Implementing Cybersecurity Exercises for Teaching Information Security

Kevin Ortiz-Sánchez
Master in Computer Science
Alfredo Cruz-Triana, PhD
Electrical & Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract *— To accommodate student demand and close the skills gap in the workplace, many colleges have created—or are in the process of creation—new courses or degree programs in cybersecurity. Cybersecurity encompasses extensive topics, which can be presented and taught in different ways. To complement Principles of Information Security and Computer Security courses, this paper presents cyber security exercises on various of the topics covered, taking an extra dimension to the students' education, in which they can apply what they've learned from the materials of the educators. Supplemented by presentations, examples, and detailed solutions to the main exercises, the approach to the exercises, are delivered from the objective of presenting a practical way to teach information security and computer security while staying in the technical specifications and depth of the courses.*

Key Terms *— Computer-Security, Cyber-Security-Exercise, Education, Information-Security.*

INTRODUCTION

The need to train more cybersecurity professionals is urgent. As a result, numerous educational efforts were launched to meet this need. In 2013, cyber-security was included in ACM/IEEE computing curricula [1]. The primary cybersecurity topics being discussed among curriculums are secure programming, network security and monitoring, cyber attacks, malware, hacking, offensive security and exploitation, human aspects, law, ethics, cryptography, authentication, and authorization [2].

The work developed to complement Principles of Information Security and Computer Security courses was comprised mostly of exercises. Given one of the primary tools used in cyber-security professional development is hands-on cyber-security exercises [3]. What makes cyber-security education a challenge is that a requirement for a top information security specialist is his practical experience in various security related tasks [4], but exercises and labs that may take an hour to complete, can span months to prepare and implement in a large scale or classroom size of participants.

Not even considering extra time for configuration of tools, and compatibility issues with students' equipment if a computer laboratory isn't implemented. To overcome these issues, all exercises can be completed either with online tools or through command line-tools, easing the students' experience, focusing on the technical details of what they have learned from the associated lecture topics.

When implementing these, learning objectives, legal, operational, and pedagogical challenges to create safe, secure, and reusability must be considered [5]. Which is why as technology changes and advances, so will cybersecurity education and the different methods used to teach it.

METHODOLOGY

The body of work created was composed of six documents: two presentations to educate and supplement on the topics, two documents with instructions and exercises, and two documents with step-by-step solutions to the exercises. Each of the exercise documents is composed of 7 exercises and contains the necessary instruction, as well as links to a Google Drive with the files required to complete the exercises. The topics of the exercises can be seen in Table 1.

Table 1
Exercise Topics

Topic	Number of Exercises
Cryptography	4
Steganography	3
Metadata	2
OSINT	1
Hashes	2
Public-Key Cryptography	1
Logs	1

The exercises were developed to be complementary material for Principles of Information Security and Computer Security courses. Table 2 shows how the material is distributed by topic.

Table 2
Exercises by Course

	Number of Exercises by Course	
Topic	Principles of Information Security	Computer Security
Cryptography	2	2
Steganography	3	-
Metadata	2	-
OSINT	-	1
Hashes	-	2
Public-Key Cryptography	-	1
Logs	-	1

Description of the Developed Exercises by Topic

Cryptography

Cryptography, the science of secret message writing [6], transforms information into an unintelligible form while it is being transmitted or stored so that unauthorized users cannot decipher it. It involves the use of mathematical algorithms and protocols to protect information and ensure confidentiality, integrity, and authenticity of data. With classical cryptography exercises, students can begin to understand the principles of information security and network security.

Cryptography Exercise 1

The Caesar Cipher is a substitution monoalphabetic cipher. Given that it is also one of the earliest ciphers and due to its simplicity, it's the first given to students. The method was named after the roman general Julius Caesar, who used this cipher to communicate secret messages during war.

The method consists of 'shifting' the letters of a normal alphabet, like the one shown in Figure 1, a certain number of places down. The shift in the cipher is known as its key and this key will be an integer from 1-25. The shift parameter is responsible for the transformation of the plaintext or the ciphertext. Based on the shift parameter, the alphabets that are aligned together will be rotated to the left or to the right, depending on the number of shifts.

