Levereging Apache Guacamole, Linux LXD and Docker Containers to Deliver a Secure Online Lab for a Large Cybersecurity Course.

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

***Abstract*—This Innovate Practice Full Paper presents a technique to create a flexible, secure lab environment for a comprehensive practical course in cybersecurity.**

A common challenge for universities is integrating a hands-on learning environment as part of the cybersecurity curriculum. As a result, university programs that can allow students to practice and experiment with security tools similar to those used by both professionals and malicious actors are in high demand. Unfortunately, getting the needed IT infrastructure for a practical cybersecurity lab environment in academia is challenging for educators. Many instructors are therefore resorting to using virtual machines, containers, and cloud services to provide virtual labs environments that students can access anytime and anywhere. However, despite the vast benefits these technologies offer, deploying, configuring, and maintaining a secure virtual lab infrastructure is demanding and requires much preparation and testing.

Utilizing open-source technologies such as Apache Guacamole, Docker, and LXD containers, the author, managed to provide a simple, secure lab setup that only required a web browser to access the lab. Students could then work on individualized lab exercises using time-constrained ephemeral Linux containers with the necessary software and tools to complete their weekly assignments.

We deployed the solution in 2021 during the Covid-19 pandemic to an undergraduate cybersecurity class with 250 students. Furthermore, the anonymous online survey results were overwhelmingly positive, and the participants responded that accessing a Linux machine through the browser anytime, anywhere enabled them to learn the material effectively.

***Keywords*—Cybersecurity, Secure Online Lab, Linux terminal, Docker, LXD, Authentic Learning Environment**

I. Introduction

Network connectivity and the consumption of online digital services have exploded within the past decade, profoundly affecting and changing the fabrics of societies worldwide. Moreover, the ease at which digital services are accessible makes it convenient for us to use them, thus making them an essential and integral part of our lives.

Despite the considerable benefits of online digital services, governments, businesses, and consumers are experiencing a staggering growth in the number of cyberattacks [1], [2]. As a result, securing the critical services on which we are heavily dependent is challenging, and the demand for professionals with cybersecurity competency and technical skills is increasing every year [3], [4], and [5].

The EU, USA, and other countries worldwide have crafted policies and strategies to address the cybersecurity skills shortage and gap. Some suggested remedies are changes in higher education programs, improved collaboration between academics and industry, and an expansion in security certifications and internship opportunities. In particular, universities are considered a central component of educating skilled professionals to meet the increasing demand [6].

A common criticism regarding cybersecurity education programs is that an over-emphasis on theory and the lack of integrating hands-on learning environment into the curriculum [7]. Hands-on experience is consistently rated higher than all other aspects by the industry when considering recruiting new employees. In addition, providing an authentic learning environment where students practice on real-world cybersecurity challenges is recognized as one of the critical factors distinguishing between education programs. [4].

Theories and principles of Information Security(InfoSec) are a fundamental part of the curriculum; however, students need to acquire the practical skills required to execute that knowledge. Designing a practical cybersecurity course requires considerable preparation and resources. Nevertheless, one of the substantial advantages of providing hands-on lab environments is the opportunity for students to get familiar with the tools used by professionals and malicious adversaries. Unfortunately, obtaining the needed IT infrastructure for a practical cybersecurity lab environment in academia is often challenging for educators. [8].

Traditionally, universities provide physical computer labs for students. The IT department's domain is securing, running, and maintaining the infrastructure for these labs. It is also prevalent for IT departments to impose strict security policies on the usage of lab computers.

Therefore, educators have explored alternative solutions such as virtual machines, containers, and cloud services to provide virtual labs environments that students can access anytime and anywhere. However, despite the vast benefits these technologies offer, deploying a secure virtual lab infrastructure requires much preparation and testing.

II. Motivation and Background

Our department's cybersecurity course curriculum covers topics recommended by the ACM 2017 Cybersecurity Curricula Guidelines [9] and hands-on experience is an integral part of the course learning outcomes.

The university has several rooms equipped with computers containing the software needed for students to practice. However, changes introduced by the Ministry of Higher Education in recent years opened the possibility for universities to require students to have a modern laptop capable of supporting the software needed for their study program. As a result, students are no longer limited to only using physical workstations on campus but can use their laptops instead.

Due to strict security policies imposed on the usage and maintenance of lab computers, the cybersecurity course at our department has been employing VirtualBox as the mechanism for delivering hands-on lab environments to the students for many years.

VirtualBox is a virtualizing software for the x86 computing architecture [10] which is run locally on the student's laptop, thus eliminating the dependency on physical labs.

Virtualization separates processes executing in a virtualized environment from the underlying host or operating system by abstracting the hardware.

A. Challenges of using VirtualBox for the hands-on labs

The VirtualBox solution served us well throughout the years; nonetheless, there were several challenges that we constantly faced every year that we had to address.

