[redacted]

Automated Self-Managed Offensive Security Training Environment

By Dean

B0

Department of Informatics, School of Informatics and Engineering, XXXXXXXXXXXXXXXXXXXXXXX

Submitted to XXXXXXXXXXXXX in partial fulfillment of the requirements for the degree of

HIGHER CERTIFICATE IN SCIENCE IN COMPUTING IN NETWORKING TECHNOLOGIES

Supervisor:

xxxxxxxxxxxxxxxxx

20th May 2019

1 –Abstraction

The purpose of this project was to design a system or application during the work placement module which expressed and reflected on the students two years of XXXXX education (Higher Certificate Computer Science in Networking Technologies), for the XXXXXXXXXXXXXXXXXX, Ireland.

This project therefore incorporates adaptations from modules spanning the past two years of education such as Windows system administration, Linux system administration, computer infrastructures, virtualization technologies, business communications and others sciences.

Given that my work placement was as a junior member of a penetration team in a Security Department this project report is designed and tailored towards their needs. From discussions with the security department head and senior team members they felt that such a project within the training environment could enhance team cohesion, identify potential threats and prove beneficial for all new comers to the team.

This project allows the user to initiate contact through a web application to a web-server and introduce vulnerability into the operating system and over a period of time to expand from this template to disable individual machine on the system and subsequently delete them from the server. Research on public sites using Reddit and Stack Overflow reveals this has never been done. Previously undertaking this kind of Project would normally require the intervention of an internal systems administrator to build or delete the systems required.

The presented system in this paper was designed and built using VMware virtualization Technologies and specifically the vSphere application, a PHP web application and a Java application called Jenkins webUI that facilitates full server automation. Separate to this project paper the platform design allows the system to be 100% self-sufficient at any time without the need of the systems designer to control it, thus the platform may continue its usage within the company after the student had returned to education.

Table of Contents

2 -Brief

Work Based Learning & Project – 20 Credits Module

2.1 Description:

The aim of this module is to give students a deeper understanding of the industry in which they are placed for work experience. As part of the placement students will undertake a significant work based project to satisfy the requirements of this module. The project provides students with an opportunity to undertake a substantial project by taking responsibility for a full project life cycle. This module will enable students to consolidate and advance their chosen area of study. It will also integrate the knowledge and skills acquired from earlier modules that are part of this course. The students' Project Report will form a major component of the assessment of this module. The auxiliary benefits are numerous. The student is required to enhance their written and oral communication skills alongside their practical design and implementation skills. In conjunction with enabling intellectual development, this module also aims to motivate the student to develop their independence, confidence and self-esteem for they are required to:

- Plan, design, undertake and document a substantial work based project to a professional standard.
- Report on relevant existing work and do research relating to a project question, demonstrating appropriate academic referencing and presentation.
- Identify industry standard tools and technologies for the design, development and implementation of a work based project.
- Describe and reflect on the problems and challenges arising from a work based project conducted over a telling period of time.
- Describe (and apply) relevant system testing and evaluation methods
- Reflect on the learning outcomes achieved through the student's participation in the work placement.
- Plan, design, present an orally defend the project.

2.2 Assessment Details:

2.2.1 Work Placement Element – 8 Months in Duration

Work Placement Performance Evaluation – 25%
Reflective Journal – 15%
Presentation – 10%

2.2.2 Project Element – 3 Months in Duration (March – End of May)

Project Implementation and Written Report - 40%
Project Presentation and Defence - 10%*

*(Note that the project presentation and defence is a mandatory element and that no marks can be awarded for the project implementation if the project is not formally presented and defended.)

Deadline for the project submission is May 31st, with the presentation taking place on June 8th.

3 -Project Planning

Working in the security industry over the past two years as a penetration tester and member of a red team, I infer a penetration tester is an individual who searches for ways of exploiting systems. “Penetration testing is one of the oldest methods for assessing the security of computer systems”(FarkhodAlisherov A, 2009) and a member of a red team is a person who acts as the attacker on a network or more simply put “Red teaming is the process of studying a problem by anticipating adversary behaviours”(Ang Yang 2006).

On assessment I noticed that a lack of available resources or services to test vulnerabilities or software updates resulted in increased time on the penetration testing preparation stage. This meant that a further loss of time on the actual exploitation stage resulted in increased cost to both the client and the company during a security test.

Companies such as the one who wrote Cobalt Strike, a more professional version of Armitage the user friendly Linux penetration testing tool, welcome open-platform assistance. To this end they maintain a cluster of servers that could you could run their trail version of Cobalt Strike against. Unfortunately these servers were operating on out of date software and several uncommon operation systems to demonstrate their software for sale.

