Fall 2023

Docker Technology for Small Scenario-Based Excercises in cybersecurity

Zeinab Ahmed

COLOMBUS STATE UNIVERSITY

DOCKER TECHNOLOGY FOR SMALL SCENARIO-BASED EXCERCISES IN CYBERSECURITY

A THESIS SUBMITTED TO
THE COLLEGE OF BUSINESS
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

BY
ZEINAB AHMED

COLUMBUS, GEORGIA
2023

DOCKER TECHNOLOGY FOR SMALL SCENARIO-BASED EXCERCISES IN CYBERSECURITY

By
Zeinab Ahmed

Committee Chair:
Dr. Yeşem Kurt-Peker

Committee Members:

Dr. Lydia Ray
Dr. Lixin Wang

Columbus State University
December 2023

ABSTRACT

This study aims to better prepare students for cybersecurity roles by providing practical tools that bridge the gap between theory and real-world applications. We investigate the role of small scenario-based exercises for students' understanding of cybersecurity concepts. In particular, we assess the use of Docker technology to deliver training that includes a simple small scenario on html code injection. The effectiveness of scenario-based learning has long been defined and by using SBL, we are going to create hands-on activity that involves the fundamental topics in cybersecurity using Docker technology, allowing students to see the exploitation of the vulnerabilities and defense mechanisms against the attacks. The study's results deduced from the analysis of responses from two surveys favored small scenario-based exercises for hands on activities in cybersecurity and the use of Docker technology for building such scenarios.

By underlining the links between the studies discussed in the paper and our study, it shows the importance of scenario-based learning in cybersecurity education, highlighting its hands-on and practical nature. Also, it advocates for tailored educational programs that effectively integrate theoretical knowledge with practical application, emphasizing the value of this approach in preparing students for real-world cybersecurity challenges.

INDEX WORDS: Docker, SBL, cybersecurity, education, HTML, scenario-based learning.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The reliance on technology and connectedness in almost all aspects of life comes with a major challenge which is assurance of security and privacy. Cyberattacks are non-stop and threats are evolving as technology evolves. There are two crucial factors in combatting cybersecurity threats and attacks which are user awareness and mission-ready workforce.

In this study, we focus on workforce development, particularly education that prepares students to take on cybersecurity roles upon graduation. Cybersecurity is a broad area of study that covers technical concepts such as network security, log analysis, digital forensics to name a few, and non-technical concepts such as policies, legal aspects, risk management. Cybersecurity curriculum in higher education, in some cases, attempts to cover a broad range of topics and other cases, focuses more on the technical. Because of the use of computers in technology and communications, cybersecurity professionals need to have a good understanding of the fundamentals of computers and programming, and how various components of a computer system, a network, or an application work together.

Content knowledge is an essential part of understanding the fundamentals. In addition, students should have hands-on activities that help them see the connections between various parts of a computer system, network, or an application. A significant amount of work has been done to provide hands-on experiences to students. Netlab, virtualization technologies such as VMWare and VirtualBox, scenario-based activities created by these tools. Simulations in cyber ranges and individual efforts by faculty and professionals are some of the ways hands-on activities are integrated into curriculums.

In this study, we explore the use of the relatively new Docker technology for creating scenario-based hands-on activities to be integrated into introductory cybersecurity curriculums. We develop a hands-on activity to help students learn about HTML code injection and its prevention along with the Docker technology. We assess the effectiveness of Docker-based small scenarios by using a survey instrument where we collect information about students' knowledge of Docker, web servers and applications and web security before the training as well as their view of the effectiveness of training after they go through the training.

CHAPTER 2: OVERVIEW AND BACKGROUND

Since the main focus of the study is education that prepares students to embark on cybersecurity roles, this chapter presents the most-commonly used tools for hands-on activities, namely Netlab and VMWare and VirtualBox, along with their advantages and limitations. Moreover, it elaborates on Docker Technology and how it is an improved approach in terms of application portability, scalability, resource efficiency, and consistency across many settings.

2.1 Tools for Hands-on Activities

Netlab and some virtualization technologies such as VMWare and VirtualBox are some of the ways the hands-on activities are integrated into curriculums. Netlab is an example where students connect to previously configured virtual machines on a web portal and follow step-by-step instructions provided to perform some security tasks. It is a great tool for students to get familiar with some of the security tools in a safe environment. Another advantage is that it is system agnostic. A browser is all one needs to connect to the network. All students in the class get the same network with the same configurations to start with. However, Netlab has limitations depending on who is using it. First, it is a paid product and not all institutions have the funds to purchase it. Moreover, when some systems, such as a server in a pod, do not work in Netlab, we cannot fix those easily. students cannot connect to the Internet while working on Netlab. Thus, they cannot freely explore many of the tools available in Netlab. Also, the labs are prescribed, and making changes in the lab or creating new labs requires expertise and time commitment on the instructor's side, and time is usually scarce for an instructor.

Another commonly used technology for hands-on activities in cybersecurity is virtualization technologies such as VMWare and VirtualBox. VMware is widely recognized for its robust enterprise-level features, providing advanced functionalities and performance optimization. It excels in scalability and is often preferred in large-scale corporate environments. However, one notable challenge is its cost, making it more suitable for well-established businesses with substantial budgets. On the other hand, VirtualBox is an open-source solution known for its user-friendly interface and flexibility, making it accessible to a broader audience, including individual users and small businesses. Its main challenge lies in scalability, as it may not perform as seamlessly in resource-intensive scenarios as VMware. The challenge with these technologies is that they may get resource intensive. Each virtual machine in a network of these technologies requires resources to be allocated only to itself.

2.2 Docker Technology and its use

The creation, distribution, and management of software programs within portable, isolated environments known as containers are made possible by the open-source technology known as Docker. The software, libraries, and dependencies needed for an application to function reliably across various computer environments are packaged into self-contained units called containers. Docker, as a lightweight containerization platform, simplifies the deployment of isolated and reproducible environments, minimizing the risk of system conflicts and ensuring consistent setups for learners. Thus, learners can practice authentic scenarios of malware analysis, network reconnaissance, or vulnerability exploitation without impacting the host system or the integrity of other exercises. This allows learners to gain hands-on experience with different systems and configurations, deepening their understanding of cybersecurity concepts.

Software development and deployment across a range of infrastructure and operating systems are made simpler by Docker, which offers a standardized approach to create, distribute, and operate programs. It makes use of containerization technologies to keep consistency and reproducibility and to separate the application and its dependencies from the underlying system.

Improved application portability, scalability, resource efficiency, and consistency across many settings are all advantages of adopting Docker. Collaboration between the development and operations teams is made easier, compatibility problems are decreased, and the deployment process is made simpler.

Overall, by utilizing containerization technology, Docker offers a flexible and effective way to manage and deploy software, making it a popular choice for development workflows across a range of sectors. ￼

2.3 Problem Statement and Research Questions

This study investigates the utilization of Docker technology in crafting scenario-based hands-on activities for incorporation into introductory cybersecurity curriculums. The focus is on developing a hands-on activity aimed at enhancing students' knowledge of HTML code injection and its prevention along with understanding the Docker Technology.

The questions we'd like to answer in this study are:

- How effective is the integration of Docker technology in creating small scenarios for cybersecurity education?
- What are students' perceptions of the effectiveness of the training after participating in the hands-on activity?

CHAPTER 3: LITERATURE ON SCENARIO-BASED EXERCISES

The literature review includes the examination of 5 papers whose different approaches are scenario-based and whose findings are its favor, despite the limitations. Those papers are concerned with scenario-based approach for cyber security vulnerability analysis in a production facility, cybersecurity scenarios in college classes at Georgia Gwinnett College integrated with the American Government courses, scenario-based inquiry for engaging in general education computing, accessing competencies using scenario-based learning in cybersecurity, modeling effective cybersecurity framework: A Delphi method-based study, and scenario-based cyber defense education system on virtual machines integrated by web technologies.

In [1], Paul Baybutt described an approach for identifying the vulnerabilities of computer systems used in manufacturing and process plants. The author presented various approaches developed to perform Security Vulnerability Analysis (SVA) in a manufacturing and production facility. The approaches used scenarios similar to the concept of Scenario-Based Learning. Using SVA allowed them to know the deliberate ways to cause harm or attack and to take action against it. The SVA method was presented to provide an analysis of the facility security's posture and potential vulnerabilities which may include incomplete documentation, lack of training, or insufficient resources. It was also supporting an existing SVA which is the scenario-based method. This paper describes the CSVA approach that has been incorporated into Process Vulnerability Analysis (PVA), a Primatech SVA (which is the company that created the PVA approach) approach that is similar to a scenario based SVA approach.

In [2] David et al. shared their work integrating cybersecurity scenarios in college classes at Georgia Gwinnett College (GGC), a four-year public institution in Georgia. The authors used scenario-based simulations provided by the National Integrated Cyber Education Research Center (NICERC) and integrated them into their American Government courses taken by students of all ages and majors as well as in the Political Negotiation course which is an upper-level course for Political Science and Criminal Justice students. The scenarios provided by NICERC were developed by multi-disciplinary faculty teams for cybersecurity summer high school camps. They comprised a series of “briefings” provided to a special investigative team over a while. The scenarios were based on actual economic and political events in the news and included well-known politicians, heads of state, and geo-political organizations. The authors adapted this existing curriculum for college class use, expanding the reach beyond the summer high school program.

The authors conducted a pre-and post-survey to assess students' attitudes towards computing, technology, their precipitation for computer literacy courses, and scenarios used for inquiry-based learning. Out of 130 students who participated in classes, 82 of them signed the consent form. Thus, the reported results are from 82 participants. Even though the authors do not provide a quantitative analysis of their results in their article, based on student responses to the open-ended questions, they state that the scenario-based activity worked and was found effective in providing a unique learning experience for students.

Ghosh et al. [3] stated that there is a real disconnect between what students are learning and what is expected from them in the real world. The traditional way will leave a gap regarding what students are learning and what their employers are expecting from them and for this reason, many universities have been engaged in designing and delivering scenario-based curricula that impart knowledge, skills, and competencies. NIST published a draft of the NICE Cybersecurity Competency workforce framework competencies as a good starting point for the US to hire and attract a Federal Cybersecurity Workforce and a list of competencies. The authors assessed the competencies by developing an approach to define the latest skill and knowledge requirements needed by individuals to improve the security posture of their organization. The model complements the framework by including competencies needed by the workforce and those needed by cybersecurity professionals. The article proposes two approaches with which to assess competencies using scenario-based learning. Both approaches map to the NICE Cybersecurity Workforce Framework. The first approach starts with the knowledge, skills, and tasks to satisfy the chosen competency. The scenarios in this approach contain guiding questions that lead learners to make progress as they work on the scenario. Each answer is mapped to a set of knowledge, skills, and tasks thus assessing whether the students can perform the task and realize the relevant skills and knowledge. The second approach uses the revised NICE Cybersecurity Workforce Framework as a guide and starts with a work role for the chosen competency. The scenarios list tasks that are needed for the chosen work role and achieve the chosen competency. This is followed by sets of knowledge and skills needed to perform relevant tasks. Both approaches impart skills and knowledge to students by providing real-world examples that would help them recreate cyberattacks and purpose solutions.