- Students from the different undergraduate study programs had varying experiences with VirtualBox. Some were acquainted with it through other courses, while some had never used it before and often required extensive assistance.
- Modern CPUs include hardware virtualization features that vastly increase the performance of virtual machines. Unfortunately, some vendors disable this in the BIOS. Without the virtualization extensions enabled, VirtualBox, for example, will not start the virtual machines, or the VM will be so slow that it will be useless. As a result, the student will need help.
- A VirtualBox image conforming to the Open Virtualization Format (OVF or OVA) was prepared and distributed to the students at the start of the semester. The resulting image, which is made up of multiple Linux virtual machines that contain all of the software needed for the course curriculum, is a massive OVA file of around 10 GB. As a result, some students would often have issues downloading the image and importing it to their system.
- Students using computers with less than 8 GB of RAM found it difficult to run many virtual machines concurrently under VirtualBox. The OVA image provided to the students contained several virtual machines, each requiring a minimum of 1 GB of memory.
- We observed that an increased number of students show up with newer Apple laptops with an M1 ARM-based chipset. Unfortunately, as of this writing, VirtualBox does not support the M1 chipset and has no immediate plans to do so.

Furthermore, our university was also affected by the 2020 pandemic due to the Covid-19 virus. As a result, Lockdown and strict social distancing policies were imposed on higher education across the country. Prior to Covid, the cybersecurity course offered at our university to Computer Science students in the 5th semester primarily ran on campus in face-to-face mode. The schedule for each week consisted of two hours of lectures and two hours of hands-on lab sessions.

Apart from the issues we had prior to Covid-19, transitioning to complete online teaching for the cybersecurity course did not pose significant challenges, and any issues we encountered often related to students needing assistance with VirtualBox. Nevertheless, the pandemic accelerated the need to address our challenges before Covid-19.

B. Explored solutions

During our investigation[1] , several options were considered. Our main requirements were that it must be open-source, should not require extensive infrastructure or resources, be capable of handling a large number of users concurrently, and be easy to install and operate. We also wanted the solution to offer a level of secure, isolated environment similar to the one provided by the VirtualBox hypervisor. The following solutions were explored:

- Openstack
- Proxmox
- OpenNebula
- oVirt
- Kubernetes
- Openshift

The investigated solutions are widely popular and reported extensively in the literature as viable, robust solutions to deliver secure remote virtual lab environments. However, none of them fit our needs since they could require massive infrastructure and resources, a complex setup, or are hard to operate and manage.

III. Contributions of this Paper

This paper presents an innovative approach to creating a flexible, secure virtual lab environment for a comprehensive practical course in cybersecurity with over 250 enrolled students.

The novel contributions of this paper are the following:

1) A solution to access the lab remotely only through a web browser. Students could then work on individualizing lab exercises anytime and anywhere using lightweight, time-constrained Linux containers with the necessary software and tools to complete their weekly assignments. In addition, the launched containers are ephemeral, use

[1]Public clouds such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) were not considered as they would require a monthly subscription, and the budget needed to run the infrastructure was not feasible.

less than 256 MB of memory, and are disposed of when they are no longer needed, thus freeing CPU, memory, and storage resources.
2) By cleverly utilizing a small number open-source software, our solutions have fewer moving parts, and the whole virtual lab environment that over 250 students used was dynamically managed with four simple bash scripts shown in Listing 4, 5, 6 and 7.
3) While there are various virtual lab solutions based on virtualization or container technology in the literature, our novel contribution is the simplicity of our solution and the ease at which it can be managed, modified, and extended.

Details of the deployed solution is presented in Section V.

IV. Related work

Employing containers, virtualization, and cloud technologies to deliver virtual labs is extensively studied and widely recognized.

As early as 2003, educators were exploring the possibility of utilizing emerging virtualization technologies such as User Mode Linux(UML), Connectix Virtual PC, VMware Workstation, and Linux Kernel-based Virtual Machine (KVM) [11]–[14] to overcome the challenges of providing remote virtual lab environments. As virtualization technologies evolved and cloud services entered the IT domain, educators once again investigated the possibility of employing these new technologies. One such technology which has gained tremendous popularity is containers that allow us to run multiple isolated applications on a single server utilizing far fewer resources than virtual machines.

Sianipar et al. [15] presented a solution based on Docker containers extending the virtual machine-based Tele-Lab platform [16]. The solution provided access to virtual lab infrastructure through a web interface. While the main concepts are similar to our employed solution, the underlying architecture presented differs substantially from ours. The Tele-Lab architecture heavily relies on the Open Nebula cloud platform and employs various other software components such as frontend and backend applications.

Using Apache Guacamole as a remote desktop gateway, Marián et al. [17] proposed a solution that gave students access to physical training workstations to work on practical exercises. A similar approach is also presented by the authors of the paper [18] where a solution based on Apache Guacamole and virtual machines was used to deliver a Cyber Range training platform.

The study that closely resembles our solution is the study presented by Panum et al. [19]. Their solution Haaukins, uses Docker containers and Oracle VirtualBox virtual machines for deploying multiple isolated services within labs. Access to Haaukins is provided through Apache Guacamole.

Although some of the related work presented in this section has some parallels to our solution, none of them fully meets the needs of our unique use case.

TABLE I
Server specifications

Type	**Specification**
Vendor:	Dell
Model:	PowerEdge R6515
Processor:	AMD EPYC 7552
Number of cores:	48
Architecture:	x86-64
CPU GHz:	2.20
RAID System:	PERC 833 RAID controller
Total amount of memory:	755 GB
Total amount of storage:	2.0 TB
Operating System:	Ubuntu (Linux), 20.04

V. The deployed solution

Before we delve into our solution, we will first present an overview of the hardware and software used for the remote virtual lab.