They may have thought it was a good idea but it required the use of many scripts and software tools that in essence were not secure as they overlooked the older vulnerabilities of the server’s operating system. This led the Red team to believe the operating system for testing the software on did not give sufficient evidence of functionality making it completely pointless, because doing a penetration test on a modern active network would contain updates, security patches or newer tool configurations.

It could be argued that the Penetration Team could just build a Local Virtual machine in a VMware workstation and install updates and configurations on that machine. This task would take a day or so to achieve as it requires installation, configuration and downloading the mandatory updates.

Personally speaking working on the Red Team and having a need to get an operating system up fast enough with the features needed to replicate the target system and then to run custom scripts or test vulnerabilities against it would take a lot of time out of the actual system testing timeframe. This requires spending extra time on the preparation and losing time from the actual testing and this obviously leads to increased costs to the clients and company.

According to Microsoft’s Windows Server Administration manual MCSA 70-410 Cert Guide, it can take anywhere from 45 minutes to 1.5 hours to install the base operating Windows Server 2012 R2 with the basic configurations and updates. However my real world experience with the Institute of Technology building machines for testing for the Windows System Administration course, it could take anywhere from 1.5 to 3 hours dependant on the Internet Connection and configurations needed for the network.

4 -The Solution

Good project planning should focus on developing a system that would give the Penetration Testers operating systems that are self-updated and use the most common preconfigured operation systems and applications to test the vulnerabilities.

This would be highly beneficial for the Red or even the Blue Team to test security patches or new updates before they are installed on production or development operating systems on a live network.

The difference between the Red Team and the Blue Team can be summed up by explaining how they work.

The Red Team and the Blue Team exercise methods of evaluating a security system by creating a ‘game’ where the Red Team attempts to attack a target and the Blue Team’ tries to defend the attack.

Operating systems that are self-updated offer many benefits beyond testing tools and scripts. This platform could also be used to allow Penetration Testers to practice for Certification exams such as OSCP “Offensive Security Certified Professional” or the new CompTIA PenTest+ exam. In fact this could be easily done as the only difference would by using pre-configured vulnerable operating systems instead of building operating systems. Since the OSCP exam has been around for a decent amount of time a lot of information has been released giving a reasonable expectation of the exam material, such as a Windows Server 2016 system with SMB vulnerabilities.

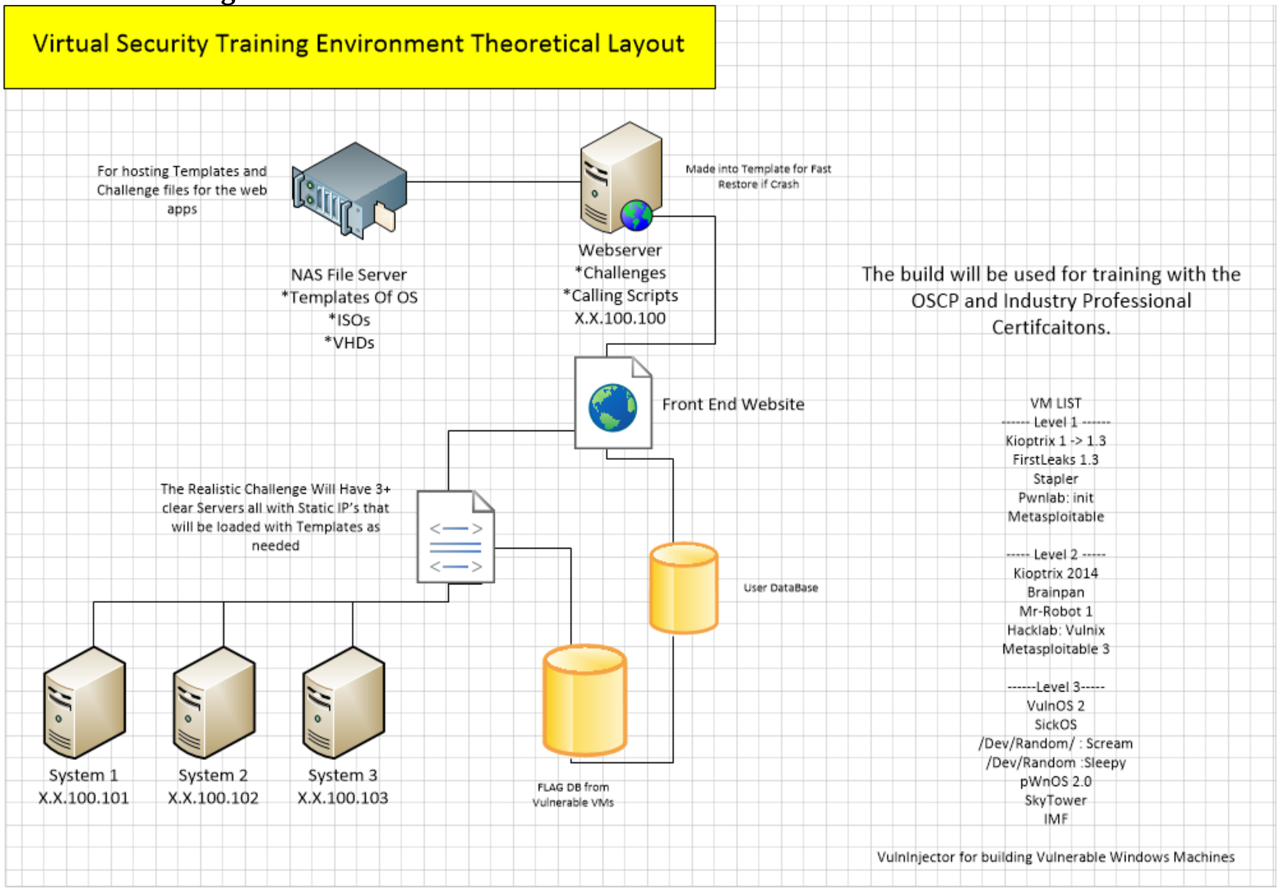
Such a system could be used to build test models with levels of vulnerabilities similar to what would be found in practical situations thus giving a person the ability to practice in an almost near real environment. Through research and discussions with Penetration Testers, such a tool does not appear to be widely available.