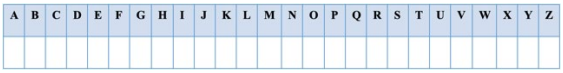


Figure 1
Untransformed Caesar Alphabet

Figure 2 demonstrates the transformation of the alphabet after the shift with a key of 3. The transformed alphabet now starts with the letter *D* where it was originally *A*. It's important to notice that once the transformed alphabet reaches the final letter, it will continue by using the letters from the start of the original alphabet until it fills the whole transformed alphabet and each letter from the original alphabet has a corresponding letter in the transformed alphabet.

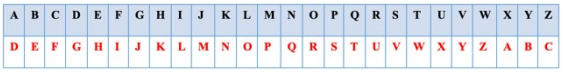


Figure 2
Transformed Caesar Alphabet with shift of 3

Given alphabets with a shift of 3 and the word *SECRET* as the plaintext, the cyphertext *VHFUHW* would be the result after the Caesar Cipher process as seen in Figure 3. Where *S* corresponds to *V*, *E* corresponds to *H*, *C* corresponds to *F*, *R* corresponds to *U*, and *T* corresponds to *W*.

S	E	C	R	E	T
V	H	F	U	H	W

Figure 3
Caesar Encryption

To decrypt, the same process would be applied but starting with the transformed alphabet and going to the untransformed alphabet, as seen in Figure 4. Students are to encrypt and decrypt using this cipher, being given the necessary information.

D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Figure 4
Transformed Caesar Alphabet with shift of 3

Cryptography Exercise 2

The Rail Fence Cipher is a transposition cypher that mixes the order of text. Its use of a key makes it harder to decrypt, as one same message can be mixed in different ways with this encryption method. Its name comes from the patterns it forms on tables once its methods are applied. The key to a Rail Fence Cipher corresponds to the number of rows the table will have. In Figure 5, a table with a key of 3 can be seen.

R				O				H				S
	U		T		T		E		I		L	
		N				H				L		

Figure 5
Rail Fence Matrix

Once the key is decided, the cells where the plaintext will go are marked, following a zig-zag pattern. Next, the order in which the plaintext will go into the table is marked, starting from the top left corner. In Figure 5, the populated table with the plaintext *RUN TO THE HILLS* can be seen. Finally, to encrypt, the text across each row is combined. First row being ROHS, second row UTTEIL, and third row NHL. Resulting in the ciphertext ROHSUTTEILNHL.

To decrypt, another key 3 table is created and \ the places where the ciphertext will go are marked by following the zig-zag pattern. This time, the table is populated in the order of the places where the text will go by row, starting from the top first row to the bottom row. Then the text is read in order of the original zig-zag pattern, returning the plaintext. Students are to encrypt and decrypt using this cipher, being given the necessary information.

Cryptography Exercise 3

The Playfair Cipher is a polygraphic substitution cipher that employs a 5x5 table. For the encryption or decryption process of the Playfair Cipher a 5x5 table (matrix) containing the *key word* needs to be created. Resulting in a matrix like the *Filled Matrix* in Table 3. Starting from an empty 5x5 matrix followed by one with the key and finally one with the rest of the alphabet characters not used in the key.

The key in the Playfair Cipher, unlike the past ciphers, is a word, in this example the key is *PLAYFAIR*. The letters of the key can't be repeated inside the matrix. The second A in *PLAYFAIR* is omitted on the *Matrix with Key* in Table 3 since it repeats. After the key is entered, the remaining spaces are filled with the rest of the alphabet. The matrix being 5x5, the letters *I* and *J* go in the same space. After matrix being populated, the encryption process may start.

Table 3
Playfair Matrix

Empty Matrix	Matrix with Key	Filled Matrix
	P L A Y F I/J R	P L A Y F I/J R B C D E G H K M N O Q S T U V W X Z

The next step in encrypting with the Playfair is the division of the plaintext into pairs of letters. If it's composed of an uneven number of letters, a single letter *X* is placed at the end. For this example, the plaintext *CARDS* is to be encrypted. Separated into pairs, *CA RD SX*. An *X* was added to complete the pair. To encrypt each pair, it must be

verified if the pair is in the same row, same column, or different row and column. Then, the encryption rules seen on Table 4 are applied.

- Same Row: If the two letters appear in the same row, each letter must be replaced letter with the one immediately to the right. If the letter to the right is at the end of the row, the replacement must continue with the first cell at the opposite side.
- Same Column: If the two letters appear in the same column, each letter must be replaced letter with the one immediately to the below. If there are no letters below, the replacement must continue with the first letter at the top of the column.
- Different Row and Different Column: If the two letters appear in different rows and columns, a rectangle must be created using both letters as corners. Then each letter must be replaced with its opposite corner in the same row.