A. The underlying infrastructure

The Computer Science department at our faculty owns and operates a small data center. A new hardware was purchased for the purpose of testing and deploying the solution presented in this paper.

The hardware is composed of one server with the specifications listed in Table I. In addition to the base Linux operating system, the following software were installed :

- **LXD:** an open-source container and virtual machine management system that builds upon Linux Containers (LXC) [20].
- **Docker:** provides the ability to package and run an application in a loosely isolated environment called a container. [21].
- **Docker Compose:** a tool for defining and running multi-container Docker applications. by using a YAML file as shown in Listing. 1. Then, with a single command, you create and start all the services from your configuration [22].
- **Apache Guacamole:** a clientless remote desktop gateway. It supports standard protocols like VNC, RDP, and SSH [23].

Installing Ubuntu as the base operating system is straightforward and trivial. In addition, Ubuntu provides packages for LXD, Docker, and Docker Compose. The packages were installed through the APT and SNAP package management system.

Apache Guacamole provides official Docker images simplifying the installation processes by using the Docker compose file in Listing 1. Executing the *docker-compose up -d* command would then pull the docker containers, install and run them on the system.

The author would like to emphasize that setting up the infrastructure which is a one time process is not a skill required from instructors and can be tasked to the IT department. Nevertheless, the author of this paper were able to set up the infrastructure with ease.

```
version: '3'
services:
  guacd:
    container_name: guacd
    image: 'guacamole/guacd'
    restart: unless-stopped

  guacdb:
    container_name: guacdb
    image: 'mariadb/server:latest'
    restart: unless-stopped
    environment:
      MYSQL_ROOT_PASSWORD: '**********'
      MYSQL_DATABASE: 'guacamole_db'
      MYSQL_USER: 'guacamole_user'
      MYSQL_PASSWORD: '**********'
    volumes:
      - ./guacdb-data:/var/lib/mysql

  guacamole:
    container_name: guacamole
    image: 'guacamole/guacamole:latest'
    restart: unless-stopped
    ports:
      - '8088:8080'
    environment:
      GUACD_HOSTNAME: 'guacd'
      MYSQL_HOSTNAME: 'guacdb'
      GUACD_PORT: '4822'
      MYSQL_ROOT_PASSWORD: '**********'
      MYSQL_DATABASE: 'guacamole_db'
      MYSQL_USER: 'guacamole_user'
      MYSQL_PASSWORD: '**********'
    depends_on:
      - guacdb
      - guacd

volumes:
  guacdb-data:
```

Listing 1. A Docker compose file for creating a MYSQL database and Guacamole container.

B. Configuring Apache Guacamole

Apache Guacamole is composed of two components:

1) guacamole-client: The HTML5 web application which serves the Guacamole client to users.

2) guacamole-server: The remote desktop proxy which the web application communicates with.

The guacamole-server is also responsible for communicating and connecting to resources that are presented to the web application. It supports the standard protocols VNC, RDP, and SSH and each protocol has its own set of configuration parameters.

The SSH connection type was configured in Guacamole to establish SSH connections to the host machine. This allows Guacamole to execute the scripts in Listing 6 and 7 which are needed to launch the containers. The SSH connection type requires at least the hostname or IP address, the port, and the credentials (the password or SSH private key) of the user that will connect to the host.

C. Docker and LXD images

The instructors created customized Ubuntu 20.04 Docker and LXD images containing software required for the cybersecurity course. We leveraged the automated build process provided by Dockerfile to build a custom image for Docker. Likewise, we used HashiCorps open-source Packer [24] tool to build custom LXD images. Although images could be built manually by adding the needed software to an existing container, the author opted for using DockerFile and Packer as they provide a consistent and better way of building and managing the images.

Furthermore, if additional software is needed for the curriculum, rebuilding the images is trivial using the automated methods. Listing 2 shows the Dockerfile used for our solution and Listing 3 shows the file used by Packer.