To deal with the limited resources on the ESXI server a script is required to run after a period of time say, 3-4 hours and then the server would auto shut down and the existing operating system is removed sending it back to a basic clean slate, awaiting the next assimilated assault.

So the ideal situation would be to build systems to the needed configurations and create a template of each machine, meaning they are available to be quickly reformatted to be up and running as soon as possible. This would also helps with storage as a template requires a lot less space and complications than hosting a virtual server. Should a templates fail, it wouldn’t take much to revert to a Snapshots to return the operating system to its original state.

The image inserted below is an impromptu design idea of how the whole system could be connected to the ESXI server to assist development with a visual aid.

4.1 - Solution Diagram


Virtual Security Training Environment Theoretical Layout
For hosting Templates and Challenge files for the web apps
NAS File Server
*Templates Of OS
*ISOs
*VHDs
Made into Template for Fast Restore if Crash
Webserver
*Challenges
*Calling Scripts
X.X.100.100
The build will be used for training with the OSCP and Industry Professional Certifcaitons.
Front End Website
The Realistic Challenge Will Have 3+ clear Servers all with Static IP's that will be loaded with Templates as needed
User DataBase
System 1
X.X.100.101
System 2
X.X.100.102
System 3
X.X.100.103
FLAG DB from Vulnerable VMs
VM LIST
------ Level 1 ------
Kioptrix 1 -> 1.3
FirstLeaks 1.3
Stapler
Pwnlab: init
Metasploitable
----- Level 2 -----
Kioptrix 2014
Brainpan
Mr-Robot 1
Hacklab: Vulnix
Metasploitable 3
------Level 3-----
VulnOS 2
SickOS
/Dev/Random/ : Scream
/Dev/Random :Sleepy
pWnOS 2.0
SkyTower
IMF
VulnInjector for building Vulnerable Windows Machines

5 -Literature Review and Research

5.1- Past Projects

The most important thing was to search if anyone had a similarly designed system that would interact with the ESXI server to initialise the building of Virtual Machines from a pre designed template. If there was a similar project it could give a generalised idea on how to start building the system.

From researching the topic and idea on forums and discussing with people in the community, there appeared to be no project available with the required designed system. This appeared to be a completely new concept.

5.2 -What is Virtualisation

Virtualisation is the creation of virtual systems that can mirror operating systems, storage devices and servers. It comes in two formats Hosted and Bare-Metal.

A hosted system is virtualisation installed on an OS such as VMware Workstation or HyperV installed on, for example, a Microsoft windows operating system, whereas a bare metal installation has no common operating system, such as Microsoft Windows, Linux or Apple Mac. On a metal installation the only software installed is the Virtualization software itself.

Virtualization gives companies and users the ability to scale up or down without the need to have individual systems for each application such as a server for DNS or DHCP or even a webserver. In a virtualised system this can all be hosted and controlled from one main location.

5.2.1 - What is a ESXI

ESXI is a bare metal hosted operating system developed by VMware. It is an enterprise class hypervisor with the goal of trying to minimise the footprint of the servers and services in an enterprise communications room. This is achieved by allowing multiple virtual machines to be installed on the bare metal host allowing services that would traditional required a standalone system to be able to function, for example DNS or DHCP can now be all run off the one server virtualised. This all means less space is required for the server, central maintenance is possible and less cost for server acquisition and although the points of failure are increased by having all services on one host the advantages nearly always out way the cons.

