CSU Cyber Practice Range

4/20/2022

Nathan Satterfield

Cybersecurity

Bachelor of Science

Julie Henderson

**Statement of Purpose**

The Charleston Southern cyber competition team currently lacks an environment to practice many of the skills and techniques that they are learning. For many students it is very difficult to truly learn and understand without hands on practice. This leads to students having the knowledge of what to do but without the experience to be able to implement that knowledge. The simplest way to fix this issue is to setup an environment where they can practice what they have learned leading to a better understanding of the topic. This would also increase the team’s performance in many of the competitions that they compete in as it could be used to simulate the competition environment.

The goal of this project is to provide Charleston Southern computer science and cyber security students somewhere to practice the skills they are learning. This will be accomplished by providing a server that will host several virtual machines to emulate the format of many of the competitions the students participate in. These virtual machines will be purposefully misconfigured to be insecure so the students can practice hardening systems. There will be several different operating systems included consisting of at least a few windows and Linux based systems. Also included will be a list of items that will be scored vulnerabilities and a guide on how to find and fix them along with information on why it is a vulnerability when possible. Along with the list of vulnerabilities, when possible, there will be an automatic scoring engine, so the students know how many of the scored issues, they have found without having to cheek the list. There will also be a check list of items to complete for each operating system type which will allow the students to follow along which will help them know what steps are most important and what they have completed along with what steps they have remaining.

**Research & Background**

Research was done into what systems already existed for this purpose as there was a cyber range in place although it was never fully completed. The existing infrastructure needed to be evaluated to see if it would work with the new system. The majority of the research done was on ESXI into how to get a limited but free license as well as how to create and setup the server. Research was also needed to find a way to score the vulnerabilities on the Linux systems as the CyberPatriot scoring engine only works for windows-based machines.

**Project Languages Software and Hardware**

There will be some scripts written using windows power shell, bat files, bash, and sed to automate many of the tasks for hardening the systems. For hardware there will need to be a server to host the virtual machines and networking equipment to be able to access the server. For software Windows and Linux operating systems will be needed as well as ESXI as a hypervisor to host the virtual machines. The CyberPatriot Scoring Engine will be needed to automate scoring on the machines from <https://www.uscyberpatriot.org/competition/training-materials/practice-images>. For the Linux operating systems, the pysel scoring system will be used which can be found at <https://github.com/FWSquatch/pysel>.

**Project Requirements**

Req #1 Enumerate network

Functional

Create a network map of the network that the cyber range will run on.

The goal of this project is to create a cyber range that can be accessed over the network which requires knowing how the network is currently setup so the cyber range can be added to the network.

After this requirement is completed there should be a finished network map available that describes the network.

Priority #1

Dependencies none

Req #2 Configure network

Functional

Ensure the network is configured properly to let students access the cyber range over the proper network.

If the network is misconfigured students will not be able to access the cyber range and thus will not be able to use it.

After this requirement is completed, students should be able to connect to the network that the range will run on

Priority #2

Dependencies #1

Req #3 Setup the host server.

Functional

Configure the server that will host the virtual machines.

This server is the base for the cyber range without it the virtual machines that make up the range will not be accessible by the students.

After this requirement is completed the host server for the cyber range should be ready to host virtual machines to students over the network.

Priority #3

Dependencies #1 and #2

Req #4 Create virtual machines.

Functional

Create the virtual machines that will make up the environment.

The virtual machines are what the students are going to use to practice on they are the main part of the project that the students will interact with.

After this requirement is completed there should be standalone virtual machines that students can use to practice on.

Priority #4

Dependencies none

Req #5 Host the virtual machines on the server.

Functional

Take the virtual machines that have been created and move them onto the sever to be hosted.

Once the virtual machines have been created and the host server configured the last step to create a functional cyber range is to combine the two so that students can access the virtual machines via the server.

After this requirement is completed there should a functional cyber range with virtual machines accessed over the network.

Priority #5

Dependencies #1-4

Req #6 Accessibility

Security/Access

The cyber range should be configured in such that it can only be accessed via the correct local area network.

This is important as unauthorized users should not be able to access the range so the server should only allow host to connect from the CSU computer lab network.

After this task is done the server should only be accessible from the CSU computer lab network.

Priority #6

Dependencies #1-5

Req #7 User capacity

Performance/ Capacity

The cyber range should be able to support up to 6 people concurrently.

This is important as most competitions are team competitions generally allowing up to six team members so this would allow the whole team to practice at the same time.

After this task is done the server should be able to support a minimum of six users connecting at the same time.

Priority #7

Dependencies #1-5

Req #8 expandability

Performance/ Scalability

The cyber range should be configured in such that new virtual machines can be added at any time.