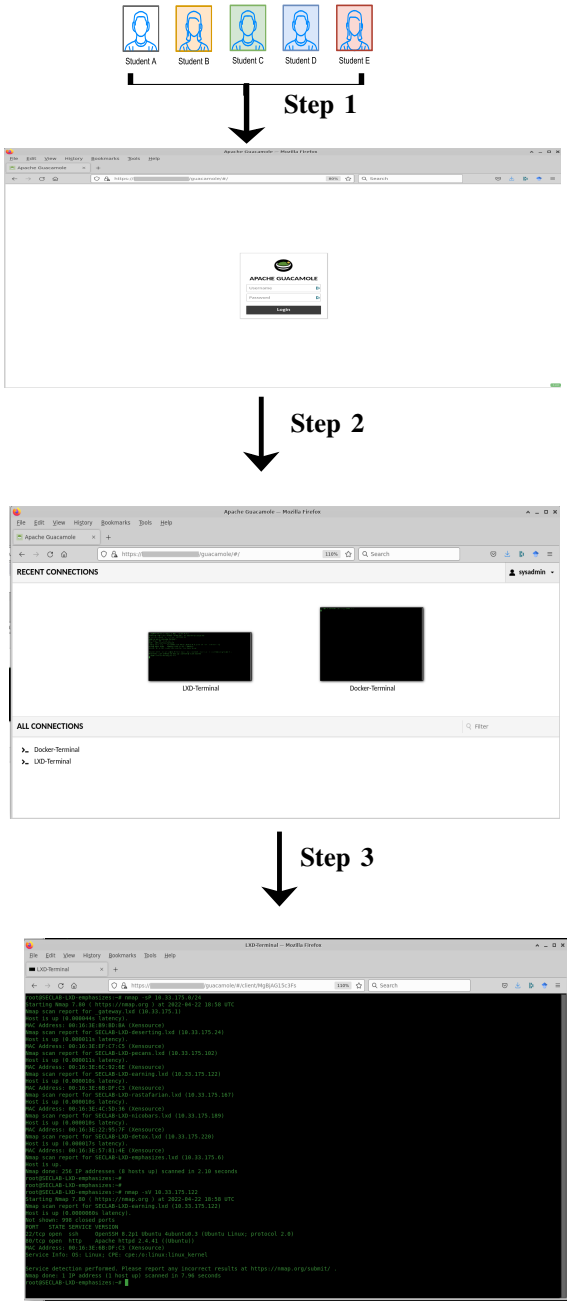

Fig. 1. The steps involved in getting access to the Linux terminal

D. Accessing the virtual lab environment

Students connect to the Apache Guacamole server using their preferred web browser and the provided URL address. They will then be presented a login screen as shown in Step 1 of Figure 1.

One of our goals was to keep the deployment simple, as a result, the instructor created a single user that all the students could use. The advantage of using the same single credentials is its simplicity. There is no need for user management or

```
FROM ubuntu:20.04

# Install all the packegaes needed for the course
RUN apt-get update && apt-get -y install nano openssl steghide python-pip man
      manpages mathomatic-primes apcalc
RUN apt-get -y install hydra cupp3 apache2 nmap sudo acl net-tools httpie bash-
      completion ccze libpwquality-tools

# Adds a user to the system
RUN useradd -m -d /home/seclabuser -s /bin/bash seclabuser

# Changes the password for the user
RUN echo "seclabuser:*********" | chpasswd

#Copies the directory containing files for the labs into the container
RUN mkdir -p /home/seclabuser/Labs
copy Labs /home/seclabuser/Labs

# Change to the directory
WORKDIR /home/seclabuser
```

Listing 2. A Dockerfile for creating a customized Docker image

```
{
  "variables": {
    "hostname": "seclab"
  },

  "builders": [
    {
      "type": "lxd",
      "name": "seclab-base",
      "image": "lab2021",
      "output_image": "ubuntu-seclab",
      "publish_properties": {
        "description": "Image for cybersecurity course"
      }
    }
  ],

  "provisioners": [
        {
            "type": "ansible-local",
            "playbook_file": "/opt/Ansible/apache-playbook.yml"
         }

      ]
}
```

Listing 3. A Packer file for creating a customized LXD image

integration with a centralized authentication mechanism that could complicate the setup. Instead, the Guacamole server uses a local MYSQL database to store and manage the user and will create a unique session for each login. The same user can log in to the Guacamole server concurrently, and the sessions are isolated from each other.

Once students login, as illustrated in Step 2 of Figure 1, they are presented with a page showing the available resources to them. Students can then click on desired resource to start the container. Depending on the resource chosen and as shown in Step 3 of Figure 1, Guacamole will connect to the host through SSH and execute a script that will launch the container selected and present the Linux terminal to the student.

E. Launching the containers

Listing 6 shows the script that will launch a random, ephemeral Docker container. Likewise, Listing 7 shows the script that launches the LXD container. The created containers are per default configured with resource and time limitations.

When students start a container, they will have approximately 2 hours to complete the practical assignment. The purpose of the time restriction, which the instructor can adjust, is to prevent students from misusing valuable resources. The scripts in Listing 4 and Listing 5 are scheduled to run every 15 minutes using the Linux cron service. The script will check if a container has been running for more than the maximum time

```
#!/bin/bash

# Setting the maximum amount of time the container is allowed to run
PREFIX="SECLAB-DOCKER"
MAX=2
#TIME="minutes"
TIME="hours"
#TIME="days"

docker ps |grep $PREFIX | while read -r line ; do
CONTID=$(echo $line |awk '{print $1}')
RUNING=$(echo $line |awk '{print $8}')
CONTIME=$(echo $line |awk '{print $9}')

# Check if the MAX value is reached
if [ $RUNING -gt $MAX ] && [ $TIME == $CONTIME ]; then
  docker kill $CONTID
fi

done
```

Listing 4. The script terminates Docker containers when MAX time is reached.

```
#!/bin/bash

PREFIX="SECLAB-LXD"     # Prefix as the search parameter.
MPHR=60                 # Minutes per hour.
MAX=120                 # Maximum amount of minutes the container is allowed to run
       .

lxc ls --format csv |grep $PREFIX | while read -r line ; do
NAME=$(echo $line |awk -F "," '{print $1}')
STATUS=$(lxc info $NAME |grep Status|awk '{print $2}')
CREATED_AT=$(lxc info $NAME |grep Created |awk '{print $2, " ", $3}')
TIME_NOW=$(date '+%s')
TIME_CREATED=$(date --date="$CREATED_AT" '+%s')
ELAPSED_TIME=$(( ($TIME_NOW - $TIME_CREATED) / $MPHR ))
echo $ELAPSED_TIME

# If the container
if [ $STATUS == "RUNNING" ] && [ $ELAPSED_TIME > $MAX ]; then
       echo "The container $NAME has been running for $ELAPSED_TIME minutes and
      will be terminated"
        lxc stop $NAME

fi

done
```

Listing 5. The script terminates LXD containers when MAX time is reached.

allocated and automatically delete it if the maximum time is exceeded.