5.2.2 - What is vSphere

vSphere formerly known as VMware infrastructure is a suite of virtualization products developed by VMware for use on their ESXI servers. vSphere is an application that is installed in a currently running ESXI server which allows the administrator to have more pre-scripted features such as the ability to turn an active VM into a template for faster installing, or to clone a current VM to have an identical system for the purpose of fault tolerance.

5.2.3 - Scripting with VMware and vSphere

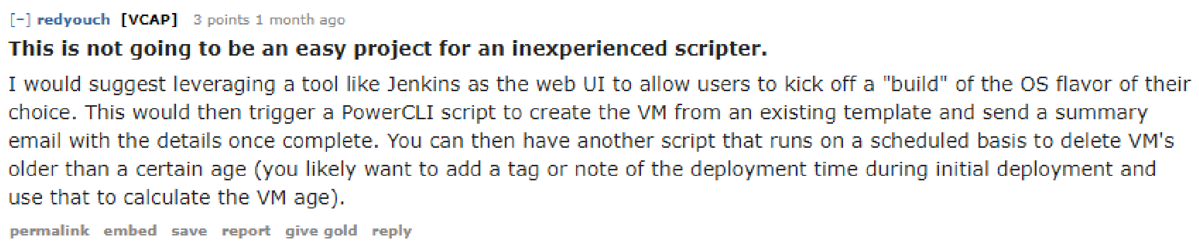
From reading the VMware manual on scripting for vSphere systems, called vsp4_41_vlci_inst_script.pdf, I have learned that it is indeed possible to write scripts that will interact with the ESXI server allowing configuration changes and options such as build from a template. The manual gives examples of commands for tasks in PowerShell, for example the command to power on the virtual machine looks like "vmware-cmd -H -U -P --vihost /vmfs/volumes/Storage2/testvm/testvm.vmx start soft" (vmware 2010). All commands are wrote and run via PowerShell, from the understanding of the paper.

5.3- Community Research

Community research for this subject is very limited due to the obscurity for the project and its design. Checking and posting to the normal programming and administration forums for any information or ideas proved very unsatisfying. Only one site Reddit proved useful for project material.

For the first time ever the site Stack Overflow completely failed to give feedback or even find relevant data. No one could answer the question I posed on how to commence design of the system, never mind have it running with self-sustainable.

Reddit users called the idea crazy and suggested that this was not a beginner’s project as it was very complicated, and it would not be able to function independent of an administrator input in the building stages. However one user “redyouch” offered a great idea, “redyouch” suggested looking at a Java application that builds a web user interface on the installed machine called Jenkins WebUI that has multiple functions facilitated by plugins.



[–] redyouch [VCAP] 3 points 1 month ago

This is not going to be an easy project for an inexperienced scripter.

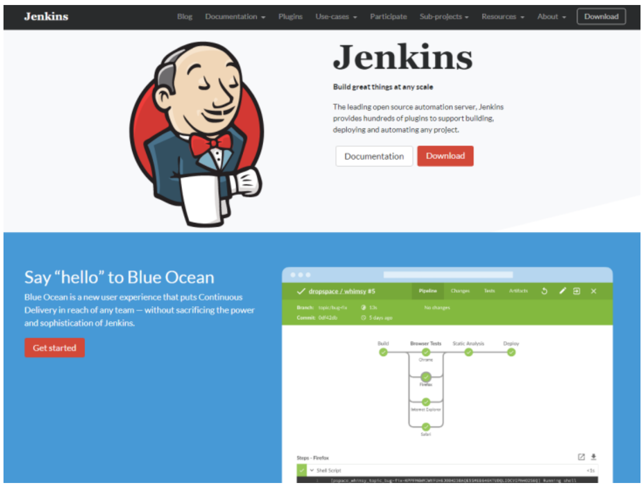
I would suggest leveraging a tool like Jenkins as the web UI to allow users to kick off a "build" of the OS flavor of their choice. This would then trigger a PowerCLI script to create the VM from an existing template and send a summary email with the details once complete. You can then have another script that runs on a scheduled basis to delete VM's older than a certain age (you likely want to add a tag or note of the deployment time during initial deployment and use that to calculate the VM age).

(Redyouch 2018)

5.4- What is JenkinsWebUI

Jenkins WebUI operation is based on plugins to keep the system slim and small, so to use this platform it needs to have a plugin installed that would allow it to be able to connect the ESXI. Lucky enough there was already a vSphere Cloud Plugin created by “Jswager” that allows features such as powering on a virtual machine through pre scripted PowerShell scripts built into the plug, deleting a virtual machine or most importantly to me to build a virtual machine from a template.