Table 4
Playfair Encryption

Plaintext Diagraph	Square					Rule	Ciphertext Diagraph
CA	P	L	A→	Y	F	Different Row Different Column	BY
	I / J	R	B←	C	D		
	E	G	H	K	M		
	N	O	Q	S	T		
	U	V	W	X	Z		
RD	P	L	A	Y	F	Same Row	BI
	I / J	R →	B	C	D		
	E	G	H	K	M		
	N	O	Q	S	T		
	U	V	W	X	Z		
SX	P	L	A	Y	F	Same Column	XY
	I / J	R	B	C	D		
	E	G	H	K	M		
	N	O	Q	S	T		
	U	V	W	X	Z		

To decrypt, the same process is followed, but if the two letters appear in the same row, they are replaced by the letters immediately to their left. If the two letters appear in the same column, each letter is replaced with the one immediately on top. Students are to encrypt and decrypt using this cypher, being given the necessary information.

Cryptography Exercise 4

The Vigenère cipher is a polyalphabetic substitution cipher that incorporates 26 Caesar Ciphers in sequence with different shift values. To encrypt and decrypt, a table of alphabets is used, termed a tabula-recta or Vigenère square, seen in Figure 6. At different points in the encryption process, the cipher uses a different alphabet from one of the rows.

The alphabet used at each point depends on a repeating keyword. The top border row of the Vigenère Square corresponds to the key, which gets repeated if it's shorter than the text. The left outermost column border corresponds to the plaintext, and the intersection corresponds to the ciphertext. Students are to encrypt and decrypt using this cypher, being given the necessary information.

For example, given the key *COE* and the plaintext *PLANET,* the first character of the key and the first character of the plaintext are taken. Since *C* corresponds to the key, the *C* is marked on the top row and since the first character of the plaintext is *P*, the *P* is marked in the far-left column. Then, the intersection becomes the first character of the ciphertext, that being *R*, as seen in Figure 6.

Similarly, for the second character of the plaintext, the second character of the key (*O*) and the second character of plaintext (*L*) are taken. Then corresponding cells are marked, and the intersection points to the second character of the ciphertext. This will continue for the rest of the plaintext. The key is repeated once since the plaintext is six characters long and the key is three characters long.

To decrypt, the first character of the key and ciphertext are taken. The ciphertext character must be in the column of the key. When the ciphertext character is in the column of the key, the plaintext character at the left of that row signals the plaintext character for that sequence.

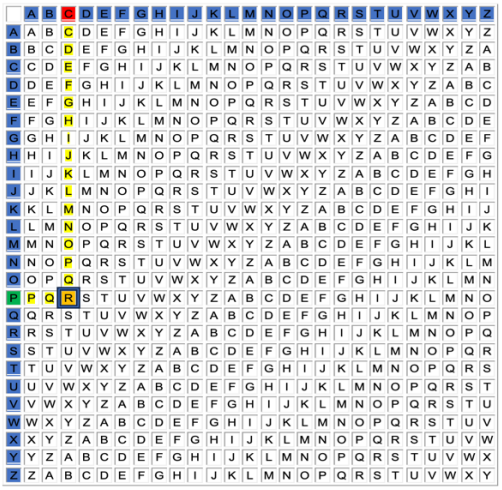


Figure 6
Vigenère Encryption

Steganography

Steganography [7] is the art or practice of concealing an image, file, message within another image, file, or message. Unlike cryptography, which focuses on making messages unreadable to unauthorized parties, steganography focuses on hiding the existence of the message itself. This distinction is why this topic comes next in the sequence.

Steganography Exercise 1

On this exercise, students are given the image *RaspberryPi4.jpg* and made aware that it has a hidden message in the form a string. It is embedded in the image's binary. Students need to analyze its contents in hexadecimal view either by a command line tool or hex editor.

A hex editor displays the contents of a file in hexadecimal format, which means that each byte of data is represented by two hexadecimal digits (0-9 and A-F). This format is useful for viewing and editing binary data because it provides a more concise and readable representation of the data than the raw binary format. With a hex editor, users can view and edit the contents of a binary file at the individual byte level. This allows users to make precise changes to the file, such as modifying specific bytes or replacing sections of the file with new data.