F. Measures taken to secure the solution

By default, LXD and Docker start containers in unprivileged mode with a restricted set of capabilities provided by the Linux Kernel. LXD and Docker create a virtual private networks for the containers. These networks are contained within the

```
#!/bin/bash

# PREFIX is used to identify all the running lab containers.
PREFIX="SECLAB-DOCKER"

# A unique network that the lab containers are attached. The network is already
      created
NETWORK="LABS"

# We restrict the amount of memory and swap each container can use.
MEM_SIZE="256m"
SWAP_SIZE="256m"

# The name of the image in the local registry that we create the container from
CONT="ourlabs/lab:2021v3"

# Create a random lower case word
WORD=`shuf -n1 /usr/share/dict/american-english | sed 's/[^a-zA-Z0-9]//g' | tr '[:
      upper:]' '[:lower:]'`

# Combine the PREFIX with the random word to create a random unique container name
NAME=$PREFIX"-"$WORD2"


docker run --name $NAME --network=$LABS -m $MEM_SIZE --memory-swap $SWAP_SIZE -it
      --rm $CONT /bin/bash

End of script
```

Listing 6. The script starts the random ephemeral Docker containers.

```
#!/bin/bash

# A variable that we use as a prefix for the random container name.
PREFIX="SECLAB-LXD"

# We are creating a new variable that is assigned to a random string from the
      English dictionary
WORD=`shuf -n1 /usr/share/dict/american-english | sed 's/[^a-zA-Z0-9]//g' | tr '[:
      upper:]' '[:lower:]'`

# We create the random name by combining the two variables PREFIX and WORD
NAME=$PREFIX"-"$WORD

# Here we are starting the LXD container using the image Lab-2021-image and
      assigned a random name that starts with SECLAB.
# The container is also started with the option --ephemeral which will delete the
      container when stopped.
lxc launch --ephemeral -c limits.memory=256MiB Lab-2021-image $NAME

# End of script
```

Listing 7. The script starts the random ephemeral LXD containers.

server and per default not reachable from the outside world or university network.

In addition to the restriction provided by default, our IT engineers applied the following security measures:

- Enabled a firewall on the server that restricted the container network traffic to specific virtual networks and IP addresses. Students could then scan, connect and attack containers on those networks as part of the hands-on assignments without fear of causing any damage.
- Configured the firewall to restrict traffic between the containers, which prevented students from communicating or attacking each other's containers.
- The open-source Host Intrusion Detection System OSSEC was installed on the server to monitor potential malicious activities.
- Traffic to Apache Guacamole was routed through an Nginx proxy that provided SSL/TLS certificate termination, ensuring that all traffic between the student's browser and Guacamole was encrypted.

VI. Research Method and Data Collection

The research methodologies employed in this study are described in this section.

The general goal of our study is to understand how the participants perceived the online cybersecurity lab accessed through the browser, (a) to develop practical cybersecurity knowledge and skills through an authentic learning environment, (b) to get familiar with the cybersecurity tools used by professionals and malicious adversaries and (c) whether they preferred the web-based solution more than the VirtualBox solution.

A mixed methods approach was used for this study, including quantitative and qualitative data analysis.

At the end of the semester, a questionnaire survey was administrated to ask students about their experiences with the cybersecurity lab. The data was collected using Microsoft online Forms and students could submit the survey anonymously. Participation was voluntary and out of the 251 students enrolled in the class, 32.3% (**n = 81**) completed the questionnaire.

A. Quantitative method

For the quantitative approach, a set of online survey questions were administered to the class at the end of the semester using Microsoft online Forms. Table II shows the closed-ended survey questions. Participation was voluntary, and the students could submit the survey anonymously.

The survey asked participants to rate their agreement regarding their attitudes towards the statements in Table II. A 5-point Likert scale ranging from "Strongly Agree (5)" to "Strongly Disagree (1)" was used for the survey. Likert scales are mostly employed in questionnaires to collect participant's level of agreement with a set of statements.

TABLE II
Closed-ended survey questions.

Q1	The lab assignments in Canvas were flexible and not mandatory thus allowing students to do them at anytime, not constrained by deadlines and amount of attempts..
Q2	The lab assignments helped me develop practical Computer Security skills and improved my understanding of the subject content.
Q3	I like the fact that students had access to an actual virtual machine/container so that they could do hands-on practical computer security assignments in Canvas.
Q4	The Linux terminal provided and the tools in it allowed me to practice the security topics covered in the course.

Furthermore, open-ended questions were included in the survey to evaluate the course. Table. III shows the questions. The response to the open-ended questions were further analysed using qualitative methods.

TABLE III
Open-ended survey questions.

Q5	What parts of the course did you enjoy?
Q6	What parts of the course would you like to see improved?
Q7	Any other suggestions that you would like to recommend?

B. Qualitative method

Text analysis methods are commonly used in research to find attitudes or opinions about specific targets. For the qualitative approach in our study, word frequency and Network Text Analysis method are used to analyse responses from the open-ended question.