Bellow you can see an image of the Jenkins.io landing pages,



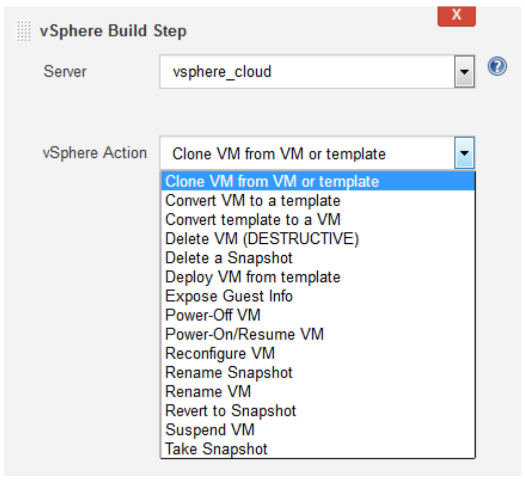
Build great things at any scale

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

Say “hello” to Blue Ocean

Blue Ocean is a new user experience that puts Continuous Delivery in reach of any team – without sacrificing the power and sophistication of Jenkins.

Below are some of the features the vSphere plugin by Jswager offers on the Jenkins platform.


vSphere Build Step

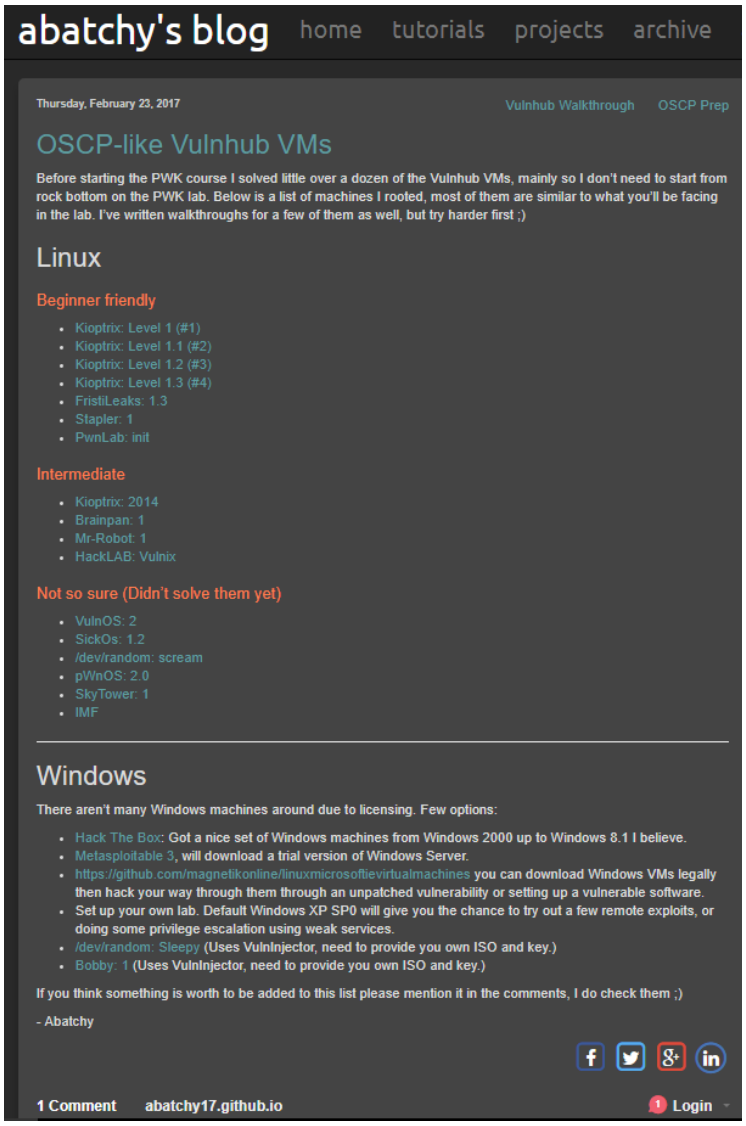
Server: vsphere_cloud

vSphere Action: Clone VM from VM or template

5.5 -Researching Operating Systems to use

This is an important step to be able to filter the needed virtual machines from the vast pool that there is prebuilt all over the internet. After some research I found a great list on a blog owned by “Abatchy” after he completed his PWK / OSCP exam he decided to share which vulnerable systems helped him practice for it, so I decided to base my virtual machines off his list. A Link to the blog post is available in the bibliography under the author’s name.