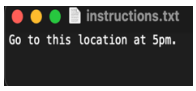
When analyzing the image through a hex editor, students will have three rows of information. The data position, the hex data representation, and the ASCII representation. The string is readable in the ASCII character translation of the hex dump. By going through the ASCII, students will eventually find the message "Meet me at the cafe" at the end of the file's bytes. The exercise serves to teach students how a message can be hidden inside a file that may look completely normal from the outside.

Steganography Exercise 2

One common trick in steganography is to hide a file inside of an image, not just a string of text. In this exercise students are given the image *background.jpg* and made aware that it has a hidden zip file. Students are informed on how to extract the contents by changing the extension of the image to *.zip*. Then, by unzipping the zip file, the embedded files are presented inside a document with the original file and the rest of the contents.

The embedded zip has a directory named *Secret-Message*, which contains an image file and a text file. Students are to explain the contents found in the embedded zip file, as seen in Table 5, the tools used, and process. This exercise shows students different file formats can be hidden inside of an image.

Table 5
background.jpg contents

File	Content	Description
background.jpg		Original image with embedded content.
instructions.txt		Message inside the *instructions.txt* file.
morro.jpg		Embedded image *morro.jpg* that accompanies *instructions.txt*.

Steganography Exercise 3

The objective of this exercise is to introduce students into File Format Headers and Footers, widely used in fields like Forensics. A file's header is the first few bytes of a file which contains identifying information about the type of file it is. While the footer is the last few bytes which signal the end of that file. The information between the Header and Footer signal to the computer where the file starts, ends, and the contents in between.

This exercise allows students to see how information can be hidden in a file by adding it after the footer. In this exercise, students are given an image file and instructed to locate and retrieve the hex after the footer when they see a new header. They then create a *.jpg* file based on the header of the hidden file with the retrieved hex data to reveal the concealed image.

The image is retrieved by opening the file on a hex editor and locating the JFIF header of a *.jpg* file after the original image's footer. The location of that header is in the *00008270* data position. By selecting all the hex data from the *00008270* data position to the *000096A0* data position, students now have all the contents of the hidden *.jpg* file. They may then export the selected bytes to a file and change the extension to *.jpg*, getting the hidden image.

Metadata

Metadata refers to information that describes other data. In other words, it's data about data. If a picture is taken with a smart-phone, metadata of that picture might include information such as the date and time the picture was taken, the phone model, settings used, the resolution of the image, and even the GPS coordinates. This means that all information published on social networks, discussion forums, and group chats, among other sources, is free and accessible to anyone, considering the restrictions that may apply [9].

Metadata Exercise 1

In this exercise students are given the image *cat.jpg* seen in Figure 7 and instructed to get the metadata from the picture by using programs, websites, or terminal commands. The purpose of this exercise is for students to see how much information can be found from the properties of a file. Also allowing students to distinguish the different information that can be expected depending on the file. Given students will be asked to find metadata from a different file type in the next exercise.

Figure 7
cat.jpg

To get the metadata from the picture *cat.jpg*, the command-line application *ExifTool* is used. Which can read, write, and edit metadata information in a wide variety of files. By running the *ExifTool* command, the metadata shown in Figure 8 is obtained. In which, for example, it can be seen the image is 2.2 MB in size and is a HEIC file type. Similar results can be gathered from online tools and other programs from which students can choose. At the end, students are expected to provide the metadata shown in Table 6 from the *cat.jpg* file.

```
(base) kevinortiz@Kevins-MacBook-Pro Desktop % exiftool cat.png
ExifTool Version Number         : 12.42
File Name                       : cat.png
Directory                       : .
File Size                       : 2.2 MB
File Modification Date/Time     : 2023:09:14 18:59:58-04:00
File Access Date/Time           : 2023:10:16 14:15:12-04:00
File Inode Change Date/Time     : 2023:10:16 14:14:57-04:00
File Permissions                : -rw-r--r--
File Type                       : HEIC
File Type Extension             : heic
MIME Type                       : image/heic
```

Figure 8
Exif-Tool

Table 6
cat.jpg Metadata

Maker of Device	Apple
Model of Device	iPhone 8
Software Version	16.0
GPS Position	18 deg 25' 19.64" N 66 deg 3' 21.16" W
Create Date	2022:10:11
Create Time	18:05:35
Flash (On/Off)	Off
Lens Model	iPhone 8 back camera 3.99mm f/1.8

Metadata Exercise 2

This exercise looks to show the difference in metadata depending on the file being analyzed. In this case, a Word document. Documents from Microsoft Office Suite are essentially compressed files working together. By the extension being changed to .zip and then being unzipped, it decompresses the Word document and different files are extracted that make up the document whose content is described in Table 7.