Nabin et al. [4] introduce another example of a scenario-based method. Their main work was developing a CS training framework reliant on the current progress in research in learning theory and its application in education and training. They found related work on the same topic as they came up with CS frameworks. They conducted reviews of CS training offerings which include CS training framework among other types of offerings with a focus on training offerings for critical infrastructure (CI) protection. Based on the findings of their literature review, delivery methods that offered hands-on experience were preferred over the traditional ways of learning. Simulation-based learning or game-based learning was shown to be a popular CS training tool for both CI training and CS general training. In this article, an agreement was concluded on which solution should be considered.

Yet further research was needed to figure out how to optimally integrate the desired attributes found across different proposals to solve the traditional way of learning and have the best solution. Therefore, the authors introduced the Delphi method which is a Cybersecurity (CS) training framework based on a revised version of the ADDIE (Analysis, Design, Development, Implementation, and Evaluation) which is a learning model used by instructional designers and training developers to create effective learning experiences and more recent research personalized learning theory. The Delphi model was used to develop and validate our decisions that are gathered from experts using a questionnaire or a survey during the development of the training framework model. The results of the decision of the Delphi method would later be compared to recommendations to create a finalized framework as mentioned in the paper. This article shows the major differences from other CS training framework models. The use of the Delphi method and the involvement of expert stakeholders from different sides of academia and industry gave wide insights into current needs and recommendations for CS training as well as formal validation for the final development.

After reaching an agreement on the selected topics, the information collected was utilized to develop a model for CS training framework that followed the recommendation found by the authors in the literature and the recommendation they got from experts. This made the authors create a more effective CS training framework. According to Personalized Learning Theory (PLT) and by focusing on each participant's objective and preferences, the overall learning process would be improved to be more effective. By using the Delphi method, they were able to obtain feedback from a panel of experts with different CS backgrounds, understand which aspects each group saw in need of prioritization, and finally, reach an overall agreement.

In [5] article. Seongkyu et al. highlighted security issues that increased with the distribution of social media content. To be able to protect users' privacy and identity in social media networking, the authors used active learning. They developed scenarios based on specific malware such as Meltdown, Mirai Malware, Carbanak APT (Advanced Persistent Threats), and Ransomware scenarios that recreated a real multimedia content distribution. They also built and stored individual virtual environments for each scenario from the scenarios they developed for each malware integrating them on VMWare ESXI which is an Enterprise developed by VMWare to deploy and serve virtual computers. This way the attack and defense could be conducted in an environment similar to real-world networks.

Related work to the authors' work is that scenarios were constructed with less information to understand the nature of the problem along with the study cases and trends of the incident that would become a scenario. Researchers also had an environment for students who want to be information security operators or information security experts. The information protection education system was one of the systems that was made to help the students as a hands-on cybersecurity education which was an environment that provided the students with educational content related to the prevention of cyber incidents, countermeasures against cyber incidents, and learning problems. As a result, learners could develop different abilities and practical skills to cope with cyber incidents.

Using VMW provided an educational environment without the installation of any additional software. This increased learning efficiency by focusing on education without needing to build a new environment for each scenario. What needed to be done was to give more attention to analyzing, evaluating, and providing customized training.

Based on all the aforementioned studies undertaken in different contexts, the scenario-based activity was found effective in providing a distinguished learning experiences for students, improved their skills and knowledge by providing real-world examples that helped recreate cyberattacks and purpose solutions. As a result, learners could develop different abilities and practical skills to cope with cyber incidents.

CHAPTER 4. METHODOLOGY

In this section we provide the steps we followed in the training

- Develop a small scenario leveraging Docker technology for the training exercise.
- Create assessment instruments in the form of surveys to collect student background information and perceptions. Seek Institutional Review Board (IRB) approval for the study.
- Prepare the training content, incorporating the scenario developed in the initial step.
- Following IRB approval, we conduct the training sessions with the participating students.
- Analyze the results obtained from the surveys, focusing on participants' familiarity with the covered topics and their perceived effectiveness of the training.

Training Steps

1. Students will be introduced to Docker Technology using a PowerPoint Presentation.
2. Docker Desktop is already installed on the lab computers.
3. Students will be introduced to Web servers, Web applications and Web security.
4. Students learn to build containers and map it with a port number.
5. Students will be introduced to a simple scenario which includes a website with simple graphics.
6. Students connect to the website using the browser or the port number in Docker Desktop.
7. Students take some time to explore and attack the website.

8. The vulnerability will be explained to the students (the HTML code will be explained, and the attack will be demonstrated)
9. Students will be asked to come up with solutions
10. A secure method will be introduced and explained to the students to prevent HTML code along with other preventative methods for their information
11. They will connect to the website after applying the Secure method and try the attack.
12. Finally, students will understand the attack and how to prevent it using one of many ways.

CHAPTER 5: THE TRAINING

This chapter thoroughly explains the training scenario and the prerequisites for the participants, as well as its delivery in terms of the number of participants, the consent form that had to be signed by participants, the surveys developed to collect relevant data from the participants, and the different parts the training.

5.1 Training Scenario

Prerequisites: Participants have a basic understanding of security, web applications, web servers, and HTML.

A straightforward and user-friendly scenario was created for the training exercise. The scenario was tailored to provide a clear and accessible context, facilitating an easier understanding of what happened and what they should do. The aim was to create an environment where participants could navigate and engage with the material with ease, fostering a smoother learning experience. The simplicity of the scenario was intentional, ensuring that participants could readily grasp the content and apply their knowledge in a manageable and user-friendly setting.

The "Columbus Flower Shop" is an online platform that allows customers to browse and check a wide variety of flowers and floral arrangements. Recently, the website has come under scrutiny due to reports of security issues. The management of Columbus Flower Shop has decided to launch an investigation to identify and address the issues on their website.

5.2 Training Preparation:

The participants were students in the Computer Science department who have fundamental knowledge of computers, computer networks, and cybersecurity. The number of students who volunteered and approved to participate in the training was 9 students.

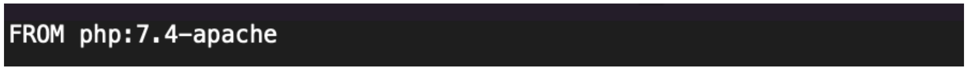
The Informed Consent form was provided to the participants as a pdf file at the beginning of the first session of the exercise. Those who attended the session received a link to the pdf in the chat utility of the platform (Microsoft Teams). Students were given 5 minutes to read the form. The time extended to for those who needed more time.

Responses to the consent form were collected via Microsoft Forms. The informed consent form is included in Appendix 4. Only those participants who provided their consent received the survey links via the email addresses they supplied, as an integral component of the study.

The commencement of the training involved the formulation of a scenario, strategically designed to uphold an element of surprise, and maintain the participants' unawareness of the impending cybersecurity training. To set the stage, we commenced the session by providing an overview and basic knowledge on Docker technology, encompassing the intricacies of containers, web applications, web servers, and the procedural aspects of their creation. Subsequently, participants were guided through the intricacies of the crafted scenario, which specifically illustrated the mechanics of HTML injection.

We used the Php:Apache image available in the Docker hub. It contained resources such as PHP and Apache configurations.

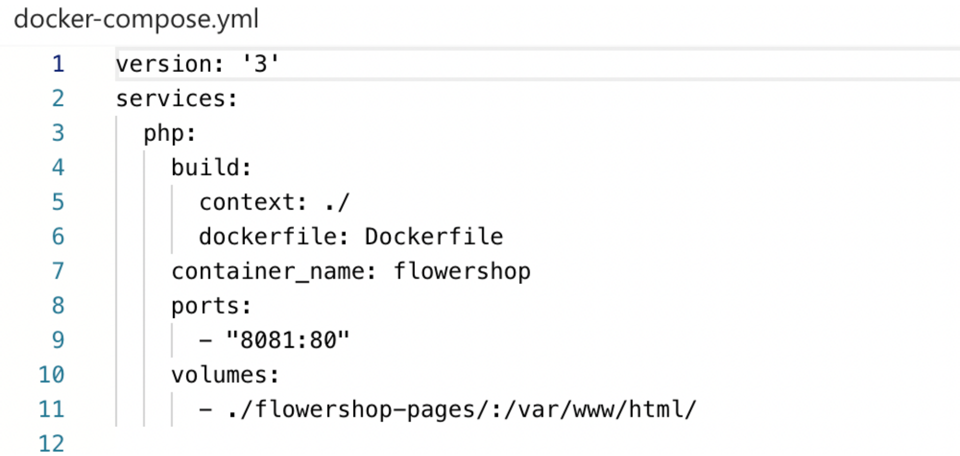
This image provided the foundational elements needed to build our container, and we subsequently constructed our Dockerfile based on the features encapsulated within this image. The container was built using the following image:



```
FROM php:7.4-apache
```

Figure 5.1: Dockerfile

Then we created a Docker-compose file that contains all the configuration like the container name, container port number, the dockerfile that we used and the directory that has all the web pages.



docker-compose.yml

```
version: '3'
services:
  php:
    build:
      context: ./
      dockerfile: Dockerfile
    container_name: flowershop
    ports:
      - "8081:80"
    volumes:
      - ./flowershop-pages/:/var/www/html/

```

Figure 5.2: Docker-compose File

At this point, we specified that the image was built from the Dockerfile that existed in the same directory as Docker compose file, an instance of a container named "flowershop" was built to operate on port 8081 on the host machine, establishing a connection to port 80 within the container.

And then copying all the HTML pages that were designed for this training existed in flowershop-pages directory (i.e. existed in the same directory as Dockerfile and Docker-compose file) into the path “/var/www/html/” which existed inside the container.

We provided the students with the link to the whole directory that contained all the files and configurations needed for this training that was uploaded to the Drive in the chat utility.

5.3 Training Delivery

The training was done in CYBR 3108 Defensive Programming Class. It was designed to be conducted in two 75-minute virtual sessions but due to technical issues, it took 3 sessions. The training enabled hands-on experience with Docker Technology, covering container creation, running webservers, understanding web vulnerabilities, and coming up with solutions for the vulnerabilities found. Two surveys were designed to collect the background and the perceptions of the students. The data was Collected through the two surveys to assess the effectiveness of the small scenario using Docker technology and the whole training then, analyzed to document results providing insights into student learning outcomes.

