT-related things like configuring group policy.

- Too many library and old style coding