Below is a picture of the original blog post in case the blog goes offline for any reason.
(Abatchy 2017)



Thursday, February 23, 2017

Vulnhub Walkthrough OSCP Prep

OSCP-like Vulnhub VMs

Before starting the PWK course I solved little over a dozen of the Vulnhub VMs, mainly so I don't need to start from rock bottom on the PWK lab. Below is a list of machines I rooted, most of them are similar to what you'll be facing in the lab. I've written walkthroughs for a few of them as well, but try harder first ;)

Linux

Beginner friendly

- Kioptrix: Level 1 (#1)
- Kioptrix: Level 1.1 (#2)
- Kioptrix: Level 1.2 (#3)
- Kioptrix: Level 1.3 (#4)
- FristiLeaks: 1.3
- Stapler: 1
- PwnLab: init

Intermediate

- Kioptrix: 2014
- Brainpan: 1
- Mr-Robot: 1
- HackLAB: Vulnix

Not so sure (Didn't solve them yet)

- VulnOS: 2
- SickOs: 1.2
- /dev/random: scream
- pWnOS: 2.0
- SkyTower: 1
- IMF

Windows

There aren't many Windows machines around due to licensing. Few options:

- Hack The Box: Got a nice set of Windows machines from Windows 2000 up to Windows 8.1 I believe.
- Metasploitable 3, will download a trial version of Windows Server.
- https://github.com/magnetikonline/linuxmicrosoftievirtualmachines you can download Windows VMs legally then hack your way through them through an unpatched vulnerability or setting up a vulnerable software.
- Set up your own lab. Default Windows XP SP0 will give you the chance to try out a few remote exploits, or doing some privilege escalation using weak services.
- /dev/random: Sleepy (Uses VulnInjector, need to provide you own ISO and key.)
- Bobby: 1 (Uses VulnInjector, need to provide you own ISO and key.)

If you think something is worth to be added to this list please mention it in the comments, I do check them ;)

- Abatchy

1 Comment abatchy17.github.io 1 Login

5.6- Operating Systems

Part of the research for the project found that these pre-built vulnerable systems can offer the user a similar training environment to what they would expect to see on an OSCP certification. Currently there are limited systems running on the ESXI platform. This is in part due to the limited storage on the ESXI server and the department is currently seeking budget approval for upgrades to expand storage.

From personal experience in System Administration and conversing with fellow students and professional staff I have identified the most common operating systems. They are listed below, in the order that would be appropriate to allow the testers to test any needed vulnerability's or to test scripts against before releasing for general use.

5.6.1 -Operating System List

Vulnerable Operating Systems	**Current Build Operating Systems**
MrRobot CTF	Windows 95
Vuln VoIP	Windows XP
Basic Pentesting 1	Windows 7
Brainpan 1	Windows 8.1
First iLeaks 1.3	Windows 10
IMF 1	Windows Server 2003
Metasploitable 2	Windows Server 2012
Sick OS	Windows Server 2016
Vul	Ubuntu Desktop
MilNet	Ubuntu Server
Troll Cave	Red Hat Linux
VulNix	Cent OS
Vulnerable Docker	Apple Mac Sierra
Pwn Lab: init	IBM Z/OS Mainframe

Unfortunately having had to prioritise the virtual machines the penetration testing team felt it would be more important to have a working training environment than a patch testing environment, as that can be easily expanded in future when extra storage space becomes available. At current the virtual environments only contains Vulnerable Operating Systems to practice on.

5.6.1.1 - Operating System Information

Mr Robot CTF Is deigned on a Ubuntu system with hash key hidden each different locations each more harder than the next and is designed for remote exploitation, and is considered a beginner to intermediate challenge.

VulnVoip based on a old system called AsteriskNOW with multiple weakness, the aim of the attack is to get access to VoIP users and crack their passwords to gain access to privileged accounts, it is considered to be a more advanced challenge.

Basic Pentesting 1 is a very basic challenge with multiple remote vulnerabilities and multiple privilege escalation vectors to get advanced access on the system.

Brainpan an advanced remote system exploitation system with little write-ups, designed to test even the best penetration testers.

First iLeaks 1.3 made as part of a project by a Dutch hacking collective called FirstILeaks, it is designed to be cracked in a few hours without the use of many common tools for reverse engineering or debuggers.

IMF 1 designed to look like an intelligence agency you have to exploit and get root access to get the flag, designed as a relay race each flag leads to another. The application level is considered beginner to moderate.

Metasploitable 2 wrote by the Rapid 7 team designed to be exploited by tools, and vulnerabilities inside the Metasploit tool also wrote by Rapid 7 to help Penetration Testers experiment using the metasploit platform for attacks.

MilNet is designed with the specifications of older Department of Defence systems, to give the feel of attacking an older DoD system, it requires root level access to get the flag and also has multiple avenues of exploitation

Troll Cave set up to simulate war games, you start with no credentials at all and have to find or bruteforce passwords and usernames to get access to the root directory to find the flag files.

VulNix developed by Hacklab is a Linux system with vulnerabilities not through normal exploiting paths but by attacking miss configurations on the system itself.