1) Word frequency: is a widely used analysis method to reduce and compress the language of a text. The method ignores the order in which the words appear in a document and counts the number of times words occur. Unfortunately, ignoring the order leads to losing the text's semantic meaning. However, the benefit of word frequency is that data will acquire properties that can be analyzed statistically [25].

2) Network Text Analysis: is a method for encoding the relationships between words in a text and constructing a network of linked words [26]. For instance, a corpus of texts may be represented as a network, with each node representing a document and the thickness or strength of the edges describing similarities between the words used in any two documents. [27], [28].

VII. Results and analysis

A. The Likert scale responses

Table IV shows the percent and frequency of the Likert scale responses to **Q1** of the closed-ended questionnaire.

On the question regarding the flexibility of the solution, 62.3% of the participants responded Strongly Agree, while 25.9% respectively responded agree. As shown on Table IV, a substantial percentage of the participants deemed the solution to be very flexible.

Furthermore, whether the solution provided an authentic learning environment to help develop practical cybersecurity skills, 34.6% and 48.1% responded Strongly Agree or Agree. The results on Table V show an overwhelmingly positive response regarding **Q2**.

The high cumulative percent of the participants that responded to Strongly Agree and Agree in **Q3** and **Q4** indicate a very positive attitude towards the solution provided. In addition, students enjoyed easy access to an actual Linux system.

The results of the closed-ended questionnaire show an overwhelming satisfaction with the solution. Students regarded the virtual lab environment provided as valuable supplement to the weekly theoretical lectures. The flexibility the solution offered was also highlighted in the open-ended questionnaire.

In the next section, we present the qualitative analysis of the open-ended responses, but first we would like to present in Table VIII, excerpts from the students' qualitative answers which support our finding in the open-ended survey.

TABLE IV
Frequency & Percent of Q1 responses.

	Frequency	Percent	Cumulative Percent
Strongly Agree	51	62.963	62.963
Agree	21	25.926	88,889
Neutral	6	7.407	96,296
Disagree	3	3.704	100.000
Strongly Disagree	0	0.000	100.000
Total	81	100.000	100.000

B. The Word frequencies

Furthermore, the student's responses to the open-ended qualitative question about what sections of the course they appreciated the most were used to build a word frequency plot. The most common terms in the students' responses are depicted in the Fig. 2. Lectures, demos, practical, lab, Linux, terminal, and assignment, for example, are evident as the most frequent words, thus showing the parts of the course they enjoyed the best.

TABLE V
Frequency & Percent of Q2 responses.

	Frequency	Percent	Cumulative Percent
Strongly Agree	28	34.568	34.568
Agree	39	48.148	82,716
Neutral	13	16.049	98,765
Disagree	1	1.235	100.000
Strongly Disagree	0	0.000	100.000
Total	81	100.000	100.000

TABLE VI
Frequency & Percent of Q3 responses.

	Frequency	Percent	Cumulative Percent
Strongly Agree	49	60.494	60.494
Agree	24	29.630	90.124
Neutral	7	8.642	98,766
Disagree	1	1.235	100.000
Strongly Disagree	0	0.000	100.000
Total	81	100.000	100.000

TABLE VII
Frequency & Percent of Q4 responses.

	Frequency	Percent	Cumulative Percent
Strongly Agree	39	48.148	48.148
Agree	29	35.802	83.950
Neutral	13	16.050	100.000
Disagree	0	0.000	100.000
Strongly Disagree	0	0.000	100.000
Total	81	100.000	100.000

TABLE VIII
Excerpts from the students' responses to the open-ended survey

Participant 77	"I enjoyed that there were lots of practical examples in the labs, and that the examples you gave were very useful."
Participant 46	"The use of terminals to complete task like such as using hydra (the practical part of the course)."
Participant 17	"Big focus on practical assignments that makes the course interesting. "
Participant 42	"Learning about security principles and then being able to see how it works in a practical setting made it easier to learn and understand the importance of computer security."

1) The Network text analysis: The text analysis was carried out using R and RStudio. Additionally, the textnets package was used to automate the Network Text Analysis of the open-ended responses. PrepText is a function included in the package that prepares the text for network analysis. In addition, PrepText can take other optional arguments, for example, removing the English stop words such as "a", "the", "is", "are" etc. Listing 8 shows the R script used to analyze the responses.

The graph in Figure 3 illustrates the text network constructed from the responses. The words that are frequently used together in the same context are closer together and have