Table 7
Word Elements Descriptions

Element	Description
app.xml	This element specifies the app information to a Word document. Like application, company, and version.
core.xml	This element contains the core properties to a word document. Like creator, subject, number of revisions, and date created.
document.xml	This element specifies the content within a Word document with its fonts and styles.
fontTable.xml	This element specifies the font settings that are applied to a Word document.
media	This directory contains all the media to a word documents.
settings.xml	This element specifies the settings that are applied to a Word document.
styles.xml	The styles element contains all the style settings applied to a Word document.
webSettings.xml	The webSettings element contains all the web page settings applied to a Word document.

Students are to go through these files and collect the metadata of the Word file and fill in a table like the one shown in Table 8. Although some of this information can be gathered without going through this process, this enables ways to get metadata from files from which properties can't be accessed or are obfuscated. Serving as the steps to create scripts to get metadata from many files at once [8].

Table 8
Word Metadata

Title	Secret File
Subject	Social Engineering
Creator	Kevin Mitnick
Last modified by	Microsoft User
Create date and time	2022-10-19T03:16:00
Total Time	6
Application	Microsoft Office Word
Company	KnowBe4
Pages	1
Words	105
Characters	559
Paragraphs	4

Open Source-Intelligence

Open-source intelligence, or OSINT, is a technique used to obtain data from open sources. These sources could be, for instance, government databases, websites, or phone books.

Open Source-Intelligence Exercise 1

The objective of this exercise is to familiarize students with the possibilities of Open Source-Intelligence. Students are given images to reverse search using *TinEye* or *Google.* Google has a service called *Google Search by Image* that allows a user to search for images using an image as the starting point, rather than a written or spoken search query seen in like in Figure 9. Similarly, *TinEye* is a reverse image search engine that serves the same purpose as *Google Search by Image*.

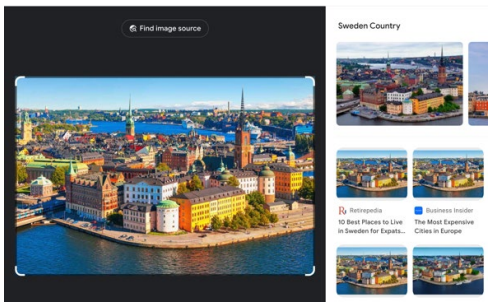


Figure 9
Reverse Image Search

Students are also given public IPs to look up details as seen in Figure 10. One of the most popular IP-related queries is geolocation search. Dozens of different resources offer a general

location, including a city and a specific area. This technique may be used to fight against illegal spammers and hackers or to locate the sources of a problem. Although it would be nice to be able to find the precise location of a visitor, it is very difficult to find the exact location of a host given its IP address. However, there are tools available to help identify the approximate location of the host. Showing how an investigation can be done without the need of private information or private tools.

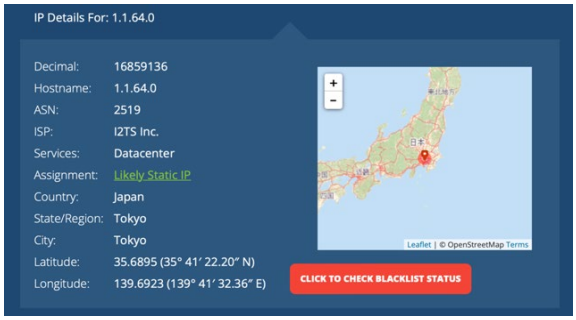


Figure 10
Public IP look-up

Hashes

A hashing algorithm uses mathematical functions that takes data and makes it unreadable. Hashing algorithms are one-way programs, the data can't be unscrambled or decoded. These algorithms are mostly used to store data.

Hashes Exercise 1

The objective of this exercise is to familiarize students with hashing and the differences between one-way and two-way functions. Students are given different MD5 hashes from which they need to find the original plaintext. The hashes are given in the context that they are simple one-word passwords that can be solved with online tools. Since by being hashes of weak passwords, they are available in tools or websites with password dictionaries.