This is important as what the students want to practice may change as new operating systems and updates to existing ones happen.

After this task is done the server should be configured such that adding new virtual machines or making changes to existing ones can be done.

Priority #8

Dependencies #1-5

Project Implementation Description & Explanation

The first step in create this virtual practice environment was to setup the server to host the virtual machines. To do this I used the ESXI hypervisor which after creating an account and logging in can be downloaded as shown in Fig 1.

Graphical user interface, text, application, email

Description automatically generated

Figure

Next, I needed to create a bootable usb with the image file for EXSI, to do this I used a tool called rufus. The first step is to plug in a removable drive and then select the image file that was previously downloaded as showin in Fig 2. Graphical user interface, text, application, email

Description automatically generated

Figure

Then I installed ESXI as you would with any operating system starting by plugging in the bootable usb created in the previous step. After following the instructions given during the instillation process after which you will be prompted to restart the machine. Then I needed to log into the system itself and set the Ip address and ensure that it was set to a static address. Once the network configuration is completed you will be able to access the ESXI interface by navigating to the provided Ip address. Once any security risks are accepted you will see a screen similar to the one shown in Fig 3.

Graphical user interface, text, application, email

Description automatically generated

Figure

On the page show in Fig 3 there are two options you can download the vSphere client, or you can access everything through the browser with the host client link. If you decide to use the web interface you should see a login screen as shown in Fig 4.Graphical user interface

Description automatically generated

Figure

The user can then log into the system and will be able to interact with the virtual machines that are hosted on the ESXI server. There are a few ways to create or add a virtual machine to the server, the first way is to upload the iso image file by going to the data store tab on the left and selecting datastore browser as seen in Fig 5.

Graphical user interface

Description automatically generated

Figure

You are then able to use the upload button to upload needed files as shown in fig 6.

Graphical user interface, application

Description automatically generated

Figure

Once the file is uploaded you need to go to the host tab and select create/register VM seen in Fig 7.

Graphical user interface, application

Description automatically generated

Figure

After selecting the option to create a virtual machine the process is the same as creating a virtual machine in any other hypervisor. The second way to create or add a virtual machine is to upload the files for an existing virtual machine as you would with the iso file shown previously and then when clicking the create/register VM button there is an option to import an existing system. The last way to add a machine is to make a copy of an existing machine by copying the files for the machine using the datastore browser and then going through the process of importing an existing machine.

To setup the virtual machines used for the project I started by uploading the files to create one ubuntu and one windows system. I then set up the windows machine using the process outlined above, Next in the virtual machine I went to the CyberPatriot site and downloaded the scoring engine they provide. After installing the scoring engine, I shutdown the system and then made two copies of the windows system. Then I imported the two copies of the original windows system to have three different window machines with the scoring engine installed. I then followed the same process with the ubuntu machines uploading and creating one machine then downloading the scoring engine with the command seen in Fig 8.



Figure

After cloning the repository move into the new directory that was created then you will need to install it. The first step is to run the commands show in Fig 9.

A screenshot of a computer

Description automatically generated with low confidence

Figure

Then run the installer that comes with the repository by running the command ./install.sh. I then shut the machine down and created the copies so there were three machines all setup with the scoring engine. The next step was to configure the items the scoring engines would give points for, this one done one image at a time starting with the windows machines. During the installation of the scoring engine there should be two shortcuts added to the desktop as shown in Fig 10.

A picture containing graphical user interface

Description automatically generated

Figure

The first shortcuts is used to configure the scored vulnerabilities and the second is used to view the current score and solved issues. So to configure the machines open the coach configuration tool and you should be greeted with a windows as shown in Fig 11.

Graphical user interface

Description automatically generated

Figure

This tools is very easy to use all that needs to be done to add a scored issue is to pick an item put the setting that you want it to be set to and then make sure the box in the scoring tab is checked. Once you have made your selections you can view list of all the scored items by going to the summary screen shown in Fig 12.

Graphical user interface, text, application, email

Description automatically generated

Figure

The screen shown in Fig 12 also allows you to export the list of issues and easily save it to an html file. After configuring each of the windows machines I exported the list of issues for backup purposes as well as to provide to users if they wish to see the full list of items to see what they missed. Once the windows machines were finished, I moved onto the Ubuntu machines which is slightly less user friendly to configure. To configure the pysel scoring engine you need to edit the file PySEL.conf inside the pysel directory where the scoring engine was installed. To add a new scored item, you need to reference one of the python scripts that is stored in the Event\_checks directory. This is done by giving each item a name then putting the name of the script to run. Then you will need to set the enable status a category, a number of points, the parameters to be use, a description and finally the message to show the user when they have fixed the issue. A few examples can be seen below in Fig 13.