Vulnerable Docker, since docker is become a major part of the security industry for hosting virtual machines and tool kits, it seems important to have a system that would allow the team to be able to practicing on a vulnerable docker, this is considered a harder challenge as the only way to exploit would be to attack miss configurations in the system.

Pwn Lab: init is a low level attack designed for new people to attack the system to learn exploitation, it is known as a Boot2Root system.

6 -Time and Costs

After outlining the design of the project to senior management they gave permission for the design and development of the platform to begin. The associated costs were in the form of power usage and network bandwidth as license keys for the operating systems such as Windows will operate on a trial basis, meaning no purchases was necessary .

6.1 -Costs:

Hardware	€0
Software	€0
Operating System Licences	€0

Overall costs for the Setup is zero 100% free as the software is all open sources, evaluations versions or freeware.

To build the system from scratch without the resources of the company could be within the range of €5,000. This cost would include purchasing a server for virtualization, purchasing the virtualization software from VMware, bandwidth usage, power consumption and required cables and fittings.

6.2 -Estimated Time:

Building of the Website est.5-6 hours
Writing of the Scripts and Fine Tuning est. 6-10 hours
Building Blank Systems to have templates installed on est. 1-2 hours
Building Operating System Templates est. 20-30hours
Write Presentation for staff on use est. 1 hour

6.3 - Actual Time Taken on project

Building of the website 3 hours taken
Writing of scripts needed and fine tuning of scripts 2 hours taken
Building of blank systems for templates 1 hour taken
Building of Operating System Templates (Not Applicable due to budget)
Writing of presentation for staff on use 30 minutes taken

7 -To Do List for the Project

- Research the Idea of the Server
- Research into using the JenkinWebUI
- Estimate Costs Associated with the project
- Build the website for Penetration Testers to log into
- Build web application for Penetration Testers To be able to use to call builds
- Write Server Side Scripts to interact between VMwareESXI server in Python
- Write Shell and Batch scripts to auto update systems one boot
- Build Templates of the Operations Systems needed
- Give access to the pen-testers to test the system
- Write Project Report
- Write a power point presentation on the use of the platform

8 -Building the Server

8.1 - The ESXI Server

Jenkins WebUI required the instillation of the vSphere application on the ESXI server to be able to control and manage the system with my configured builds. Thankfully VMware makes this relatively easy. Using a student access account to the VMware downloads, I was able to download the VMware VCSA ISO image, which contained the exe installation file that communicated with the ESXI server and installed the application over the server giving access to the vSphere module. vSphere is an application that allows an administrator to do so much more with the server such as more advanced options when creating or cloning virtual machines.

8.2 - The Webserver

The webserver was important for this build as it was going to be the main location for hosting the scripts and the website that will be used to communicate back to the ESXI server, as well as the Jenkin webUI Java application.

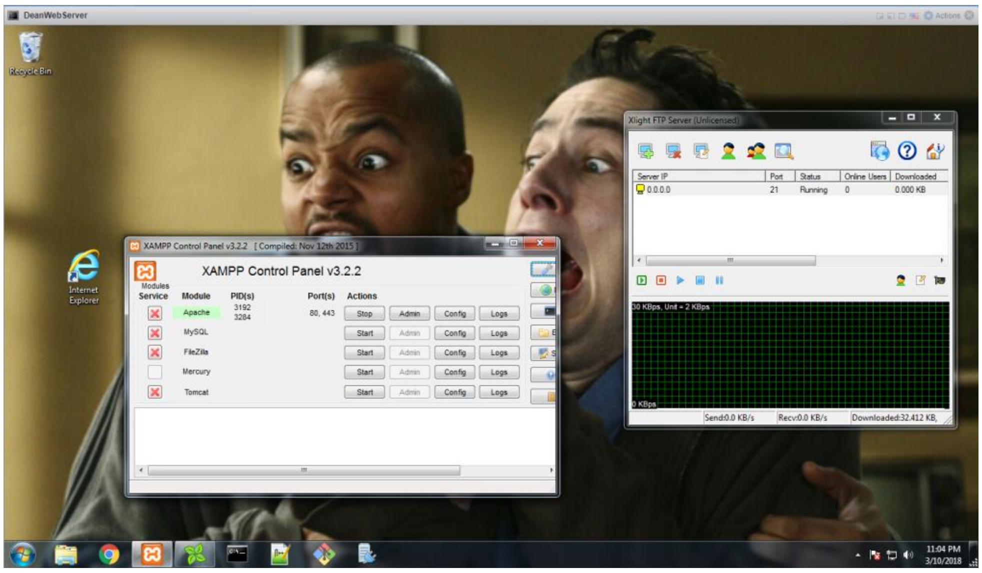
After much research I decided to use a windows 7 professional operating system to host the webserver as the company had a professional license key. This allowed the system to be constantly up and running and we would not need to worry about system crashes due to licensing restraints.