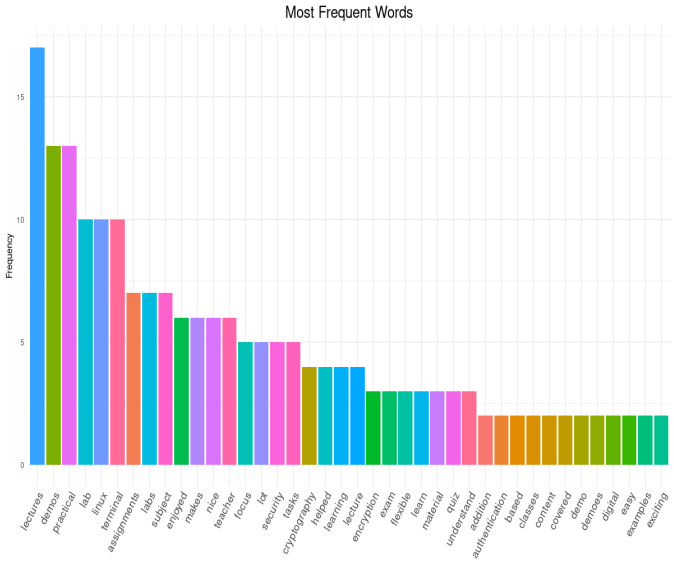

Fig. 2. Word frequency derived from responses of the open-ended questions in Table II

```
library(devtools)
install_github("cbail/textnets")
library(textnets)
# Set the working directory
setwd("~/Documents/Research/2022/FIE/Data")

csv_data <- read.csv(file = 'results-qualitative.csv')
#View(csv_data)
answers <- csv_data %>% group_by(ID)
View(answers)

# Analyze response to Q6 (What parts of the course did you enjoy?)
prepped_answers <- PrepText(answers, groupvar = "ID", textvar = "LIKE", node_type =
       "words", tokenizer = "words", pos ="nouns", remove_stop_words = TRUE,
      compound_nouns = TRUE)
answer_network <- CreateTextnet(prepped_answers)
VisTextNet(answer_network, label_degree_cut = 2)
```

Listing 8. The R script that conducts the Network Text Analysis and then visualizes it using the textnets package.

the same color on the graph.

The word nodes are arranged by measuring their network influence. The most influential nodes are the ones that connect different topics. For instance, in the graph in Figure 3, the terms terminal, lab, assignment, flexibility, and lecture appear as influential nodes.

The results of the closed-ended questions, together with the open-ended replies, indicate that the implemented solution was well received by the students.

VIII. Limitations and Future Work

The main limitation is the use of a single server. Although the server is powerful and handles the number of users and containers seamlessly, the potential for hardware failure is a looming prospect. The department has provided us with an extra server that we intend to integrate as a cluster to make the solution more robust and redundant.

Author presented the solution to other instructors running an operating system and Linux system administration courses. They were pleased with the simplicity of the solution and would like to use it for their courses.

One of the goals of this study was to investigate whether students preferred the web-based solution more than the VirtualBox solution. Unfortunately, we have not made a proper comparison and have not included it in this paper. We want to conduct a study comparing the two solutions in the future.

Apache Guacamole was mainly used to provide a Linux terminal through an HTML5 client such as a browser, but it supports other standard protocols such as VNC, and the author would like to investigate the possibility of providing GUI applications.

IX. Conclusion

This paper detailed our experience deploying an online secure virtual lab for a large cybersecurity course. The solution is simple to set up, secure, and easily expandable.

Results and analysis of the anonymous online survey found overall satisfaction with the solution. In addition, students highly appreciated the flexibility the solution offered to be able to work on the weekly practical hands-on weekly assignments at any time and that they had access to a Linux terminal via the web browser. Furthermore, the responses also indicated students' satisfaction with the authentic learning environments offering them plenty of practice with various cybersecurity tools.

References

[1] S. Altalhi and A. Gutub, "A survey on predictions of cyber-attacks utilizing real-time twitter tracing recognition," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 11, pp. 10 209–10 221, 2021.

[2] M. Albahar, "Cyber attacks and terrorism: A twenty-first century conundrum," *Science and engineering ethics*, vol. 25, no. 4, pp. 993–1006, 2019.

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

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

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

[6] J. R. Nurse, K. Adamos, A. Grammatopoulos, and F. Di Franco, "Addressing the eu cybersecurity skills shortage and gap through higher education," *European Union Agency for Cybersecurity (ENISA) Report*, 2021.

[7] W. A. Conklin, R. E. Cline, and T. Roosa, "Re-engineering cybersecurity education in the us: an analysis of the critical factors," in *2014 47th Hawaii international conference on system sciences*. IEEE, 2014, pp. 2006–2014.

[8] M. Thompson and C. Irvine, "Labtainers cyber exercises: Building and deploying fully provisioned cyber labs that run on a laptop." in *SIGCSE*, 2021, p. 1353.

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

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

[11] M. Schmitt, J. Hu, and C. Meinel, *Design and Implementation of a PHP-based Web Server for the Tele-lab IT-security*. Citeseer, 2003.