Text

Description automatically generated

Figure

More details on how to properly configure a scored item or create a custom event check can be found at <https://github.com/FWSquatch/pysel>. Once the configuration file is saved to have the changes be reflected in the scoring engine the steps to install the scoring engine need to be repeated. This process was then repeated for each of the ubuntu machines.

After creating the virtual machines and configuring the scoring engines the next step was to configure the virtual machines to be purposely insecure by changing settings and downloading programs. To see a full list of scored issues you can view the configuration files stored in the solutions folder at <https://github.com/Nathan-Satt/Senior_Project/> but a few examples include unauthorized user accounts, valid user accounts missing, missing and or insecure passwords, disabling firewalls, installing insecure applications, changing configurations in running services to be less secure.

The final step was to configure backups of the machines so users could restore the virtual machines to their misconfigured state. This was done using the built-in snapshot functionality of ESXI. To take a snapshot of a machine select the machine you want to backup and then click the actions button as seen in Fig 14. Graphical user interface, text, application

Description automatically generated

Figure

Then you will need to hover over the snapshots sections which will display several options including adding or restoring to a snapshot. After selecting the option to create a snapshot you will see need to enter a name as well as a description as shown in Fig 15.

Graphical user interface, application

Description automatically generated

Figure

Once you enter the details you can select the take snapshot button and then a full backup of the machine will be saved. This can be restored at any point by going the options as shown in Fig 14 and selecting restore snapshot instead of creating one. This will remove any and all changes made to the system since the snapshot was taken causing the machines to be set to the misconfigured states.

The next step is to configure the user accounts and roles to allow students to log into and interact with the system. This is done by going to the management tab and then selecting security & users as seen in Fig 16.

Graphical user interface, text, application, email

Description automatically generated

Figure

To create a new role or group you then select the roles tab shown in Fig 16 and then the add role button. After selecting the add role button you will need to give the role a name and assign the role permissions as seen in Fig 17. Graphical user interface, text

Description automatically generated

Figure

Once a role is created, we can then create a user and assign it to the new role. To do this you can select the Users tab that can be seen in Fig 16. Then you will need to provide a name and a password for the new user as seen in Fig 18.

Graphical user interface, application

Description automatically generated

Figure

Now that the user is created to add it to the role, we create you need to go the host tab then select actions and permissions as seen in Fig 19.

Graphical user interface, text, application

Description automatically generated

Figure

Then you should be greated with the screen shown in Fig 20.

Graphical user interface, application

Description automatically generated

Figure

Next select add user you should see the screen shown in Fig 21

Graphical user interface, text, application, email

Description automatically generated

Figure

You can then select the user and role that we just created and select the add user button on the bottom right of the window. Now the user will be member of the role and have all of the permissions given to the role.

To start working on one of the virtual machines the user will need to connect to the proper network using the ethernet ports in Ashby Hall 203 and then navigate to the login page of the server. They will then need to login to the system using one of the existing accounts. After logging into the system, the user should see the screen shown in Fig 22.

Graphical user interface, text, application, email

Description automatically generated

Figure

The user will then need to navigate to the Virtual Machines tab shown in Fig 22. The user will then be able to view the list of machines on the server as shown in Fig 23.

Graphical user interface, application

Description automatically generated

Figure

To open a machine the user will then click on the name of the machine they wish to view, they should then see a screen similar to the one in Fig 24.

Graphical user interface, text

Description automatically generated

Figure

Now the user will be able to start, stop, suspend, or restart the machine right in the browser with the buttons along the top row seen in Fig 24. Once the machine is running a console can be opened by clicking the console button shown in Fig 24 and selecting where the console should be opened. Once the user opens a console, they will be able to interact with the virtual machine just as if it was running on their own system. At this point the users goal is to harden the system and find all of the scored issues configured using the scoring engines.

Along with the ESXI server and virtual machines I created a checklist of items that should be run though when securing a system. I also created two small scripts that automate some of the system hardening tasks one for windows and one for ubuntu. These scripts mainly focus on user auditing and can remove users as well as change passwords very quickly.

The last item included with this project is a detailed walkthrough document for each of the six currently existing virtual machines. The walkthroughs include screenshots of the actual process as well as explanations on what is being done.

For more documentation on ESXI you can read the full documentation at <https://docs.vmware.com/en/VMware-vSphere/index.html>. The full documentation for the ubuntu scoring engine can be found at <https://github.com/FWSquatch/pysel>. The full documentation for the windows scoring engine can be found at <https://www.uscyberpatriot.org/competition/training-materials/practice-images/>. To view configuration files and any other documentation check the github repository for this project at https://github.com/Nathan-Satt/Senior\_Project

Test Plan

ESXI Tests:

1. test connectivity to the machine hosting the virtual machines from AH 203 Ethernet ports.

