Teaching Cybersecurity to Computer Science Students Utilizing Terminal Sessions Recording Software as a Pedagogical Tool

Ismail Hassan
Department of Computer Science
OsloMet – Oslo Metropolitan University
Oslo, Norway

***Abstract*—This Innovate Practice Full Paper presents our experience with teaching a cybersecurity course to undergraduate students utilizing terminal sessions recording software as a pedagogical tool.**

Developing a practical cybersecurity course demands numerous amount of preparation and the deployment of various security tools. One of the significant benefits of providing hands-on assignments is the opportunity for students to practice and experiment with tools similar to the ones used by both the professionals and malicious actors.

This paper presents an innovative approach to conducting interactive demonstrations in the classroom and as a supplement to the weekly practical assignments. By combining two simple open-source software, we managed to create several short tutorials on how to use various command-line security tools. The educational tool was deployed in an undergraduate cybersecurity class, and the results from the anonymous online survey were overwhelmingly positive. Furthermore, the participants responded that the demos improved their understanding of the subject content and enabled them to learn the material effectively.

***Keywords*—Cybersecurity, Asciicast, Blended Learning**

I. Introduction

The rapid advancement of Information and Communication Technology (ICT) is profoundly impacting the way we communicate, socialize, purchase goods, or consume services. Despite the substantial benefits brought by ICT, vulnerabilities, threats, and attacks are a growing phenomenon. Governments, businesses and consumers are finding it very difficult to safeguard their digital assets from potential attacks, and the need for cybersecurity professionals is overwhelming [1]–[3].

To meet the increasing demand, universities have a significant role and responsibility in educating skilled professionals. Teaching cybersecurity theories and principles are essential in understanding the subject matter. Nevertheless, it is also indispensable to equip students with the practical skills required to execute that knowledge.

Developing a practical cybersecurity course demands numerous amount of preparation and the deployment of various security tools. One of the significant benefits of providing hands-on assignments is the opportunity for students to practice and experiment with tools similar to the ones used by both the professionals and malicious actors.

II. Motivation and Background

In 2016, the author of this paper was commissioned to teach a course in cybersecurity offered at the Department of Computer Science to 3rd-year undergraduate students. The department has three bachelor programs. The course was mandatory for students enrolled in the Bachelor of Information Technology. In contrast, students could take it as an elective if enrolled in the Bachelor of Software Engineering or Bachelor of Applied Computer Technology.

A. Hands-on experience

A common criticism regarding cybersecurity education programs is that an over-emphasis on theory and the lack of integrating hands-on learning environment into the curriculum [2].

In the first year of being in charge of the course and after introducing several practical exercises to supplement the lectures, the author observed that a great deal of time was spent helping students with the tools they were required to use to conduct the weekly practical assignments.

Students from the different programs had varying experiences with Linux, Operating Systems, Computer Networking, and command-line tools. Some lacked the basics and required more extensive help than others.

B. Mandatory for all

After receiving repeated feedback from the industry on the relevance and significance of cybersecurity, the department decided to make it a mandatory course for all computer science bachelor's programs from the 2018 academic year.

This lead to an increase in the number of students taking the course from 104 in 2017, 132 in 2018 and 161 in 2019. The demand for more human resources to facilitate the lab sessions was imminent, but hiring more teaching assistance was not feasible. The option was to either cut down on the practical assignments or find alternative solutions.

As proponent believer of the benefits of hands-on learning environments, the author determined to explore means of delivering short tutorials to assist all students in getting accustomed to the tools used in the weekly practical assignments. The timeline to implement the solution was set to Autumn 2018.

C. Screencast or an Asciicast?

A vast amount of tools are freely available online, providing anyone the possibility to scan, attack, and compromise resources connected to the Internet. These tools may provide a Graphical User Interface(GUI), but often a command-line interface (CLI) through a terminal emulator is used. There are concerns that students might spend too much time trying to figure out how to use the software rather than learning the subject matter. Novice students might get discouraged due to the steep learning curve some of the software could impose. To remedy this difficulty, educators have traditionally resorted to using instructional videos or screencasts [4]–[7].

Screencasts are video recordings that capture an instructor's computer screen activities. Audio or textual annotation can be inserted into the video to describe the process. Some screencast software have simple editing capabilities, but generally, supplementary editing applications are required to produce the video. Depending on the screencasting software, a considerable time might be needed to edit the final product. Unlike screencasts, our proposed method does not require any editing or specialized applications for post-processing the result.

Asciicast, which is the method used in this study, captures the output instantly from commands executed on the terminal, saves it to file, which is then uploaded to a web site. Students can then access the captured session from anywhere and playback the recording at their convenience.

III. Related work

The use of lecture videos and screencasts as a pedagogical tool in education is not new, and it is an area that has been widely studied. Although there are a growing number of companies, educational institutions, and individuals that are utilizing Terminal Sessions Recording tools for educational purposes, limited studies have been reported.