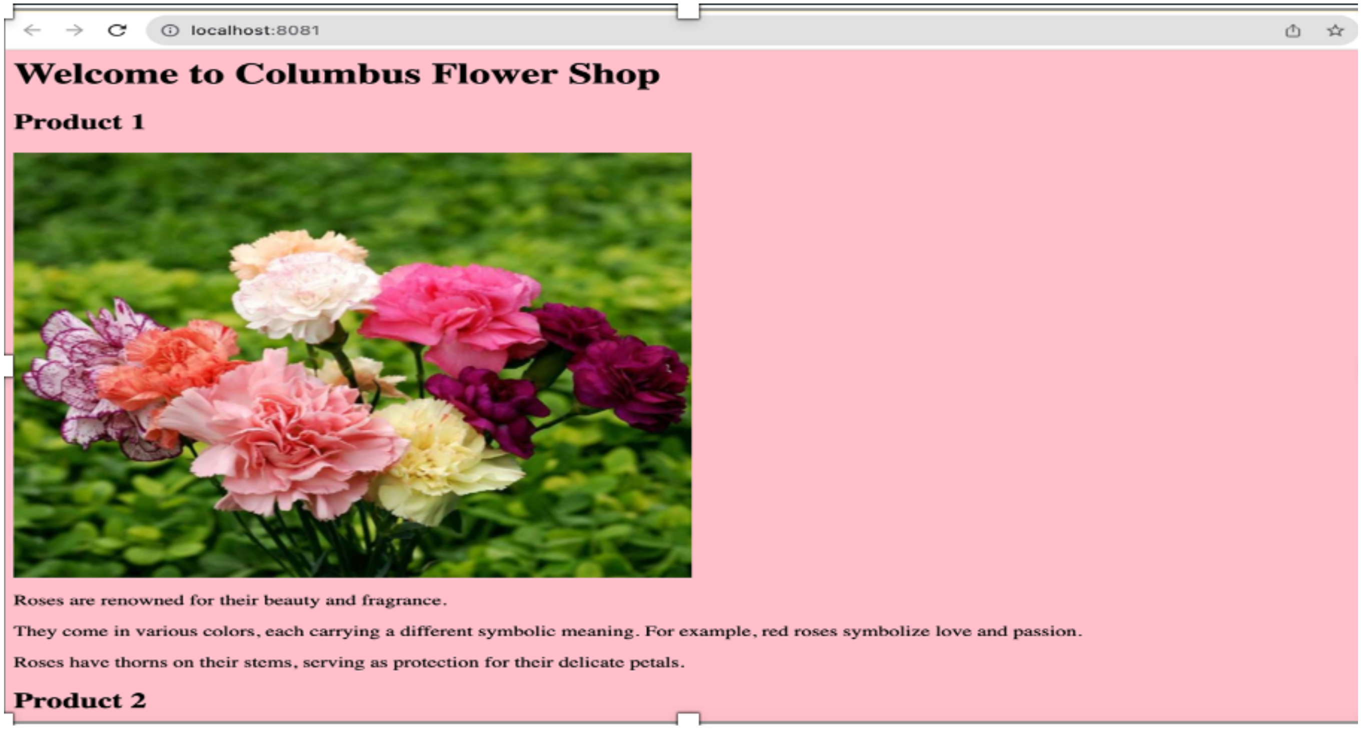
5.4 Training Walkthrough

The walkthrough took place in three phases:

5.3.1 Acquainting Students with Docker Technology

This phase encompassed an introduction to Docker Technology’s fundamental concepts, components, and significance. Students acquired practical skills in pulling images to create containers on their computers, learning to allocate port numbers for container access through either Docker Desktop or a web browser. The instructional module extended to exploring web servers, web applications, and the pivotal distinctions between Docker Compose and Dockerfile, both of which hold crucial roles in Docker operations.

Following this foundational learning, a more extensive scenario was introduced, featuring a website with an aesthetically designed interface showcasing various flowers. This scenario incorporated an interactive element—a comment section prompting students to share their favorite flower from the displayed content. This practical application served to reinforce the theoretical concepts covered in the Docker technology introduction.



Welcome to Columbus Flower Shop

Product 1

Roses are renowned for their beauty and fragrance.

They come in various colors, each carrying a different symbolic meaning. For example, red roses symbolize love and passion.

Roses have thorns on their stems, serving as protection for their delicate petals.

Product 2

Figure 5.3: Screenshot from the Columbus Flower Shop Website

5.3.2 Exploring the Website and its Vulnerabilities

The second part was giving them time to explore the website and to try and attack it. It was a chance for them to explore and discover the vulnerabilities of the website and their ideas to solve these vulnerabilities. This was the way to cover the HTML injection vulnerability. Some students discovered that the website could execute HTML tags inside the comment field on the website and the website stored it, in addition, students identified various vulnerabilities, including issues related to file permissions, where all files were found to have 777 permissions which means full access (read, write, and execute). Furthermore, they observed a security concern regarding the website's connection, noting the use of an unsecured HTTP connection instead of the more secure HTTPS protocol. They raised a flag that the website had vulnerabilities and started to give ideas about how to solve it.

5.3.3 HTML Injection

The third part focused on elucidating the concept of HTML injection, guiding students through the HTML code, and instilling preventive measures using a straightforward sanitize function to cleanse user input of HTML tags. Notably, students navigated to the directory housing all HTML files and accessed the "handling.js" file. This specific file, tasked with managing the page, especially the comment section, played a pivotal role in implementing measures to safeguard against HTML injection.

We taught the students to create a sanitize function that uses a temporary div element to safely handle the input comment as HTML, and then retrieves only the text content of this div, effectively removing any HTML tags and ensuring the sanitized comment is free from potentially malicious code.

We also presented students with brief information about other ways to prevent code injection attacks such as Input Validation and Output Encoding.

CHAPTER 6: RESULTS AND ANALYSIS

This chapter is dedicated to presenting what the two surveys on Familiarity Assessment and Training Exercise entailed as well as what the results indicated.

6.1 Familiarity Assessment Survey Results

The first survey, titled "Familiarity Assessment," involved a cohort of 14 students, and gauged participants' familiarity with various aspects of technology. The following table presents the responses regarding students' familiarity with Docker Technology, webservers and web Applications, and web security. The options range from "Quite familiar with plenty of hands-on experience" to "Never heard about it," with an additional category for "Other."

Table 6.1 tabulates the results from the Familiarity Assessment Survey

Table 6.1: Familiarity Assessment Survey Results

Familiarity Assessment Survey Results		**Quite familiar with plenty of hands-on experience**	**Familiar with some hands-on experience**	**Quite familiar with no hands-on experience**	**Never heard about it**	**Other**
How familiar are you with	**Docker Technology**	-	1 student	5 students	6 students	2 students
	Webservers and Web Applications	-	11 students	3 students	-	-
	Web Security	1 student	8 students	5 students	-	-

Notably, the results indicate the distribution of students across different familiarity levels for each technology category based on their hands-on experience.

Responses to the experience with Docker indicated a spectrum of experiences, with some having no prior exposure, a few expressing limited familiarity from verbal introductions, and others having intermediate to advanced experience, including practical applications in projects and internships. The responses reflected a spectrum of knowledge, ranging from those who had only heard of Docker to others who acknowledged its utility in maintaining software and appreciated its potential in web programming and cybersecurity courses. The diversity in experience underscored the importance of accommodating different levels of familiarity in the training exercise, ensuring it catered to both beginners and those with more advanced knowledge of Docker technology.

Furthermore, students' experiences with web servers and applications varied. Some had limited exposure, primarily through class assignments, while others had more hands-on experience, including creating and hosting websites, building full-stack applications with Django, and working on various web server technologies like Tomcat, XAMPP, and LAMP stacks. A few had practical experience in cybersecurity test scenarios, and some highlighted theoretical knowledge gained from coursework. The responses reflected a diverse range of experiences, emphasizing the importance of considering varied skill levels and backgrounds in the training exercise.

Some students took courses in Web Design-Front End and application programming, another group undertook related courses, and a third group applied web server functions in cybersecurity test scenarios.

Students expressed a range of experiences with web security. Some indicated no hands-on experience, while others detailed their participation in specific activities such as resolving a simulated web-defacement attack or engaging in a hackathon with XSS scripting attacks. Several mentioned exposures to web security concepts in coursework, with varying degrees of hands-on practice, such as labs on website infiltration, cyber range exercises, and participation in training exercises and drills. The responses highlighted a diverse background in web security knowledge among the participants, showcasing the need for adaptable training approaches. Diverse educational backgrounds and practical applications were evident, with variations in web security experiences, ranging from multiple semesters of network security courses to hands-on experience in cyber range scenarios.

6.2 Training Exercise Survey Results

Out of the 14 students who completed the familiarity survey, 9 students participated in the training and completed survey 2 that was shared after the training to elicit their feedback on the training exercise.

In analyzing students' educational backgrounds, 6 students were enrolled in the Bachelor of Science in Computer Science-Cybersecurity program, while 3 students selected "Other," showcasing diversity in academic affiliations. The survey encompassed inquiries regarding Docker technology, web servers, applications, HTML code injection, and the holistic student experience. Noteworthy positive feedback highlighted comfort with Docker after the exercise and appreciation for the HTML code injection portion.

Table 6.2 tabulates the results from the Training Exercise Survey

Table 6.2: Training Exercise Survey Rating

Training Exercise (Survey 2 Questions)		Number of Students	5 Stars (Very Helpful)	4 Stars	3 Stars	2 Stars	1 Star (Not Helpful)
Please rate how helpful the exercise was in explaining the fundamentals of …	Docker technology	9	6	3	-	-	-
	Webservers and Web applications	9	7	1	-	1	-
	html code injection	9	8	1	-	-	-

Note: "-" indicates that no students provided a rating in that category.

The following table shows different responses from the students who were asked about their feedback to Docker technology, web servers, web application and HTML code injection. They were asked about what worked well, what did not work well and how the exercise could have been improved.

Table 6.3 includes the comments from the participant's feedback on the training exercise, encompassing Docker Technology, Web Servers, Web Applications, and HTML Code Injection, explained in the training.

Table 6.3: Training Exercise Survey– Written feedback from participants

Training Exercise (Survey 2 Questions)	**Number of Students**	**Responses**
Docker Technology	**9**	• It all worked well I just had to remember after to clear the cache for the site. • I think it was very informative and interesting to learn how to do something that was like virtual box that we run in almost every class but a little different. • Even though my docker had issues, It still worked mostly ok • Having the docker container ready to go saved a lot of time (once open) and is an interesting way to get a first introduction to the technology. Unfortunately the time it takes to get it installed and running defeated the convenience of the container. If there is some way to pre-register and have the container saved to our accounts before class that would be great. • I enjoyed learning about the Docker Technology, although I feel like we only touched the surface of what can be done with the application(Hands-on wise). I feel like the majority of us interacting with the application was to do the examples. I would have loved to be able to explore what else could be done with the application. • I think the setup and very simple and I liked how the vulnerabilities were very simple to identify but also very important to be able to catch. For improvement, I think that it would be better if everything was better set up before class (such as powerpoints, meet links, etc.) if possible. Just to usage time as best as possible. Also, I liked the use of docker-compose instead of just a dockerfile. That way we got to see both and how they both worked.