2. test user account for ESXI to make sure they can be logged into.

3. test ESXI user permissions to see if they can interact with the virtual machines properly.

Virtual machine Tests:

1. test Windows scoring engine to ensure it works.

2. test Ubuntu scoring engine to ensure it works.

3. test one vulnerability from each category for each virtual machine.

4. test virtual machine backups to ensure that they can be reverted to a completely unsolved state without data lose.

Test Results

The first test was to ensure connectivity from the ethernet ports in Ashby Hall 203, to test this each of the four ethernet ports on the bottom row on the back wall were plugged into and the server was connected to. Each of the four ports behaved as expected and allowed a connection to the server. The second test was to make sure that users would be able to log into the premade accounts. To test this the server was connected to and the credentials were used to attempt to login. The admin account as well as all of the user accounts created were able to successfully login with no issues. The last test for the ESXI server was to ensure that users had the correct permissions. Users need to be able to login start virtual machines and restore them to the original state using the built-in snapshot feature. To test this the user accounts were logged into and then I attempted to start one of the virtual machines. This worked as expected and the users were able to start the machines. Finally, while logged in as a user I attempted to restore to an existing snapshot which was successful.

The second set of tests were for the virtual machines themselves. The first test in this section was for the Windows scoring engine to ensure it works. This was done after it was installed by ensuring that the scoring report was displayed and updated properly. This test was successful, and the scoring engine was working properly on all of the Windows machines. Next was the test for the Ubuntu scoring engine where the previous test was performed again but on the ubuntu engine. This test was successful as well. After the tests to ensure that the scoring engines were running properly, I begin to test the individual vulnerabilities to ensure that the scoring engine was configured properly and detecting changes correctly. This was done one virtual machine at a time by login into each system and fixing all of the scored issues. For the windows machines there were no issues, and everything was scored properly as seen in Fig XXXX-YYYYYText, letter

Description automatically generated

Text, letter

Description automatically generated

Text

Description automatically generated

After the windows machines were tested, I moved on to the ubuntu machines to begin testing them. There were no issues with the first and third ubuntu machine but there were some slight misconfigurations in the scoring engine for second machine. The script to detect the setting for the password warning age did not behave as expected so changes were made to detect the correct string in the given file. There was also an issue with the new user issue as at first the name of the user to be added was ed but was changed to edward as the string ed already existed in the passwd file inside the name of one of the other user accounts. After these issue were fixed all of the vulnerabilities were scored as expected as seen in Fig XXX-YYYTable

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

The final test was to ensure that the backups function properly and allowed a user to reset the image to its starting state. This was done by using the snapshot restore feature on each of the machines after the tests above in which all of the issues were fixed. After the snapshot was restored none of the issues had been resolved and the machine was back to the purposely misconfigured state. While there were a few issues initially all of the test passed after the proper changes to configurations were made.

Challenges Overcome

There were several challenges that needed to be overcome, this first was obtaining a license for ESXI after some research I found this webpage <https://customerconnect.vmware.com/en/group/vmware/evalcenter?p=free-esxi7> which after creating an account provides you with a license that allows for limited functionality. The next issue I ran into was a hardware compatibility with the Nehemiah server and ESXI. To fix this I was able to download an older version of ESXI which was compatible with the hardware in Nehemiah. The next problem I ran into was finding a scoring engine to work with the Ubuntu machines as the CyberPatriot does not provide one. After doing some research I found a scoring engine based on python that was available for Cyberpatriot coaches and mentors on github at <https://github.com/FWSquatch/pysel>. Once I found the pysel scoring engine I also ran into some issues getting it installed and running properly. To fix this all that needs to be done is to manually run some of the commands from the install script. The specific commands needed are used to start the service to run the scoring engine and can be seen in Fig xxxx. A screenshot of a computer

Description automatically generated with low confidence

The final challenge that I had was with the python script that monitored the password warning date. The issue was that it was looking to see if a string matching the regular '^PASS\_MIN\_DAYS\s\*0' I made a small change to the regular expression so that instead it was looking for the string needed to set the password warning age.

Future Enhancements

There are several improvements that could be made to this project, the first is to create more virtual machines which would allow users to practice on more than the six existing images. The second improvement that could be made is adding more events that the pysel scoring engine can check. This can be done by coding simple python scripts as described in the pysel documentation. The last main enhancement that could be made is to set up the environment in such a way that it could simulate an active attack environment. I believe it is possible to segment the virtual machines on different virtual networks which would allow the machines to be setup so that one group of students would be able to attempts to attack the other. This would help with active defense practice for competitions such as the Palmetto Cyber Defense Competition or the National Collegiate Cyber Defense Competition where a red team is actively working against blue team competitors.

Defense Presentation Slides