The authors of [8] developed a system that monitors Linux terminal input and output for each student and produces both summary and detailed statistics of student progress. While the underlying method is similar to our proposed solution, the intended target audience differs. The analysis tools presented produces outputs that are intended to help teachers identify which part of a practical exercise that are challenging for the student. In contrast, our solution aims to guide the student in learning how to use the security tools.

A system for teaching and assessing command line terminal skills was proposed by [9]. It allows instructors to create auto-graded terminal assignments that require students to perform a high-level action that can be completed in many ways.

The paper [10] presented the tool forscript that enables a forensic investigator to convert an interactive command-line session into a version suitable for inclusion in a printed report. The software builds upon the Linux script command and also offers capabilities similar to the tools used in this study.

Our proposed solution is closely related to the study reported by [11]. The authors designed interactive illustrations to assist the students in quickly getting acquainted with the Unix operating system and practice lab assignments at their own pace. The authors reported positive results in students satisfaction with the their approach.

IV. Course Description

Cybersecurity covers a wide range of topics encompassing many aspects of the modern computing ecosystem. It is, therefore, a challenging task for a single course to cover both broadly and in-depth all of the domain-specific knowledge demanded by the industry.

The cybersecurity course curriculum offered at our department includes a broad range of skills following the 2017 Cybersecurity Curricula guidelines [12] published by the Joint Task Force on Cybersecurity Education under the supervision of the Association for Computing Machinery (ACM).

A. Learning outcomes

After completing the course, students are expected to have achieved the following learning outcomes defined in terms of knowledge, skills and general competence:

- Knowledge:
  - know the basic security principles of confidentiality, integrity and availability.
  - identify common vulnerabilities, threats, threat agents, risks, and attack vectors.
  - know the basics of cryptography and how it is used to protect data at rest and in transit.
  - understand the different forms of authentication methods that can be utilized.
  - explain the different types of access control models that can be used to safeguard information security.
  - describe the protocols and standards related to identity, authentication and authorization.
  - understand the importance of controlling the flow of information in and out of the enterprise network.
  - have knowledge of mechanism for detecting anomalies and incidents early to detect and handle attacks.
  - understand the concept of Secure Software Development and Privacy by Design.
- Skills:
  - utilize security tools for encryption and signing.
  - utilize programs to identify and detect vulnerabilities.
  - enforce the principle of least privilege in services and other resources by using the Identity and Access Management System.
  - filter and control the traffic between the various security zones by using a firewall technology.
  - detect and manage data attacks using Intrusion Detection and Prevention Systems (IDS / IPS).
- General competence:
  - discussing and communicating issues related to security principles confidentiality, integrity and accessibility.
  - comparing, assessing and providing recommendations on the use and procurement of security solutions.

B. Course Description

The course is taught in the 5th semester to 3rd-year Computer Science students. The semester lasts for 14 weeks, and the schedule for each week consists of 2 hours of lecture and 2 hours of lab sessions. An overview of the course plan and topics covered in each lecture are depicted in Table I.

TABLE I
COURSE OUTLINE.

Lecture	Topic
1	Introduction
2	Principles of Computer Security
3	Introduction to Cryptography (Classical)
4	Symmetric & Asymmetric cryptography (Modern)
5	Cryptographic protocols and standards - SSL/TLS - PGP
6	Identity & Authentication
7	Authorization (Access Control) - ACL, MAC, DAC, RBAC and ABAC
8	Identity, Authentication and Authorization protocols - SAML2, Oauth2, OpenID connect, JWT and WebAuthn
9	Network Security I - Port scanning - Firewall protection
10	Network Security II - Intrusion Detection/Prevention Systems
11	Software Security I - OWASP
12	Software Security II - Web Security
13	Malware & Phishing
14	Privacy by Design

V. THE PROPOSED SOLUTION

Most of the weekly assignments require the students to perform some tasks using command line security tools. Learning how to use new software can be demanding and challenging, particularly for novice users. The instructor adopted a blended learning approach to keep the workload manageable.

Blended learning is an organized and systematic approach to mixing traditional face-to-face classroom teaching with activities conducted online often by means of Learning Management Systems(LMS).

Studies [13], [14] have shown that blended learning has many pedagogical benefits that improve student engagement, satisfaction, and overall learning capability.

The solution proposed in this paper achieves the following blended learning goals.

1) Provide online recordings of short demos that lead students through steps on how to complete a particular skill.
2) Provide online student-centric, self-paced weekly assignments that students can work on anytime from anywhere.
3) Provide a flexible authentic learning environment that gives the students opportunities to practice on various hands-on cybersecurity assignments at any time and on anywhere.

TABLE II
DEMO OUTLINE.