Training Exercise (Survey 2 Questions)	Number of Students	Responses
		• I liked that we implemented a sanitization function to sanitize user input. I think that Microsoft Teams was weak when it came to the video call at times so some things were lost in the class. Maybe next time use another video interface. I think the exercise was well structured as well. • I feel very comfortable in using the Docker technology after this demonstration. This exercise was very clear and concise in its instruction and therefore allowed for me to see a number of good use cases for the Docker technology. • I think it was a pretty good way to try out Docker. The explanation of the types of files used like dockercompose vs dockerfile were good to know. I think it would be improved by looking a little bit more at configuring Docker files or slightly more complex servers, since we're already learning by doing

Training Exercise (Survey 2 Questions)	Number of Students	Responses
Webservers and Web applications	**9**	• Great no input. Everything was explained very well. • The webserver and application portion of the exercise was very fun. I enjoyed trying to find the bug with the webpage. • That worked perfectly. • I think it was great to see the webserver behind-the-scenes files so we can think about how the website is actually functioning. • I really enjoyed that she went over the code that builds the website. I am not in the web-based track so it was interesting to see and to be able to interact with. I do not think there is anything to improve on in this portion. • Setting up a webserver with docker was very informative and good to know. I think that portion was also very well setup and I like the simplicity of it as well. • I liked that we learned how to use them and what they are. I think it would be good to go over code that implements the webserver and applications. • Overall, I think the webserver portion was fairly well defined and helpful in my personal understanding of the concepts. I just personally was unable to grasp the concepts as quickly as the other portions and, as a result, felt myself needing to clarify the content with examples. • For me it felt like basic concepts like what webservers are were overly explained, while details on how to work with them were missed.

Training Exercise (Survey 2 Questions)	**Number of Students**	**Responses**
html code injection	9	• It showed us a lot of how it can be fixed but It might need a little bit more information on what each thing does for sanitization. • I enjoyed this portion of the exercise a lot. It was very informative learning different intrusion methods. • Maybe take some more time to explain the code we added • I think it was a great demonstration. Having some available scripts that could be copied and pasted in for students to play with would be fun. • It may have had a typo in the code when I updated it but after we put in the sanitization code, I was still able to insert a link into the comment. I did enjoy the hands-on aspect, even though I am not familiar with HTML code, I still felt included. • I did not know the name for this attack although I knew what it was. It was very informative to put a name to the attack and to see it live and be able to experiment with it. • I think that the exercise went well since we were able to play around with the website then figure out the different vulnerabilities. It would be better if we could all use the same website/server at the same time. • I think the HTML code injection was the most effective portion of the exercise. It very clearly defined the vulnerability cause, the threats posed by injection, and a possible solution through the flowershop website example scenario. It worked very well. • I think that it was a good way to show HTML code injection. Asking students to experiment with injecting the HTML themselves is a good way to demonstrate.

Different responses from the students were provides for Docker technology, Web servers and applications, HTML code injection

Docker Technology:

Participants provided valuable feedback on the Docker technology section of the training exercise. Overall, the functionality of Docker was praised for its seamless operation, with minor considerations like cache clearing. The exercise was deemed informative and interesting, with participants drawing parallels to virtual box experiences. Despite encountering Docker challenges, the overall functionality and time-saving aspects were highlighted. Some participants expressed a desire for deeper exploration of Docker's capabilities, suggesting pre-registration for further efficiency. The setup simplicity and straightforward vulnerability identification received positive feedback, along with the appreciation for using Docker-Compose alongside Dockerfile. The implementation of a sanitization function and suggestions for alternatives were noted, contributing to participants' overall comfort, and understanding of Docker technology.

Web Servers and Application:

Participants provided positive feedback on the webserver and application section of the training exercise. The exercise was described as enjoyable and effective, with participants expressing satisfaction in identifying vulnerabilities and gaining insights into the functioning of web servers. Specific appreciation was given for the instructor's detailed explanation of the code building the website, offering valuable perspectives even for those not in the web-based track. Setting up a web server with Docker was highlighted as informative and well-structured, emphasizing the simplicity of the process. While some participants found certain concepts, such as basic web server functionalities, to be overly explained, others felt the need for more detailed guidance on working with web servers. Overall, participants found this section to be helpful and defined, with some suggesting additional clarification through examples.

HTML Code Injection:

Participants provided insightful feedback on the HTML code injection section of the exercise. While expressing appreciation for the demonstration of fixing vulnerabilities, some participants suggested the need for additional information on the specifics of each sanitization method. The hands-on aspect was enjoyed, with requests for more time dedicated to explaining the added code and the inclusion of available scripts for experimentation. Some participants noted potential typos in the code but still felt included and engaged in the exercise, despite not being familiar with HTML code. The session's effectiveness was highlighted, particularly in clarifying the causes and threats of HTML code injection, with the flower shop scenario serving as an illustrative example. Overall, participants found the HTML code injection portion to be informative, engaging, and effective in demonstrating the vulnerabilities and solutions associated with this type of cyber-attack.

Table 6.4 presents the comments from students about what they liked most about the overall training and what they didn't like.

Table 6.4: Training Exercise Survey Results – Likes and Dislikes

Training Exercise (Survey 2 Questions)	**Number of Students**	**Responses**
What did you like most about the whole exercise? Why?	9	• It was informative and instructors knew what they were talking about a lot. • I liked learning about docker the most and learning how its implemented • Using a new platform(Docker) • I liked being able to change the webserver files and see the direct impact on the site. • I enjoyed the hands-on aspect of the exercise. I feel like Iearn better this way as opposed to listening to lecture. • It was very informative overall. The fact that it was kept very simple is also a huge plus, it wasn't a lot of informative to take in at once but also not so little. I already had some experience with docker and webservers but still I benefited from this and learned a lot. • I loved the html code injection portion the most since we were able to see how it was vulnerable, the different ways to secure or to prevent the vulnerability, then actually implementing the security. I loved that we implemented the user sanitization function. • I enjoyed the "flowershop" demonstrations. The scenario allowed for me to see the whole process of starting a webserver through Docker, configuring code for the page to be more secure, and of course, provided a solid demonstration of HTML injection. • I really like the overall concept. Using Docker to let students experiment with code injection is useful, since we can see what happens in the server without actually damaging anything. I found it helpful when Zeinab responded to questions in the text chat since I could read the response. It's worth continuing to explore Docker for cybersecurity education.

Training Exercise (Survey 2 Questions)	Number of Students	Responses
What did you not like about the exercise? Why?	9	• There were too many technical issues to start off with • There wasn't anything I didn't like about the exercise. • The issues docker had • N/a, it was good. • The only thing I did not like about the exercise is the communication issues. This was no one fault, but at times it made the presentation difficult to follow because the audio would go in and out or we could not follow along with the slides as easily. • As mentioned, I think the initial setting up could of been faster, other than that it was great. • There was nothing I did not like about the exercise. • I think the webserver and application portion could use a bit clearer defining, though this could be a personal shortcoming on my end. I just struggled to grasp the concepts/application as quickly as the other portions of the exercise. • The presentation would work well as a handout but had too much information on each slide for a live talk. Your delivery relied too much on reading out the slides, which hurts any speaker's performance. I can't remember much about any coverage of web servers because it was in the middle of a bunch of other topics, for example. The exercise was badly held back by Microsoft teams: between the time it took to get started every session and the poor audio quality, I was really struggling to parse anything by the end of today's class because one of the presenter computers was echoing. I think it's good to ask for questions, but found myself avoiding giving any because either I simply didn't have anything to ask or didn't want to waste any more time than Teams already had.

What students like is that they praised the informative and well-instructed Docker technology session, expressing a strong interest in learning its implementation. The introduction of a new platform, Docker, was appreciated for its novelty. The ability to modify web server files and witness direct site impacts resonated positively, aligning with the enjoyment of a hands-on learning approach. The exercise's simplicity was commended for its balance, catering to various experience levels. The HTML code injection segment garnered enthusiasm, particularly for its vulnerability showcase, diverse security methods, and the implementation of a user sanitization function. The "flowershop" scenario was highlighted for providing a comprehensive view of the webserver setup process through Docker, enhancing page security. Participants valued the overall concept of using Docker for hands-on code injection experiments in cybersecurity education, emphasizing its utility in understanding server dynamics without causing actual damage. The ongoing exploration of Docker in this educational context was encouraged for its effectiveness in enhancing learning experiences.

In conclusion, this integrated analysis provided a holistic view of students' experiences, perceptions, and different background levels. The findings offered insights for refining pedagogical approaches, enhancing educational content, and optimizing future training exercises. On the other hand, some participants encountered technical issues at the start as the training was held from Egypt using Microsoft Teams, overall, there were positive sentiments toward the exercise, with individuals expressing satisfaction and labeling it as good. Specific concerns were raised about Docker-related problems, and participants cited communication issues as a slight drawback, attributing it to audio inconsistencies and occasional challenges following slides. Suggestions for improvement included faster initial setup, clearer definitions in the webserver and application section, and a more streamlined presentation format. Critiques were voiced regarding the abundance of information on slides during live talks, impacting the speaker's performance. Additionally, challenges with Microsoft Teams were highlighted, affecting session initiation times, and causing audio quality problems. Despite these challenges, participants acknowledged the exercise's potential as a handout and recommended addressing technical aspects for future sessions.

CHAPTER 7: CONCLUSION

This study has yielded valuable insights into the pivotal role of Docker technology in enhancing scenario-based learning within the cybersecurity domain. The results showed the efficiency of employing small, focused scenarios to augment students' hands-on experience and theoretical knowledge. The utilization of pre-exercise surveys facilitated a nuanced understanding of the exercise's impact, highlighting the need for tailored educational programs that harmonize theoretical understanding with practical application in Docker technology, web servers and applications and HTML injection. Its findings are in agreement with the concluding results of the papers examined in the literature review that emphasized the integration of practical exercises, particularly in cybersecurity contexts.

While time constraints prevented the implementation of additional scenarios, the potential for applying SSL certificates for secure connections and establishing a Dockerized honeypot to attract and analyze malicious activity or building a Docker-based penetration testing lab remain evident. This study serves as a foundational exploration in the realm of cybersecurity, presenting a compelling starting point for broader applications across various domains with larger-scale scenarios and bigger students’ pools, a prospect that warrants future investigation.’

CHAPTER 8: REFERENCES

1. P. Baybutt, P. Inc, Primatech, "A Scenario-Based Approach for Industrial Cyber Security Vulnerability Analysis," 83, 2004.

2. D. Kerven, K. Nagel, S. Smith, S. Abraham, and L. Young, "Scenario-Based Inquiry for Engagement in General Education Computing," pp. 303–308, 2017.

3. T. Ghosh and G. Francia III, "Assessing Competencies Using Scenario-Based Learning in Cybersecurity," Journal of Cybersecurity and Privacy, vol. 1, no. 4, pp. 539–552, 2021.