A password dictionary is a massive list of expected passwords used to quickly crack or guess actual passwords. These lists can include words in the form of dictionary words, common passwords, iterations of common passwords, and exposed passwords. They can also contain passwords that used to be hashed but have been subsequently cracked because they were stored in a weak password hashing algorithm, highlighting the importance of strong passwords.

Hashes Exercise 2

In this exercise, students are given different posts from a social media profile (created for the purpose of the exercise), from which they need to find potential key words or information from the user to create potential passwords to this user's profile as seen in Figure 11.

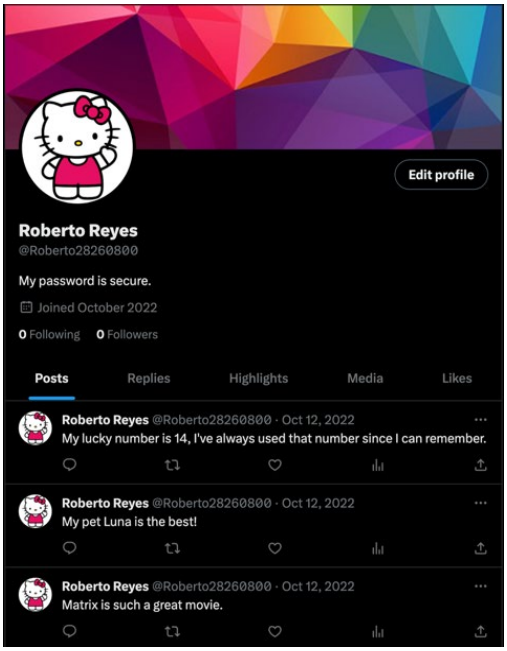


Figure 11
Twitter Account

By going through the account's posts, different keywords that can be in the user's password are gathered. In this case *14*, *Luna*, and *Matrix*. Students are instructed to create passwords combining two keywords. They will then hash the potential passwords and compare the hash with a previously given hash that corresponds to the user's password as seen in Table 9. It helps the students to learn how a combination of words and numbers creates a stronger password that is harder to crack

and why they shouldn't use personal information for passwords or share that information.

Table 9
Account Password

Keywords	MD5 Hash	Potential Passwords	Password
14	371be0e4b6 378f40b9576 ccb25397b8 95327789b	14Matrix	Matrix14
		Matrix14	
Luna		MatrixLuna	
		LunaMatrix	
Matrix		14Luna	
		Luna14	

Public Key Cryptography

In Public Key Criptography both users have a pair of unique keys, as seen in Figure 12: a private key and a public key. Public Keys are used to encrypt while private keys are used to decrypt. Public keys can be shared with no fear or compromise because they are only used to encrypt, not decrypt.

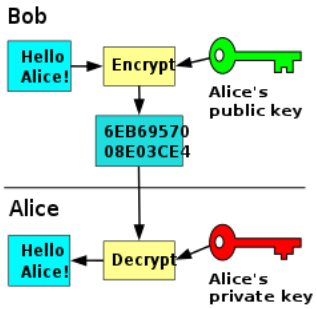


Figure 12
Public Key Encryption

Public Key Criptography Exercise 1

The purpose of this exercise is for students to understand the difference between private and public key cryptography. In this exercise, students are given an RSA encrypted message and various private keys. Students are to find which private key decrypts the message with online tools for RSA decryption.

The RSA encrypted message is as follows:

dDtbGWZkYSGU3u3PqKSedOpUH49VvOIYmnTJiuzMuf2iEr9R 8U8mqMqAQG2q14yHjhHon/LsYiSDsv7TOI/SS9pOEjUCrVj5/b LlxRgdp7TBpBztm0JZyuiHfNdQxh/+PkEMrdDKo7fAZqi9NHCJ 7EB3y+BBS4xw8DZLcA3eENs=

Students need to take the encrypted message and enter it in the Encrypted Text to Decrypt box. Then take one of the private keys and enter it in the Enter Public/Private key box. Next, select Private Key as the RSA Key Type and press Decrypt. The message "Keep your private key private, wouldn't want others to have access to these messages." will appear if the correct private key is used. Serving to highlight why the private keys are to be kept private.

Logs

Logs are records of events within a system or network. Composed of log entries with information related to a specific event. This can range from event logs (record of network traffic), server logs (tracks actions on a server), system logs (operating system actions), email logs (email data), and any other events from which records are kept.