Week	Demo	Duration
1	-Brute Force Attack on Caeser Cipher -How to use Steganography -Hash functions	1m 07s 2m 00s 2m 58s
2	-Pseudorandom Number Generators -Symmetric Cryptography (AES) -Asymmetric Cryptography (RSA)	1m 52s 2m 52s 1m 19s
3	-Digital Signatures using AES & HMAC -Digital Signatures using RSA	2m 43s 3m 56s
4	-Extract SSL/TLS information (OpenSSL) -Generate PGP keys in Linux -Upload PGP keys to a PGP key server	1m 52s 6m 46s 2m 13s
5	-Testing password complexity -Password cracing with John the Ripper	1m 56s 1m 21s
6	-Generate a list of passwords with Cupp3 -Online dictionary attack with Hydra	2m 20s 2m 02s
7	-Discretionary Access Control (DAC) -Access Control List (ACL)	2m 13s 2m 14s
8	-Gather information about ports -Scanning the network with Nmap	3m 24s 2m 23s
9	-Protecting the network with a Firewall	4m 37s
10	-Host Intrusion Detection with OSSEC	5m 20s
11	-Network Intrusion Detection with SNORT	5m 10s
12	-Phishing Demo with GoPhish	

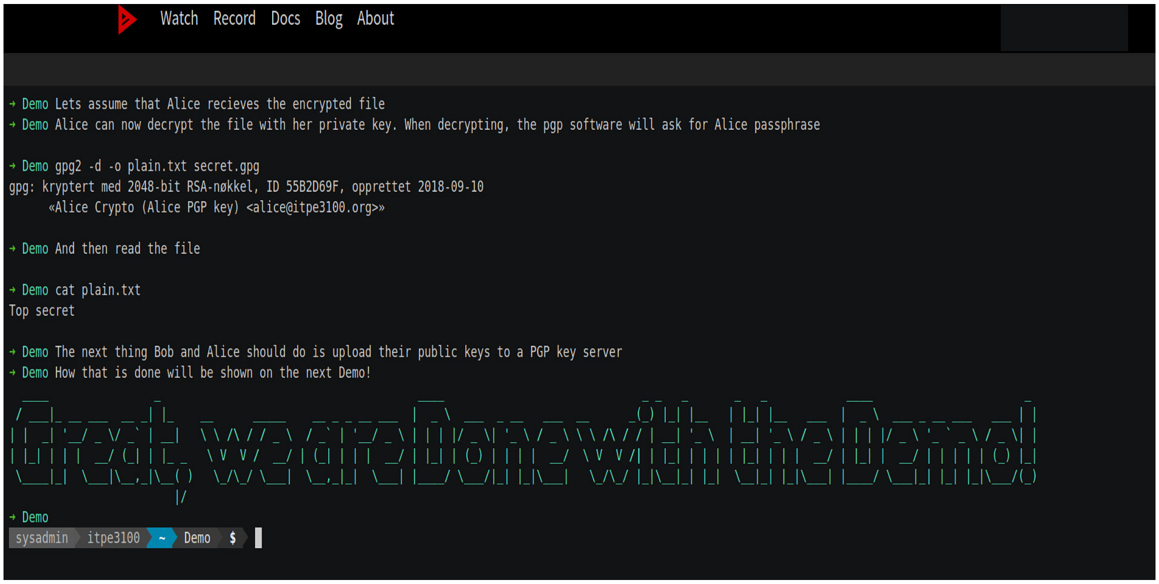
A. Online short tutorials

The instructor created several short tutorials demonstrating how various cybersecurity tools can be used. Table II shows an overview of the demos used in the course. The following open-source software were combined to create them:

- **asciinema:**a free and open source solution for recording terminal sessions and sharing them on the web [15].
  demo-magic: a handy open source shell script that enables one to script repeatable demos in a bash environment so that one does not have to type as during presentations. [16].

Asciinema is used to capture the output from a demo-magic shell script that executes sequences of steps demonstrating the whole process of performing a task from start to finish.

The captured output includes all the text and invisible escape/control sequences in a raw, unaltered form. When the recording session finishes it uploads the output (in asciicast format) to asciinema.org. Asciicast file is a JavaScript Object Notation (JSON) file containing meta-data such as duration, title of the recording, and the actual content printed to terminal's stdout during recording [15]. Fig. 1 shows the output from one of the demos. An excerpt of the demo magic script is shown in Listing. 1.



```
➜ Demo Lets assume that Alice recieves the encrypted file
➜ Demo Alice can now decrypt the file with her private key. When decrypting, the pgp software will ask for Alice passphrase

➜ Demo gpg2 -d -o plain.txt secret.gpg
gpg: kryptert med 2048-bit RSA-nøkkel, ID 55B2D69F, opprettet 2018-09-10
      «Alice Crypto (Alice PGP key) <alice@itpe3100.org>»

➜ Demo And then read the file

➜ Demo cat plain.txt
Top secret

➜ Demo The next thing Bob and Alice should do is upload their public keys to a PGP key server
➜ Demo How that is done will be shown on the next Demo!
  ____                _                          ____                                _ _   _       _   _            ____                       _ 
 / ___|_ __ ___  __ _| |_  __      _____    __ _ _ __ ___  |  _ \  ___  _ __   ___  __      _(_) |_| |__   | |_| |__   ___  |  _ \  ___ _ __ ___   ___ | |
| |  _| '__/ _ \/ _` | __| \ \ /\ / / _ \  / _` | '__/ _ \ | | | |/ _ \| '_ \ / _ \ \ \ /\ / / | __| '_ \  | __| '_ \ / _ \ | | | |/ _ \ '_ ` _ \ / _ \| |
| |_| | | |  __/ (_| | |_ _ \ V  V /  __/ | (_| | | |  __/ | |_| | (_) | | | |  __/  \ V  V /| | |_| | | | | |_| | | |  __/ | |_| |  __/ | | | | | (_) |_|
 \____|_|  \___|\__,_|\__( )  \_/\_/ \___|  \__,_|_|  \___| |____/ \___/|_| |_|\___|   \_/\_/ |_|\__|_| |_|  \__|_| |_|\___| |____/ \___|_| |_| |_|\___/(_)
                        |/
➜ Demo
sysadmin  itpe3100  ~  Demo  $
```