4. N. Chowdhury, S. Katsikas, and V. Gkioulos, "Modeling effective cybersecurity training frameworks: A Delphi method-based study," Computers & Security, vol. 113, p. 102551, 2022.

5. S. Yeom, D. Shin, and D. Shin, "Scenario-based cyber attack·defense education system on virtual machines integrated by web technologies for protection of multimedia contents in a network," Multimedia Tools and Applications, vol. 80, pp. 34085–34101, 2021.

APPENDICES

Appendix 1: Familiarity Assessment Survey

Survey 1. Before the exercise

1. How familiar are you with Docker technology?

Quite familiar with plenty of hands-on experience

Familiar with some hands-on experience

Quite familiar with no hands-on experience

Familiar with no hands-on experience

Never heard about it

2. How familiar are you with networking?

Quite familiar with plenty of hands-on experience

Familiar with some hands-on experience

Quite familiar with no hands-on experience

Familiar with no hands-on experience

Never heard about it

3. How familiar are you with how web servers and applications work?

Quite familiar with plenty of hands-on experience

Familiar with some hands-on experience

Quite familiar with no hands-on experience

Familiar with no hands-on experience

Never heard about it

4. How familiar are you with code injection?

Quite familiar with plenty of hands-on experience

Familiar with some hands-on experience

Quite familiar with no hands-on experience

Familiar with no hands-on experience

Never heard about it

Appendix 2: Training Exercise Survey

Survey 2. After the exercise

1. **Please rate how helpful the exercise was in explaining the fundamentals of Docker technology.**

1-Not helpful to 5-Very helpful

2. **Please give us feedback on the Docker technology portion of the exercise.**
3. **Please rate how helpful the exercise was in explaining fundamentals of webservers and applications.**

1-Not helpful to 5-Very helpful

4. **Please give us feedback on the webserver and applications portion of the exercise.**
5. **Please rate helpful the exercise was in explaining fundamentals of html code injection.**

1-Not helpful to 5-Very helpful

6. **Please give us feedback on the html code injection portion of the exercise.**
7. **What did you like most about the exercise? Why?**
8. **What did you not like about the exercise? Why?**
9. **Which degree program are you in?**

BS Computer Science – Cybersecurity track

BS Cybersecurity

Nexus in Cybersecurity of Fintech

BBA in Cybersecurity

MS in Applied Computer Science

MS in Cybersecurity

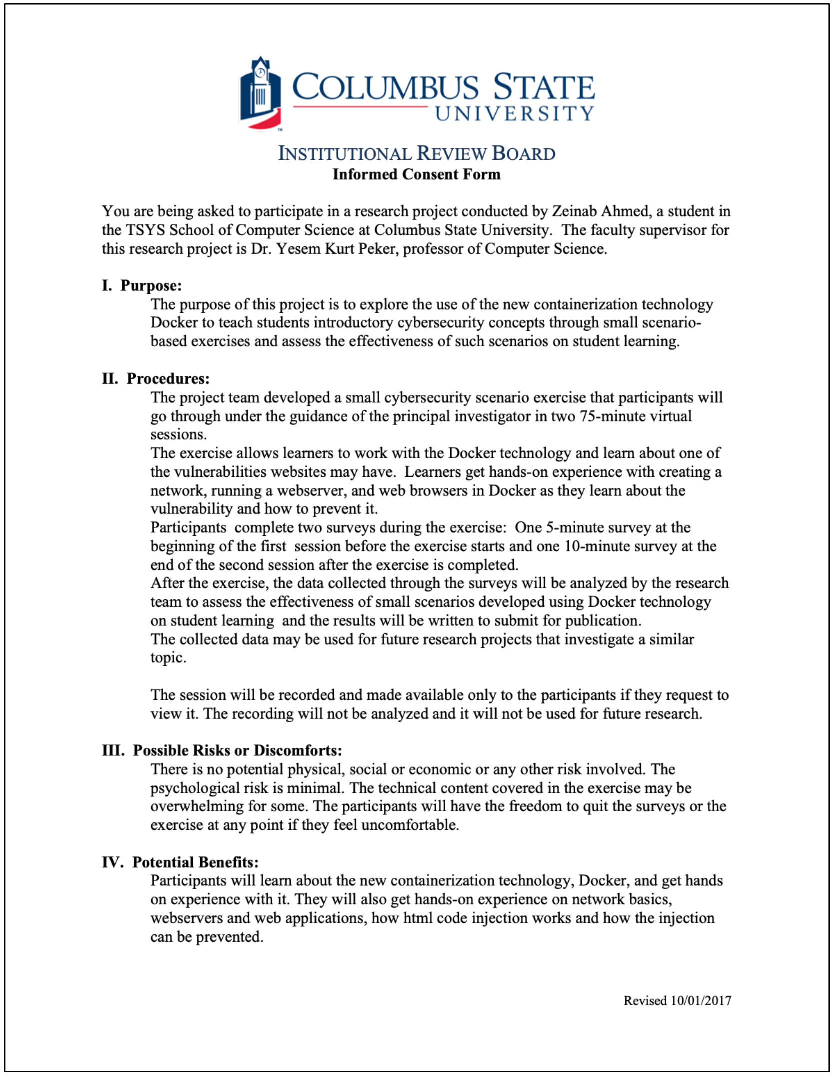
Other: Please specify:

10. Gender

11. Ethnicity

12. Which cybersecurity courses have you completed so far in the program?

Appendix 4: Informed Consent Form



COLUMBUS STATE
UNIVERSITY

INSTITUTIONAL REVIEW BOARD
Informed Consent Form

You are being asked to participate in a research project conducted by Zeinab Ahmed, a student in the TSYS School of Computer Science at Columbus State University. The faculty supervisor for this research project is Dr. Yesem Kurt Peker, professor of Computer Science.

I. Purpose:

The purpose of this project is to explore the use of the new containerization technology Docker to teach students introductory cybersecurity concepts through small scenario-based exercises and assess the effectiveness of such scenarios on student learning.

II. Procedures:

The project team developed a small cybersecurity scenario exercise that participants will go through under the guidance of the principal investigator in two 75-minute virtual sessions.
The exercise allows learners to work with the Docker technology and learn about one of the vulnerabilities websites may have. Learners get hands-on experience with creating a network, running a webserver, and web browsers in Docker as they learn about the vulnerability and how to prevent it.
Participants complete two surveys during the exercise: One 5-minute survey at the beginning of the first session before the exercise starts and one 10-minute survey at the end of the second session after the exercise is completed.
After the exercise, the data collected through the surveys will be analyzed by the research team to assess the effectiveness of small scenarios developed using Docker technology on student learning and the results will be written to submit for publication.
The collected data may be used for future research projects that investigate a similar topic.

The session will be recorded and made available only to the participants if they request to view it. The recording will not be analyzed and it will not be used for future research.

III. Possible Risks or Discomforts:

There is no potential physical, social or economic or any other risk involved. The psychological risk is minimal. The technical content covered in the exercise may be overwhelming for some. The participants will have the freedom to quit the surveys or the exercise at any point if they feel uncomfortable.

IV. Potential Benefits:

Participants will learn about the new containerization technology, Docker, and get hands on experience with it. They will also get hands-on experience on network basics, webservers and web applications, how html code injection works and how the injection can be prevented.

Revised 10/01/2017

Appendix 5: Training PowerPoint Presentation

Dive into Docker: Unleash the container magic

What is docker

DOCKER IS AN OPEN-SOURCE PLATFORM FOR DEVELOPING, SHIPPING, AND RUNNING APPLICATIONS.

DOCKER ENABLES YOU TO SEPARATE YOUR APPLICATIONS FROM YOUR INFRASTRUCTURE SO YOU CAN DELIVER SOFTWARE QUICKLY.

DOCKER ENABLES YOU TO MANAGE YOUR INFRASTRUCTURE IN THE SAME WAY YOU MANAGE YOUR APPLICATIONS.

DOCKER USES OS-LEVEL VIRTUALIZATION TO DELIVER SOFTWARE IN PACKAGES CALLED CONTAINERS.

Containers vs Virtual machines

Aspect	**Containers**	**Virtual Machines (VMs)**
Isolation and Overhead	Share the host OS kernel, lightweight	Run separate OS instances, heavier
Resource Utilization	Highly efficient, minimal overhead	Less efficient, higher resource use
Portability	Extremely portable, consistent	Portable but larger, may need config
Startup Time	Almost instant	Longer startup time
Security and Isolation	Good isolation, shared kernel	Stronger isolation, separate kernels
Maintenance & Scaling	Easier management and scaling	More complex management and scaling
Use Cases	Microservices, CI/CD, cloud-native apps	Strong isolation, legacy systems

Docker Importance

Portability

- Containers package apps and dependencies, ensuring consistent operation across environments.

Efficiency & Resources

- Lightweight containers minimize resource overhead, enabling cost-effective, scalable solutions.

Isolation

- Application-level isolation prevents issues in one container from affecting others, enhancing stability.

Rapid Deployment

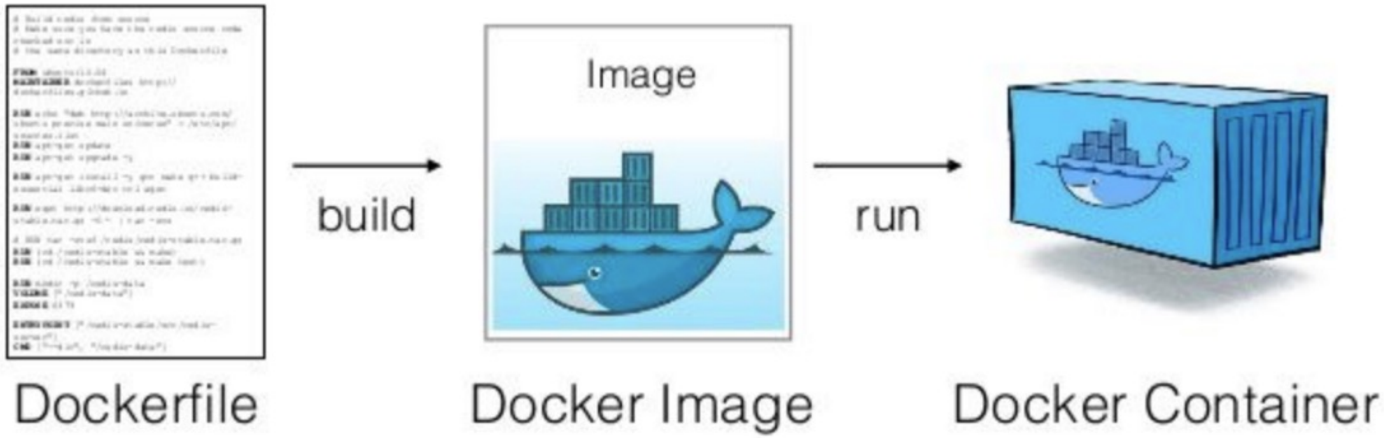
- Quick container creation and startup accelerates development, testing, and deployment cycles.