Logs Exercise 1

In this las exercise, students are given email logs from a Google account. Users are to open the file and see what information can be found about the user and his conversations. Using keywords to search through the log like 'from:', 'to:', and 'Message-ID', as seen in Table 10. Given that the keywords used are not technical for the students, they can focus on the task of looking for information and understanding the process of analyzing logs. Serving as experience for more difficult log analysis cases like event logs, system logs, and server logs.

Table 10
Log Fields

Filed Name	Description
From	E-mail address of the authors of the e-mail
To	The e-mail address of the message recipient
Cc	Carbon Copy
Bcc	Blind Carbon Copy
Subject	A summary of the topic
Date	The local date and time when the e-mail was written
Reply-to	Address that the e-mail reply will redirect to
Message-ID	Globally unique message identification string generated when sent

CONCLUSION

Through the course of this paper, the benefits of incorporating cybersecurity exercises into the curriculum have been explored, from enhancing students' practical skills to a better understanding of the threats they may face. This paper has also highlighted the significance of hands-on learning and the value of exercises in enabling students to apply theoretical knowledge by providing the opportunity to engage in cybersecurity exercises, computer professionals are not only equipping them with essential technical skills but also practical experience for them to continue into more in-depth problems, courses, and tools. By continually adapting and enhancing our educational approaches, colleges and universities can better prepare the next generation of cybersecurity professionals to meet these ever-evolving challenges.

Future Work

Some comments from and shortcomings of this project, which are worth noting, are:

- The exercises cover some but not all the topics of the courses. Given that some topics are based on security guidelines and others need a more in-depth laboratory setting.
- The use of more tools like the Kali Linux operating system.
- Real life settings or exercises where multiple topics are required to solve one exercise.
- A summative evaluation should be done to measure the impact of the exercises on student learning.

REFERENCES

[1] S. Roach and M. Sahami, "CS2013: Computer Science Curricula 2013," *Computer*, vol. 48, no. 3, pp. 114-116, 2015 [Online]. Available: DOI: 10.1145/2534860. [Accessed: Oct. 18, 2023].

[2] V. Švábenský, J. Vykopal, and P. Čeleda, "What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences" 51st ACM Technical Symposium on Computer Science Education, 2019. [Online]. Available: DOI: 10.1145/3328778.3366816. [Accessed: Oct. 18, 2023].

[3] M. Mudassar and K. Basel, "Inefficiencies in Cyber-Security Exercises Lifecycle: A Position Paper" AAAI Fall Symposium, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:58005747. [Accessed: Oct. 18, 2023].

[4] A. Furtună, V. Valeriu, and I. Bica, "A Structured Approach for Implementing Cyber Security Exercises" 8th International Conference on Communication, pp. 415-418, 2010. [Online]. Available: DOI: 10.1109/ICCOMM.2010.5509123. [Accessed: Oct. 18, 2023].

[5] J. Marquardson and D. Gomillion, "Cyber Security Curriculum Development: Protecting Students and Institutions While Providing Hands-On Experience" Information Systems Education Journal., vol. 16, no. 5, pp. 12-21, 2018. [Online]. Available: http://isedj.org/2018-16/ ISSN: 1545-679X. [Accessed: Oct. 18, 2023].

[6] S. Garera, and J. Vasconcelos, "Challenges in teaching a graduate course in applied cryptography." ACM SIGCSE Bulletin, vol. 41, no. 2, pp 103-107, 2009. [Online]. Available: DOI: 10.1145/1595453.1595485. [Accessed: Oct. 18, 2023].

[7] P. Uddin, M. Saha, and S. Jannatul, "Developing an efficient solution to information hiding through text steganography along with cryptography." 9th International Forum on Strategic Technology, 2014. [Online]. Available: DOI: 10.1109/IFOST.2014.6991061. [Accessed: Oct. 18, 2023].

[8] S. Marinai, "Metadata Extraction from PDF Papers for Digital Library Ingest." 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 2010. [Online]. Available: DOI: 10.1109/ICDAR.2009.232. [Accessed: Oct. 18, 2023].

[9] J. Herrera, P. Gaona, and S. Sánchez, "Open-Source Intelligence Educational Resources: A Visual Perspective Analysis." Applied Sciences, vol. 10, no. 21, 2020. [Online]. Available: DOI: 10.3390/app10217617. [Accessed: Oct. 18, 2023].