Fig. 1. An excerpt of a demo hosted on asciinema.org.

```
p "In this Demo we will explore a powerful Linux Host Intrusion Detection and
      Prevention system called OSSEC"

p "Before we conduct the attack simulation, lets see if the firewall on the system
      to be attacked is blocking any IP addresses"

pe "ssh 192.168.1.121 ’sudo iptables -nL | ccze -A’"

p "The results shows that no IP addresses are blocked."

p "Now lets first check the OSSEC log files on the host to be attacked. We will
      check the last 30 lines of the OSSEC log file"

pe "ssh 192.168.1.121 ’sudo tail -30 /var/ossec/logs/alerts/alerts.log | ccze -A’"
clear
p "Suppose that the attacker now executes the brute force attack from their machine
       which we do not see here in the Demo"

p "Lets check again the OSSEC log files on the host to be attacked. We will check
      the last 30 lines of the OSSEC log file"

pe "ssh 192.168.1.121 ’sudo tail -30 /var/ossec/logs/alerts/alerts.log | ccze -A’"

p "This time We see that OSSEC logs the feiled SSH login attempts."

p "If in addition, OSSEC was configured as an Intrusion Prevention system, it will
      use the firewall to block the IP address of the attacker in our case
      192.168.1.131"

p "Lets check if OSSEC blocked the IP address"

pe "ssh 192.168.1.121 ’sudo iptables -nL | ccze -A’"

p "The results now show that the IP of the attcker is blocked and depending on how
      OSSEC is configures the IP might blocked for a short time usually 5 minutes"

echo "Great, we are Done with the Demo!"
p ""
```

Listing 1. An excerpt of a demo magic script.

B. Flexible authentic learning environment

Since most the assignments are performed on the Linux command line (terminal), access to a Linux machine is essential. The department of computer science allocates laboratory space equipped with a large number of workstations for students to practice their computing skills. As the use of virtualization technologies became more widespread, several courses have migrated to using virtual machines. Students are no longer limited to only using physical workstations on campus but can use their laptops to run several virtual machines. The cybersecurity course at our depratment utilizes VirtualBox as a virtualization platform.

VirtualBox is a virtualizing software for the x86 computing architecture [17]. In contrast to cloud services, VirtualBox is run locally on the device of the student, thus eliminating the dependency on physical labs or an external cloud service that is prone to outages.

The instructor prepared a VirtualBox image containing a Linux Xubuntu core (18.04) virtual machine equipped with the bare minimum software required. This has the extra benefit of producing a small compact image that is easily distributable to the students.

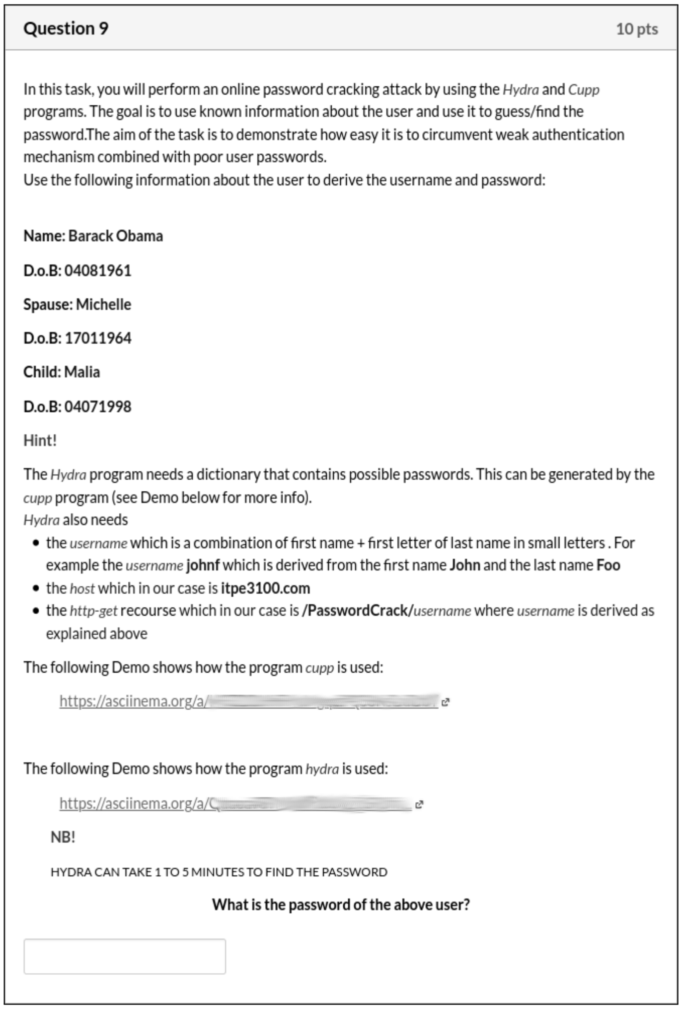
In addition, the following tools were installed on the system to support the hand-on exercises:

- **OpenSSL:** a cryptography toolkit.
- **steghide:** a steganography program that is able to hide data in various kinds of media.
- **GnuPG:** an OpenPGP encryption and signing tool.
- **john:** a tool to find weak user passwords.
- **Hydra:** a parallelized login cracker which supports numerous protocols.
- **Cupp3:** generates dictionaries for attacks from personal data.
- **Nmap:** a utility for network exploration or security auditing.
- **Snort:** a Network Intrusion Detection System.
- **Ossec:** a Host based Intrusion Detection System.

C. Flexible self-paced weekly assignments

Our university uses Canvas as the Learning Management Platform. The instructor utilized the quiz features in Canvas to design weekly practical assignments. Each assignment is generally composed of 10 questions.

In contrast to a standard multiple choice quiz, most of the tasks will require the student to solve a practical challenge. In other words, they must perform the task on the virtual



Question 9 10 pts

In this task, you will perform an online password cracking attack by using the *Hydra* and *Cupp* programs. The goal is to use known information about the user and use it to guess/find the password.The aim of the task is to demonstrate how easy it is to circumvent weak authentication mechanism combined with poor user passwords.
Use the following information about the user to derive the username and password:

Name: Barack Obama

D.o.B: 04081961

Spause: Michelle

D.o.B: 17011964

Child: Malia

D.o.B: 04071998

Hint!

The *Hydra* program needs a dictionary that contains possible passwords. This can be generated by the *cupp* program (see Demo below for more info).
Hydra also needs

- the *username* which is a combination of first name + first letter of last name in small letters . For example the *username* **johnf** which is derived from the first name **John** and the last name **Foo**
- the *host* which in our case is **itpe3100.com**
- the *http-get* recourse which in our case is **/PasswordCrack/***username* where *username* is derived as explained above

The following Demo shows how the program *cupp* is used:

https://asciinema.org/a/[redacted]

The following Demo shows how the program *hydra* is used:

https://asciinema.org/a/C[redacted]

NB!

HYDRA CAN TAKE 1 TO 5 MINUTES TO FIND THE PASSWORD

What is the password of the above user?

Fig. 2. An example of a practical weekly assignment requiring the students to use the Linux command line tool hydra and cupp3.

machine provided by the instructor. Fig. 2 shows an example of a practical task administered through the university LMS.

An added bonus of using an LMS like Canvas is the ability to create a pool of similar questions that have different parameters. Students can practice on the assignments as many times as they wish, and in each attempt, a new set of items randomly selected from the pool will be presented to them. Students can then do the assignments at their own pace and convenience.

VI. Method

For this study, the author adopted a mixed-method approach. A mixed methods research is the process of collecting and analyzing data using both quantitative and qualitative methods in a single study [18].

A. Qualitative method

For the qualitative approach, a set of online survey questions were administered to the class at the end of the semester using Microsoft online Forms. Table III shows the closed-ended survey questions. Participation was voluntary, and the students could submit the survey anonymously. Out of the 161 students enrolled in the class, 26.7% (**n = 43**) completed the questionnaire.

The survey asked participants to rate their agreement regarding their attitudes towards the statements in Table III. A 5-point Likert scale ranging from “Strongly Agree (5)” to “Strongly Disagree (1)” was used for the survey. Likert scales, named after Dr. Rensis Likert, who developed the method in 1932, are mostly employed in questionnaires to collect participant’s level of agreement with a set of statements.

TABLE III
Closed-ended survey questions.

Q1	The demos improved my understanding of the subject content and enabled me to learn the material more effectively.
Q2	The demos were well structures which helped me conduct the weekly lab assignments easily.
Q3	The demos were well documented, easy to follow and understand.
Q4	I would like to see other courses at the Faculty adopt similar demos in their classes.

Furthermore, open-ended questions were included in the survey to evaluate the course. Table. IV shows the questions. We noticed that some students used this part of the survey to further express their perception towards the demos. The author determined to include excerpts of those responses in this study.

TABLE IV
Open-ended survey questions.

S1	What parts of the course did you enjoy/like?
S2	What parts of the course would you like to see improved?
S3	Any other suggestions that you would like to recommend?

B. Quantitative method

Video hosting services such as Youtube, Vimeo, or even some Learning Management System(LMS) employed by universities may offer much statistical data on how users interact or engage with video content. In contrast, the asciinema platform provides a single integer value displaying the number of unique views for the demo. Despite this limitation, there are some benefits to collecting this metric.