Scalability

- Docker facilitates horizontal scaling for handling varying workloads and ensuring high availability.

Docker Components

1. Docker Image:
    1. A Docker image is a pre-packaged, standalone, and immutable package that includes all the necessary files and configurations to run a specific application or service. It's created from a Dockerfile and serves as a template for containers.
2. Docker Container:
    1. A Docker container is a running instance of a Docker image. It's a lightweight, isolated environment where your application can execute. Containers are portable and can run consistently on various systems, making them ideal for deploying and scaling applications.
3. Dockerfile:
    1. A Dockerfile is like a recipe or set of instructions that defines how a Docker image should be built. It specifies what operating system, software, configurations, and files should be included in the image. Think of it as a blueprint for creating containers.



Getting started-
Running a Hello-World Container

- **Run a "Hello, World!" Container**:
  - open your terminal, run the following command to pull and run a "Hello, World!" container from the Docker Hub:
    - docker run hello-world
- Observe Output:
  - Docker will download the "hello-world" image from the Docker Hub if it's not already cached on your system. Then, it will create a container from that image and execute it.
  - You'll see a message indicating that your installation appears to be working correctly. It will explain the steps Docker took and provide some information about how containers work.

Getting started-
Running a Nginx Container

- **Pull and Run an Nginx Container**:
  - Open your terminal and run the following command to pull and run an Nginx web server container:
    - `docker run -d -p 8080:80 nginx`
      - -d: This option runs the container in detached mode, meaning it runs in the background.
      - -p 8080:80: This option maps port 8080 on your host to port 80 inside the container.
      - nginx: This is the name of the Nginx container image.

- Access the Nginx Web Server:
  - Open a web browser and navigate to http://localhost:8080 . You should see the default Nginx welcome page.

- A web application is a software program that you can use through your web browser, like Chrome or Firefox, by accessing it over the internet. It's like a tool or service that runs on a remote computer (a server) and allows you to perform various tasks or access information. You don't need to install anything on your device because you interact with it online.

  Examples of web applications include email services like Gmail, social media platforms like Facebook, online shopping websites like Amazon, and online banking systems. When you use a web app, your web browser communicates with the web server hosting the application to send and receive data, enabling you to do things like send emails, share photos, shop for products, or manage your finances – all within your browser.

Web application

Web servers

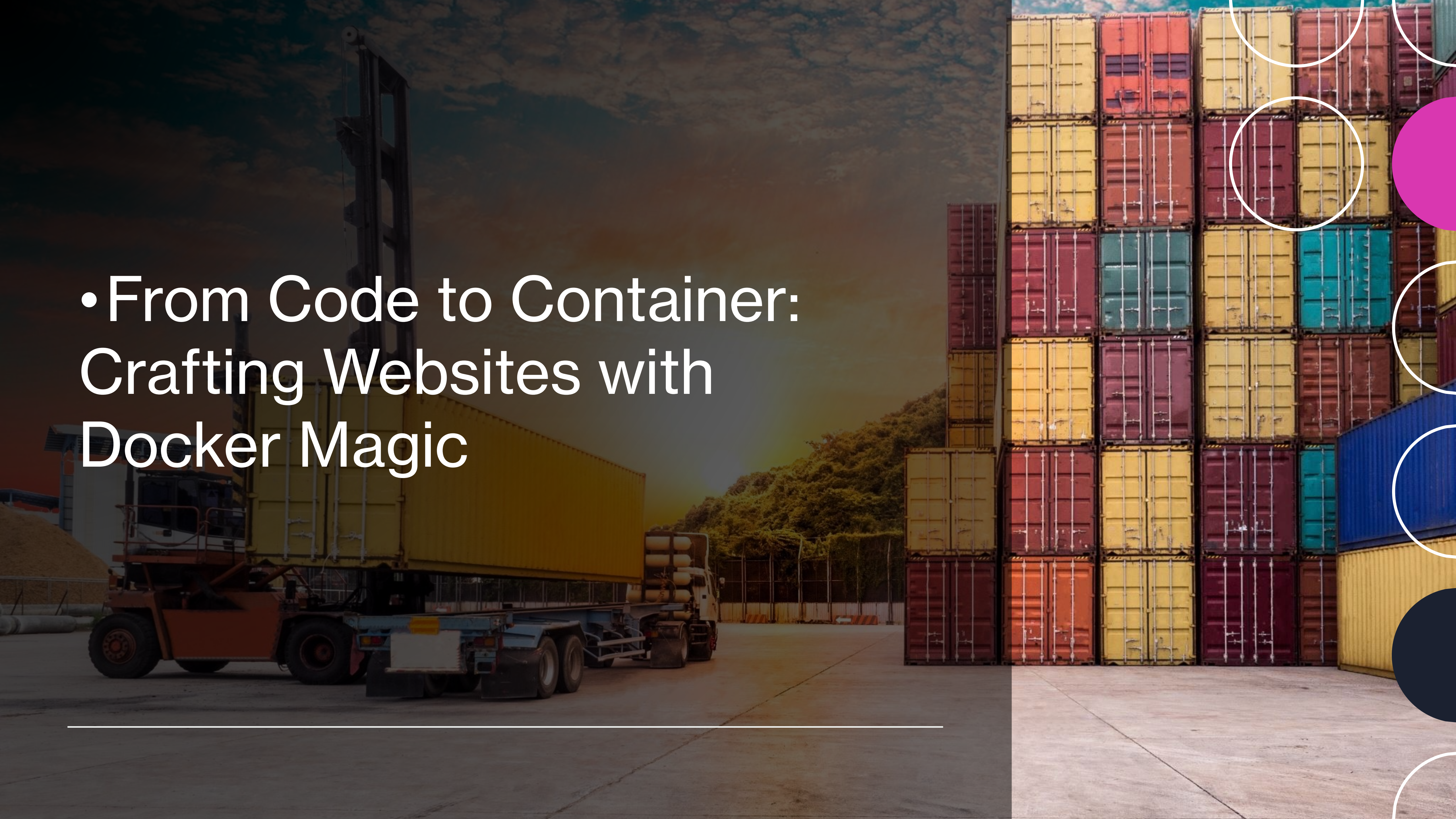
- A web server is like a special computer that stores and delivers web pages and other digital content to your web browser when you request them. Think of it as the waiter in a restaurant who brings you the food you order.
- When you type a web address (like www.example.com) into your browser and hit Enter, your browser sends a request to the web server associated with that address. The web server then finds and sends the requested web page or file back to your browser, allowing you to see and interact with it on your device.
- In essence, a web server's job is to serve up web content and make it accessible to you over the internet, enabling you to view websites, download files, or use web applications. It's a crucial component of the internet that makes the web work.

Dockerfile vs Docker-compose

ASPECT	DOCKERFILE	DOCKER COMPOSE
Purpose	Instructions for building a Docker image.	Define and run multi-container Docker applications.
File Type	Text file with a series of build steps.	YAML file with service and configuration definitions.
Function	Creates a Docker image that contains an application and its dependencies.	Orchestrates multiple containers to run together as a single application stack.
Usage	Used for image creation and customization.	Used for defining and managing multi-container applications, including their relationships and configurations.
Configuration	Specifies how to build an image, including base image, environment, software, and file copies.	Describes services, networks, volumes, and their relationships for a complex application.
Automation	Automates the image creation process.	Automates the management of interconnected containers.
Example	Defines an image for a web server: installs software, sets environment variables, and copies files.	Defines a multi-container application stack with web, database, and caching services.
Command	Typically used with the docker build command to create images.	Typically used with the docker-compose up command to start and manage containers.

Any questions

•From Code to Container: Crafting Websites with Docker Magic



Columbus Flower Shop Website

- Now, let’s create something bigger.
- We will create a website that has different types of flowers with their description, were people can browse it, read about these flowers and leave their comment……. Simple, right!!!.
- Download the provided folder and remember it’s location.
- Access the folder using terminal or command prompt.
- Write “docker-compose up” command in the terminal or command prompt.
- In the folder you will find files that was created in php, javascript and cs. Let me give you a brief explanation about what are those, just refresh your memory.

Hypertext Preprocessor (PHP)

1. **Text File**: A .php file is a plain text file, which means it contains human-readable text and can be edited with a simple text editor.
2. **PHP Code**: Inside a .php file, you'll find PHP code. PHP is a scripting language that is executed on the web server, not in the user's web browser. This code can perform various tasks such as generating dynamic web content, processing form data, connecting to databases, and more.
3. **Server-Side**: PHP code is executed on the web server when a user requests a web page. It processes the code and sends the result (usually HTML) to the user's browser. This allows you to create dynamic and interactive web pages.
4. **File Extension**: The ".php" file extension tells the web server that the file contains PHP code and needs to be processed before sending the output to the user's browser.

- In summary, a .php file is a text file containing PHP code that is executed on a web server to generate dynamic web content, making websites more interactive and functional.

JavaScript (JS)

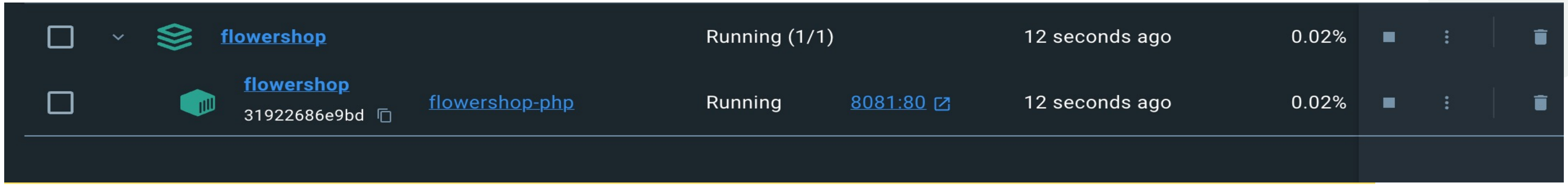
- JavaScript (often abbreviated as "JS") is a versatile and widely used programming language primarily employed for web development. Here's a straightforward explanation:
- JavaScript: JavaScript, often referred to as "JS," is a programming language that web developers use to make websites interactive and dynamic. Unlike HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets), which are used for structuring and styling web content, JavaScript focuses on adding functionality to web pages. It runs directly in a user's web browser, allowing developers to create things like interactive forms, responsive menus, animations, and more. JavaScript is a fundamental tool for creating modern, engaging, and user-friendly websites.