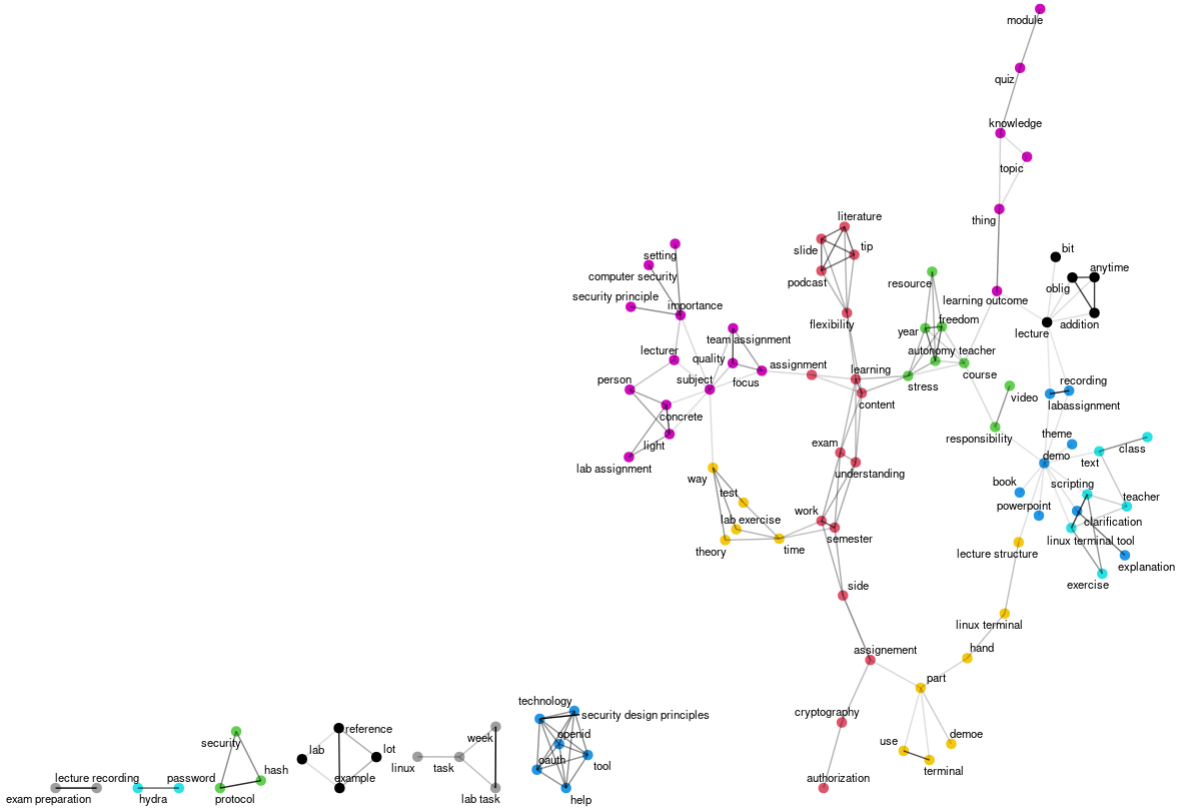

Fig. 3. Text Network derived from responses of Q6 of the open-ended questions in Table III

[12] K. Lei and P. T. Rawles, "Strategic decisions on technology selections for facilitating a network/systems laboratory using real options & total cost of ownership theories," in *Proceedings of the 4th conference on Information technology curriculum*, 2003, pp. 76–92.

[13] Y. Nakagawa, H. Suda, M. Ukigai, and Y. Miida, "An innovative hands-on laboratory for teaching a networking course," in *33rd Annual Frontiers in Education, 2003. FIE 2003.*, vol. 1. IEEE, 2003, pp. T2C–14.

[14] C. Kargin and T. Akgul, "Develop, build, and test a virtual lab to support a vulnerability training system," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Tech. Rep., 2004.

[15] J. Sianipar, C. Willems, and C. Meinel, "A container-based virtual laboratory for internet security e-learning," *International Journal of Learning and Teaching. IJLT*, vol. 2, no. 2, pp. 121–128, 2016.

[16] C. Willems and C. Meinel, "Tele-lab it-security: an architecture for an online virtual it security lab." *Int. J. Online Eng.*, vol. 4, no. 2, pp. 31–37, 2008.

[17] M. Jančík, L. Melníková, and D. I. B. Kandera, "Transformation of eurocontrol's classroom courses into remote learning," in *2021 New Trends in Aviation Development (NTAD)*. IEEE, 2021, pp. 66–69.

[18] T. Debatty and W. Mees, "Building a cyber range for training cyberdefense situation awareness," in *2019 International Conference on Military Communications and Information Systems (ICMCIS)*. IEEE, 2019, pp. 1–6.

[19] T. K. Panum, K. Hageman, J. M. Pedersen, and R. R. Hansen, "Haaukins: A highly accessible and automated virtualization platform for security education," in *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, vol. 2161. IEEE, 2019, pp. 236–238.

[20] "Lxd," 2022, https://linuxcontainers.org/, Last seen 22/04/2022.

[21] D. Inc, "Docker," 2022, https://docs.docker.com/, Last seen 22/04/2022.

[22] "Docker compose," 2022, https://docs.docker.com/compose/, Last seen 22/04/2022.

[23] "Apache guacamole," 2022, https://guacamole.apache.org/, Last seen 22/04/2022.

[24] "Hashicorp," 2022, https://www.packer.io/, Last seen 22/04/2022.

[25] G. C. Banks, H. M. Woznyj, R. S. Wesslen, and R. L. Ross, "A review of best practice recommendations for text analysis in r (and a user-friendly app)," *Journal of Business and Psychology*, vol. 33, no. 4, pp. 445–459, 2018.

[26] J. Diesner and K. M. Carley, "Using network text analysis to detect the organizational structure of covert networks," in *Proceedings of the North American Association for Computational Social and Organizational Science (NAACSOS) Conference*, vol. 3. NAACSOS Pittsburgh, 2004.

[27] A. Rule, J.-P. Cointet, and P. S. Bearman, "Lexical shifts, substantive changes, and continuity in state of the union discourse, 1790–2014," *Proceedings of the National Academy of Sciences*, vol. 112, no. 35, pp. 10 837–10 844, 2015.

[28] C. A. Bail, "Combining natural language processing and network analysis to examine how advocacy organizations stimulate conversation on social media," *Proceedings of the National Academy of Sciences*, vol. 113, no. 42, pp. 11 823–11 828, 2016.