The instructor uses a considerable amount of time and effort to produce the demos. Analyzing the number of views would give some indication of whether the students are utilizing them or not. Furthermore, the number of views per demo could reveal its popularity and students’ interest in certain parts of the course curriculum.

We examine the total number of views for each demo. The web site asciinema.org were the demos are hosted tracks the number of unique views per demo. It is the authors understanding that the site uses the IP address to distinguish between unique views. A user viewing the demo multiple times from the same IP address is counted as a single view.

VII. Results

Fig. 3 shows the responses to the closed-ended questionnaire, while Table [V - VIII] show the percent and frequency of the Likerk scale responses. In the next section we will present some of the responses to open-ended questions.

On **Q1**, regarding whether the demos improved the students understanding of the subject content, 55.8% of the participants responded Strongly Agree, while 41.9% respectively responded agree. As shown on Table V, a substantial percentage of the participants deemed the demos to be helpful to their understanding of the subject content.

Furthermore, whether the demos helped them conduct the weekly lab assignments easily, 48.8% and 41.9% responded Strongly Agree or Agree. The results on Table VI show an overwhelmingly positive response regarding **Q2**. Students consider the demos as an essential tool in helping them conduct the weekly assignments.

All though **Q3** had a high positive cumulative percent in terms of Strongly Agree and Agree, only 32.6% answered Strongly Agree. The result which is illustrated on Table VII can be interpreted that the students would like to see the documentation of the demos improved.

69.8% and 20.9% responded Strongly Agree or Agree to **Q4** on whether they would recommend other courses to adopt similar solution. The high positive response to the statement shown in Fig. 3 and Table VIII is strong evidence of the student's satisfaction with the demos and overall structure.

On the whole, candidates regarded the demos as a valuable supplement to the blended learning environment and appreciated the flexibility the solution offered.

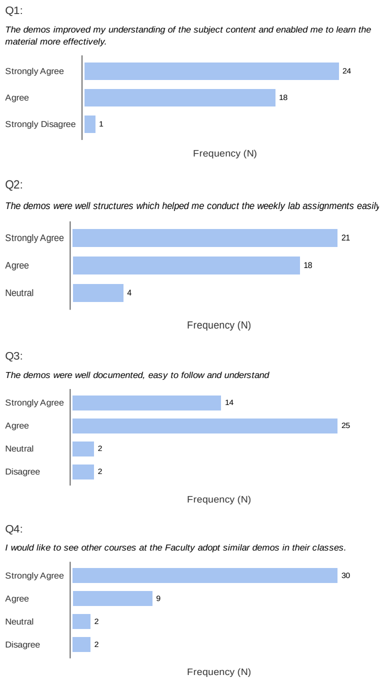


Fig. 3. Results of the closed-ended questions.

TABLE V
Percent & Frequency of the Likerk scale responses.

Q1	Frequency	Percent	Cumulative Percent
Strongly Agree	24	55.814	55.814
Agree	18	41.860	97.674
Strongly Disagree	1	2.326	100.000
Total	43	100.000	

TABLE VI
Percent & Frequency of the Likerk scale responses.

Q2	Frequency	Percent	Cumulative Percent
Strongly Agree	21	48.837	48.837
Agree	18	41.860	90.697
Neutral	4	9.302	100.000
Total	43	100.000	

TABLE VII
Percent & Frequency of the Likerk scale responses.

Q3	Frequency	Percent	Cumulative Percent
Strongly Agree	14	32.558	32.558
Agree	25	58.140	90,698
Neutral	2	4.651	95,349
Disagree	2	4.651	100.000
Total	43	100.000	

TABLE VIII
Percent & Frequency of the Likert scale responses.

Q4	Frequency	Percent	Cumulative Percent
Strongly Agree	30	69.767	69.767
Agree	9	20.930	90,697
Neutral	2	4.651	95,348
Disagree	2	4.651	100.000
Total	43	100.000	

A. *Measure of internal consistency*

Internal consistency is a measure used to show whether items on a questionnaire measuring the same perception or opinion generate consistent scores. If several questions expect to measure the same construct, a participant should answer these questions in the same way. That would indicate that the test has internal consistency. Cronbach's α and the average interitem correlation are often used to measure internal consistency [19].

TABLE IX
Scale Reliability Statistics.

	Cronbach's α	Average interitem correlation
scale	0.714	0.385

The 5-point Likert scale responses were analyzed for internal consistency reliability using Cronbach's α. We obtained an alpha score of 0.714. Values of α between 0.70 to 0.95 indicate that as a set, the items are closely related [19]. We also analyzed the average interitem correlation to measure the internal consistency reliability. As shown in Table IX, the coefficient value of 0.39 was obtained. A correlation coefficient between 0.15 to 0.50 indicates evidence of internal consistency.

B. Some qualitative student responses to the survey

Excerpts from the students' qualitative answers to the survey are presented in this section.

> ''I enjoyed the lab system with hands on practical examples. The demos gave a great introduction which made understanding the process easier.''
>
> ''I enjoyed the course because it was very structured and covered a lot of exciting topics in security. The demos and labs in Canvas were very helpful.''
>
> ''I generally liked the course. It gave lots of insight into potential problem with cybersecurity. The way Lab and Demos are structured is good. Never seen such a good way of giving students a hands on experience as this.''