Cascading Style Sheet (CSS)

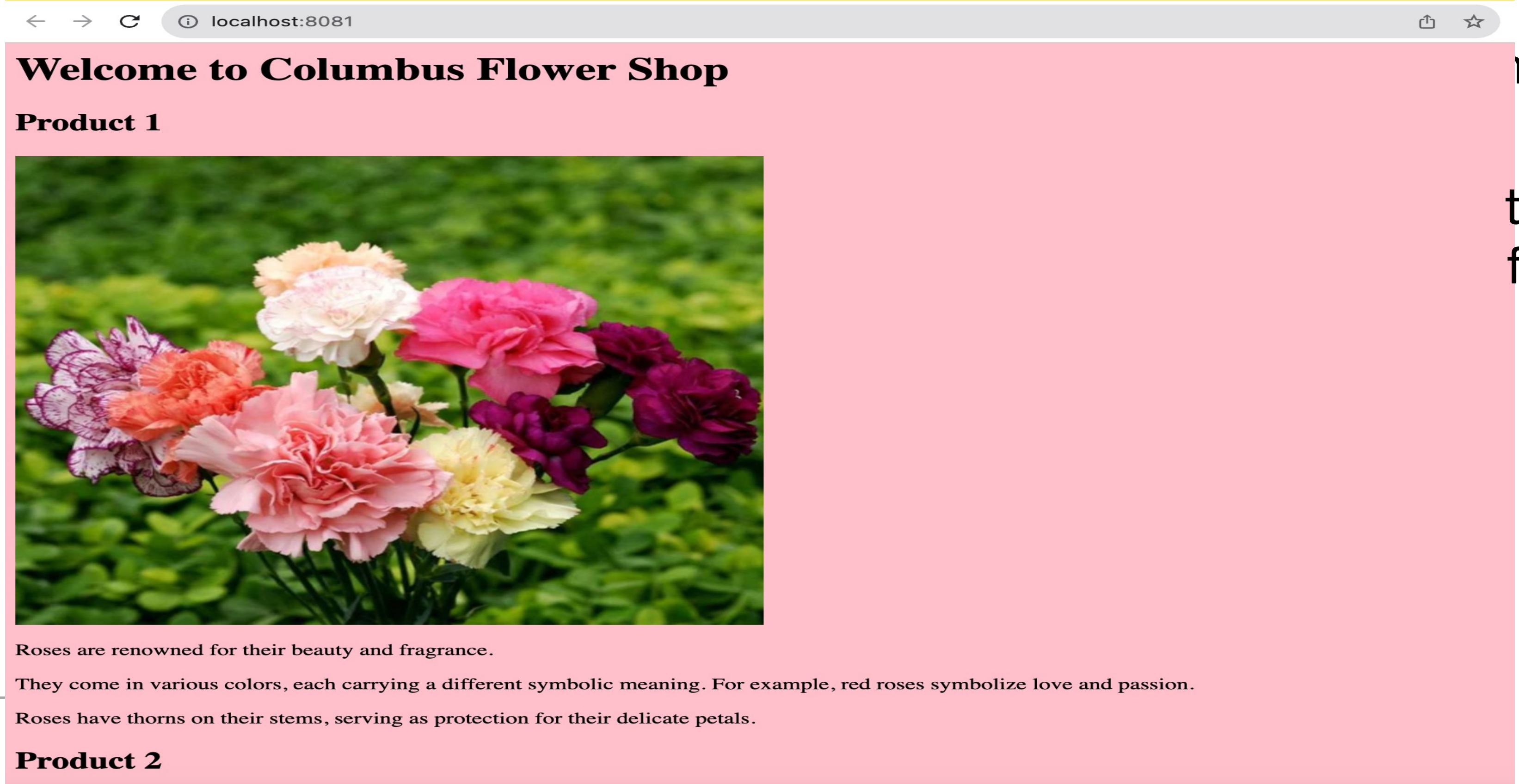
- **CSS (Cascading Style Sheets)** is a language used in web development to control the presentation and styling of web pages. It works alongside HTML (Hypertext Markup Language) to define how web content should look and be displayed on a user's screen. CSS allows web designers and developers to specify properties like fonts, colors, spacing, layout, and more for various elements on a webpage. By separating the content (HTML) from its visual appearance (CSS), it enables the creation of visually appealing and consistent websites across different devices and screen sizes. In essence, CSS is the design and styling tool that makes web pages not only functional but also aesthetically pleasing.

Back to our Flower Shop Website

- If you write docker-compose up command, you will see something like this:



```
□  ⌄  flowershop                                     Running (1/1)          12 seconds ago   0.02%
□     flowershop      flowershop-php                 Running      8081:80   12 seconds ago   0.02%
      31922686e9bd
```



localhost:8081

Welcome to Columbus Flower Shop

Product 1

Roses are renowned for their beauty and fragrance.

They come in various colors, each carrying a different symbolic meaning. For example, red roses symbolize love and passion.

Roses have thorns on their stems, serving as protection for their delicate petals.

Product 2

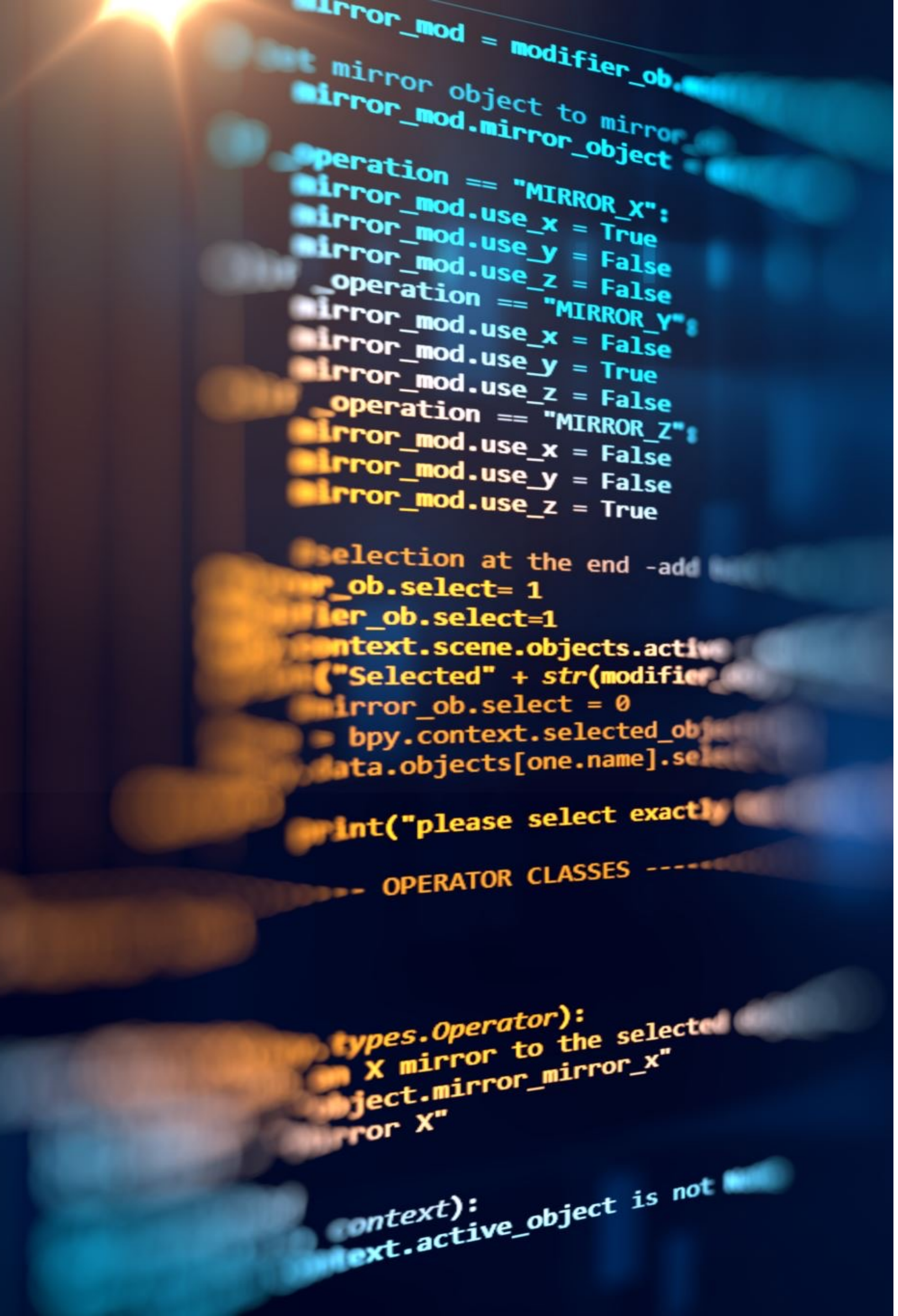
- nber 8081 which
- t number directly fari) and write

Any questions

Investigation job

Scenario

The "Columbus Flower Shop" is an online platform that allows customers to browse and check a wide variety of flowers and floral arrangements. Recently, the website has come under scrutiny due to reports of security issues. The management of Columbus Flower Shop has decided to launch an investigation to identify and address the issues in their website.



HTML Injection

- Imagine a website that lets users post comments. If the website doesn't properly validate or sanitize the input from users, a malicious user can enter HTML or JavaScript code into their comment. When other users view that comment, their web browsers will interpret and execute this malicious code, potentially allowing the attacker to steal user information, hijack user sessions, or perform other harmful actions on the website.
- In essence, HTML injection is like slipping a harmful message or code into a conversation, and when someone reads it, they unintentionally activate the harmful code, allowing the attacker to exploit vulnerabilities in the website or web application. To prevent this, web developers should always validate and sanitize user input to ensure that it doesn't contain malicious code.

Methods to Prevent HTML Injection:

1. **Input Validation:**
    1. Sanitize and validate user inputs.
    2. Use server-side validation to reject unsafe input.
    3. Avoid allowing user-generated content to include HTML tags.
2. **Output Encoding:**
    1. Encode user-generated content before displaying it in web pages.
    2. Use libraries like OWASP's Java Encoder or HTMLPurifier for encoding.
3. **Content Security Policy (CSP):**
    1. Implement CSP headers to restrict sources of content.
    2. Specify which domains are allowed to load scripts, styles, or images.
4. **Use Libraries and Frameworks:**
    1. Leverage modern web frameworks that often include built-in security mechanisms.
    2. Libraries like React, Angular, and Vue provide protection against HTML Injection.
5. **Security Headers:**
    1. Set security-related HTTP response headers.
    2. Implement HTTP Strict Transport Security (HSTS) and X-Content-Type-Options.

handling.js

- Creating santizefunction:

```
function sanitizeInput(comment) {
     const tempElement = document.createElement('div');
     tempElement.innerHTML = comment;
     return tempElement.textContent;
 }
```

handling.js

- add these lines:

```
const commentInput = document.querySelector('#comment').value;
const comment = sanitizeInput(commentInput); //Call the sanitizeInput function
```

- and comment these lines or delete:
  - const commentInput = document.querySelector('#comment');
  - const comment = commentInput.value;

- this is responsible for taking the comment, sanitize it and push it to the list.

Summary

1. HTML Injection can have severe consequences, including data breaches and cross-site scripting attacks.
2. Implement multiple layers of security, including input validation, output encoding, and security headers to protect against HTML Injection.