Moreover, the word cloud in Fig. 4 was created from the students' responses to the qualitative open-ended questions. Word cloud is a visual illustration of word frequency. The more often the word appears within the text, the larger it appears in the image produced [20]. The words, demo, lecture, lab, assignment, course, good, structured and practical are clearly visible and enlarged.

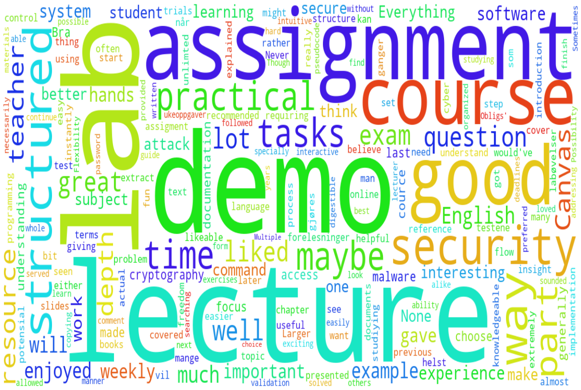


Fig. 4. Word cloud derived from responses of the open-ended questions.

Results from closed-ended questions, coupled with the open-ended responses, suggest that on the whole, candidates responded positively to the implemented solution.

C. Analysis of the quantitative data

Fig. 5 shows the total number of views for each demo ranging from the highest having 337 views and the lowest with 89 views.

As shown in, Table X, the 20 demos combined generated a total of 3743 unique views with a mean vale of 187.

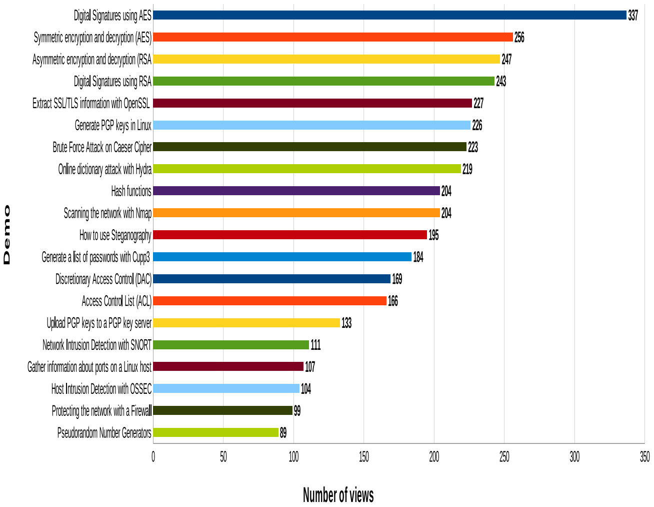


Fig. 5. Number of views per demo.

TABLE X
DESCRIPTIVE STATISTICS.

Variable	n	Mean	Std.	Min.	Max.	Range	Sum
	20	187	65	89	337	248	3743

We also observe from Fig. 5 that demos covering topics in the first 6 weeks have higher number of views than demos covering the last weeks of the semester. Particularly, topics related to cryptography in general and password attacking techniques seem to be more popular with students than network security, such as firewalls and intrusion detection.

The number of students attending the lectures and conducting the weekly assignments tends to decline as we progress through the semester. The drop in the demo views in the second half of the semester might be related to that.

Further analysis reveals that 18 of the demos received more than 100 unique views, of which eight received between 200 and 250, while 2 received more than 250 views. This is shown in Fig 6. While there are notable differences in the number of views per demo, the overall results indicate significant utilization of each.

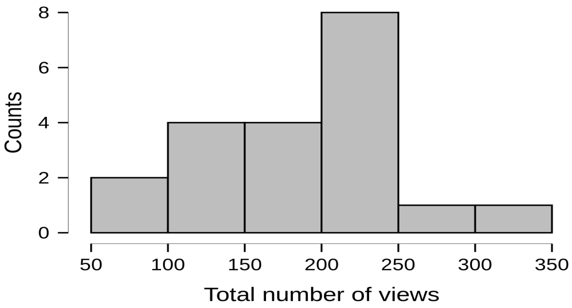


Fig. 6. Histogram of the number of views.

VIII. Limitations & Future Work

The author acknowledges that the sample size of this study is not large enough to determine and conclude the effect of the proposed solution significantly. However, our results are consistent with similar findings in related research, which reports a positive impact on short video tutorials has on the students learning.

In a large class such as ours with 161 students, it should be possible to get more participants. The author will investigate ways to increase the number.

The presented study is not without limitations. First, The participants were computer science students at one university. Whether the findings of this study can be generalized across a broader range of disciplines, needs further examination.

Second, the number of views a demo received was collected from the website asciinema.org. The site displays a simple numeric value that shows the number of unique visits. In other words, multiple views from the same IP address are considered a single point of view.

Unfortunately, the site does not provide detailed statistics on when the demo was shown and for how long. The author recognizes the need to collect more data that can further illuminate how the demos are used.

Fortunately, Asciinema offers an open source server solution to host the server on site. The instructor intends to explore the possibility of running an instance locally. By having direct access to the server, one can deploy open-source monitoring tools such as Prometheus and Grafana to collect important metrics. The instructor is also considering extending the survey to include questions about demo utilization.

IX. Conclusion

In this paper, we presented our experience with teaching a cybersecurity course to undergraduate students utilizing terminal sessions recording software as a pedagogical tool.

Results from the anonymous online survey indicate that overall, students appreciated the blended learning approach and deemed it as an essential enhancement to traditional face-to-face classes they previously encountered in other courses during their three years bachelor program.

Furthermore, the response from the participating candidates indicates overall satisfaction with the demos in particular and the ability to do weekly assignments at their convenience. The authenticity of the practical tasks simulating real attack and defense scenarios were also valued.

References

[1] K. S. Jones, A. S. Namin, and M. E. Armstrong, "The core cyber-defense knowledge, skills, and abilities that cybersecurity students should learn in school: Results from interviews with cybersecurity professionals," *ACM Transactions on Computing Education (TOCE)*, vol. 18, no. 3, pp. 1–12, 2018.

[2] W. Crumpler and J. A. Lewis, "The cybersecurity workforce gap," *Center for Strategic and International Studies, Washington, DC.[Online]. Available: https://www.csis.org/analysis/cybersecurityworkforce-gap*, 2019.

[3] D. L. Burley, J. Eisenberg, and S. E. Goodman, "Would cybersecurity professionalization help address the cybersecurity crisis?" *Communications of the ACM*, vol. 57, no. 2, pp. 24–27, 2014.

[4] K. R. Green, T. Pinder-grover, and J. M. Millunchick, "Impact of screencast technology: Connecting the perception of usefulness and the reality of performance," 2012.

[5] L. MacLeod, M.-A. Storey, and A. Bergen, "Code, camera, action: How software developers document and share program knowledge using youtube," in *2015 IEEE 23rd International Conference on Program Comprehension*. IEEE, 2015, pp. 104–114.

[6] Y. Pal and S. Iyer, "Effect of medium of instruction on programming ability acquired through screencast," in *2015 International Conference on Learning and Teaching in Computing and Engineering*. IEEE, 2015, pp. 17–21.

[7] C. Morris and G. Chikwa, "Screencasts: How effective are they and how do students engage with them?" *Active Learning in Higher Education*, vol. 15, no. 1, pp. 25–37, 2014.

[8] J. Mirkovic, A. Aggarwal, D. Weinman, P. Lepe, J. Mache, and R. Weiss, "Using terminal histories to monitor student progress on hands-on exercises," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 866–872.

[9] J. Bailey and C. Zilles, "uassign: Scalable interactive activities for teaching the unix terminal," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 2019, pp. 70–76.

[10] A. Dewald, F. C. Freiling, and T. Weber, "Design and implementation of a documentation tool for interactive commandline sessions," in *2011 Sixth International Conference on IT Security Incident Management and IT Forensics*. IEEE, 2011, pp. 62–80.

[11] M.-C. Monget, D. Bouillet, and D. Conan, "The initux system: A new way to learn gnu/linux," in *EdMedia+ Innovate Learning*. Association for the Advancement of Computing in Education (AACE), 2005, pp. 4659–4665.

[12] M. Bishop, D. Burley, S. Buck, J. J. Ekstrom, L. Futcher, D. Gibson, E. K. Hawthorne, S. Kaza, Y. Levy, H. Mattord *et al.*, "Cybersecurity curricular guidelines," in *IFIP World Conference on Information Security Education*. Springer, 2017, pp. 3–13.

[13] S. Mohorovičić and E. Tijan, "Blended learning model of teaching programming in higher education," *International Journal of Knowledge and Learning*, vol. 7, no. 1-2, pp. 86–99, 2011.

[14] U. Köse, "A blended learning model supported with web 2.0 technologies," *Procedia-Social and Behavioral Sciences*, vol. 2, no. 2, pp. 2794–2802, 2010.

[15] M. Kulik, "Asciinema," 2020, https://asciinema.org/, Last seen 10/04/2020.

[16] P. Hare, "Demo magic," 2020, https://github.com/paxtonhare/demo-magic, Last seen 10/04/2020.

[17] Oracle, "Virtualbox," 2020, https://www.virtualbox.org/, Last seen 10/04/2020.

[18] N. V. Ivankova, J. W. Creswell, and S. L. Stick, "Using mixed-methods sequential explanatory design: From theory to practice," *Field methods*, vol. 18, no. 1, pp. 3–20, 2006.

[19] K. S. Taber, "The use of cronbach's alpha when developing and reporting research instruments in science education," *Research in Science Education*, vol. 48, no. 6, pp. 1273–1296, 2018.

[20] R. Atenstaedt, "Word cloud analysis of the bjgp: 5 years on," *Br J Gen Pract*, vol. 67, no. 658, pp. 231–232, 2017.