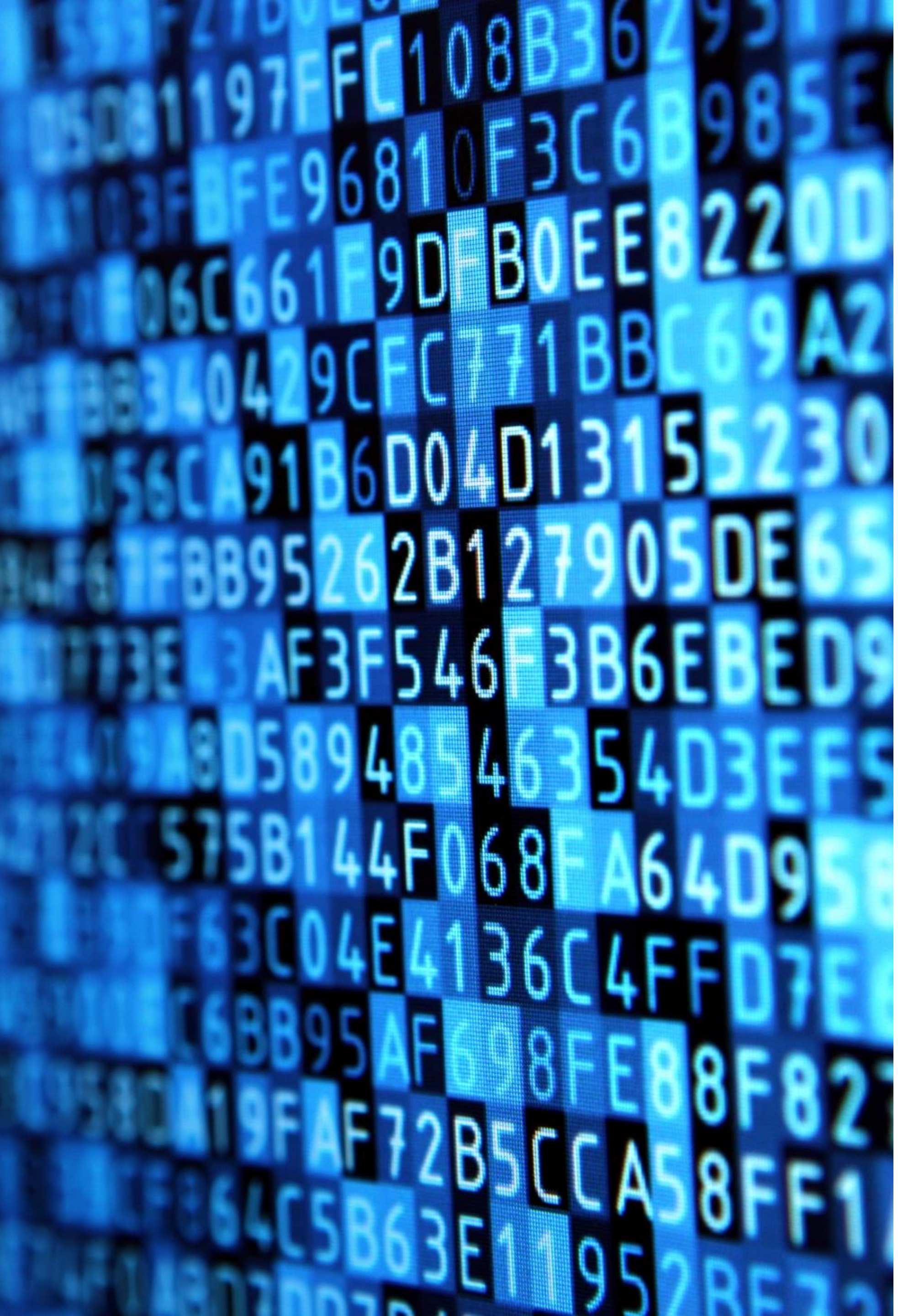
Input Validation

- Validation is the process of checking whether user input meets certain criteria, such as format, length, or range, before accepting it.

Output Encoding

Output encoding is like using a special filter to make sure that what people write on a website won't cause problems. For example, if someone types "<script>alert('Hello!')</script," output encoding changes it to plain text, so it won't run as a script and won't harm the website or users. It keeps things safe and working as they should.

Escaping is like putting a safety shield around certain characters in data. This shield prevents the web browser from thinking that the data is secret code. Instead, it understands the data as regular text or content, making things safer and working as expected. It's a way to protect websites from problems and keep data secure.



Sanitization

- Sanitization is the process of removing or escaping any potentially harmful or unwanted characters or elements from user input or output, such as HTML tags, scripts, or SQL commands.

Content Security Policy (CSP)

- Certainly! Content Security Policy (CSP) is like a set of rules for web pages. It says which places (like trusted websites) can provide content, such as scripts or images, to a webpage. If content doesn't follow these rules, it gets blocked to prevent security problems, like malicious scripts. CSP helps keep websites and users safe from potential attacks.

Security Headers

- **HTTP headers** let the client and the server pass additional information with an HTTP request or response. An HTTP header consists of its case-insensitive name followed by a colon (:), then by its value.
- HTTP security headers are a subset of HTTP headers that is related specifically to security. They are exchanged between a client (usually a web browser) and a server to specify the security details of HTTP communication

Other Vulnerabilities

FILE PERMISSIONS

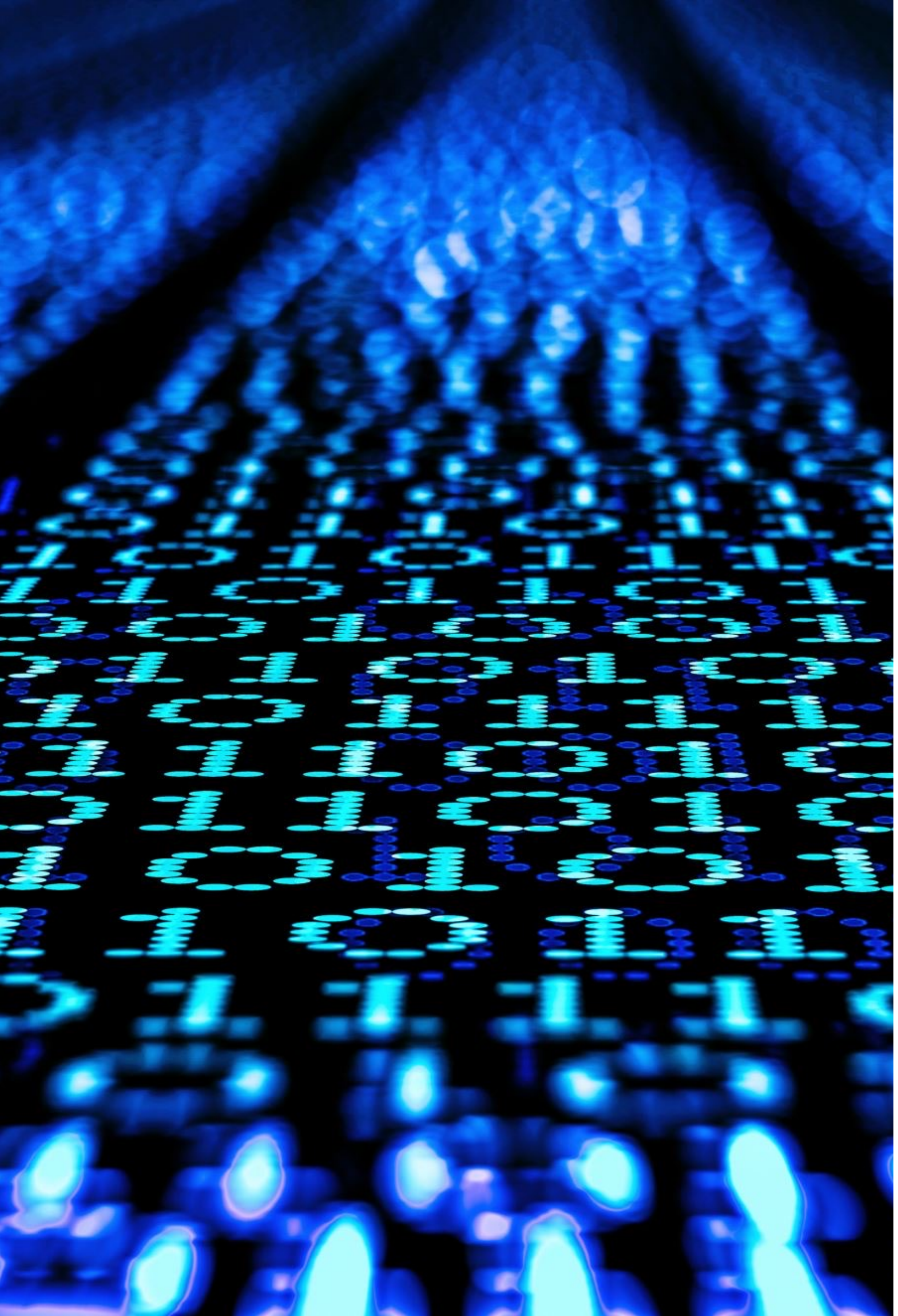
HTTP

File Permission

A vulnerability in file permissions is a security weakness that allows unauthorized users to access, modify, or delete files they shouldn't have access to. It occurs when files or directories are not properly configured with the correct permissions, and it can lead to data breaches, unauthorized changes, or other security risks. For example, if sensitive data files are set to be world-readable, anyone on the system can view their contents, leading to a privacy breach.

To prevent file permission vulnerabilities simply:

- Limit Access: Set file and directory permissions to allow only authorized users or groups to access or modify files. Avoid making them world-readable or world-writable.
- Regular Auditing: Regularly review and audit file permissions to ensure they haven't been misconfigured or inadvertently exposed to unauthorized access.
- Use Strong Passwords: Secure user accounts with strong, unique passwords to prevent unauthorized access to files and directories.
- Implement Access Controls: Use access control lists (ACLs) or role-based access control (RBAC) to finely control who can access and modify files.
- Apply the Principle of Least Privilege: Give users the minimum permissions required for their tasks. Avoid giving overly broad access to files and directories.



HTTP Vulnerabilities and Defenses

- Lack of encryption:
  - Data transferred between a user's browser and a web server is not encrypted. Sensitive information, such as login credentials or personal data, can be intercepted by attackers while in transit, posing a serious security risk
- Lack of authentication:
  - The server is not authenticated in HTTP which allows rogue servers to pose as legitimate servers fooling users to share their information

HTTPS (Hypertext Transfer Protocol Secure)

- Encrypts data during transmission
- Authenticates the server via digital certificates
- Install SSL/TLS Certificates: Acquire and install SSL/TLS certificates to enable encryption on your web server.
- Force HTTPS: Configure your web server to redirect all HTTP requests to HTTPS, ensuring secure connections.

Any questions