The first stage was to set up the Windows System on a virtual machine for the webserver, then install a virtual machine from an ISO image file. After the system was built, it was important to make sure the operating system was fully updated with the latest security and system patches.

From researching Jenkins UI, I found that it needs the latest java installation so I made sure to install that application from the oracle website, the installation of Jenkins is a simple processes as it is a one click run executable file for windows systems.

Since this is going to be the webserver making the informed decision to install XAMPP webserver, XAMPP is a small lightweight application that allows you to run an Apache Server local on the system. Being that the webserver is technically remote my system, the need to transfer files to the webserver while working on the server required a system that was formalised and comfortable with, using FTP “File Transfer Protocol” to transfer the HTML & PHP files as well as any server side scripts directly to the server. For setting up the FTP on the webserver a simple application called Xlight FTP Server was used. This application handles all the FTP configurations as well as security logging for the FTP server, making sure to have Xlights working directory as C:/xampp/htdocs/[SITE NAME] that way it immediately updates the current working Website

The next task was to set a static IP address of the webserver to 192.168.1.9, so it will always maintain the same IP address even if the server has a shutdown due to maintenance or error. vSphere has the option to force start the virtual machine, making sure to configure the static IP address is important so that if the system does shutdown due to error it should be able to come back online by itself and be ready to use.

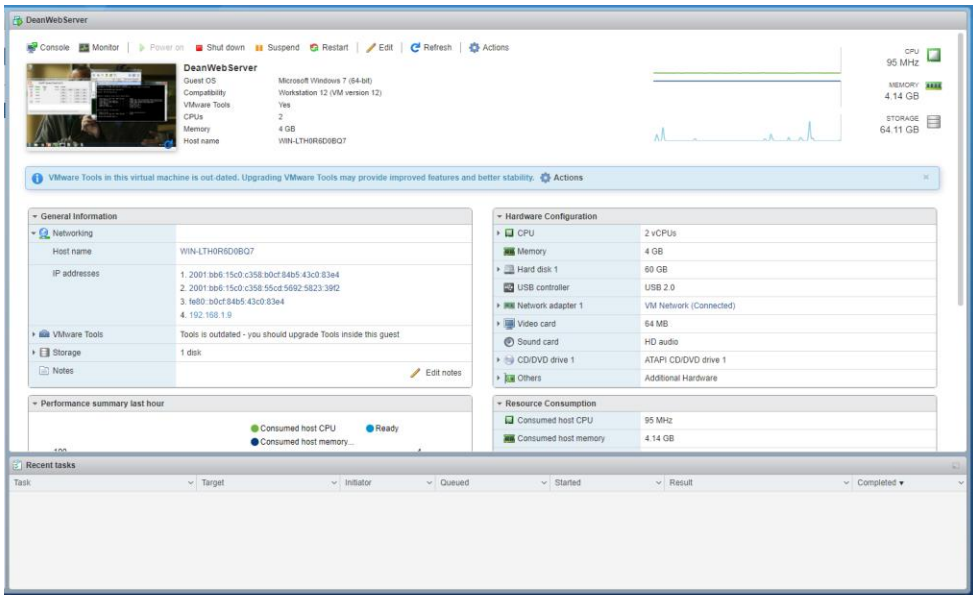

DeanWebServer
Recycle Bin
Internet Explorer
XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Xlight FTP Server (Unlicensed)
11:04 PM
3/10/2018

Modules Service	Module	PID(s)	Port(s)	Actions			
✗	Apache	3192 3284	80, 443	Stop	Admin	Config	Logs
✗	MySQL			Start	Admin	Config	Logs
✗	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
✗	Tomcat			Start	Admin	Config	Logs

Server IP	Port	Status	Online Users	Downloaded
0.0.0.0	21	Running	0	0.000 KB

```
30 KBps, Unit = 2 KBps
0 KBps
Send:0.0 KB/s   Recv:0.0 KB/s   Downloaded:32.412 KB,
```

This is an image of the desktop of the webserver running XAMPP and Xlight FTP Server



DeanWebServer

Guest OS	Microsoft Windows 7 (64-bit)
Compatibility	Workstation 12 (VM version 12)
VMware Tools	Yes
CPUs	2
Memory	4 GB
Host name	WIN-LTH0R6D0BQ7

VMware Tools in this virtual machine is out-dated. Upgrading VMware Tools may provide improved features and better stability. Actions

General Information

Networking	
Host name	WIN-LTH0R6D0BQ7
IP addresses	1. 2001:bb6:15c0:c358:b0cf:84b5:43c0:83e4 2. 2001:bb6:15c0:c358:55cd:5692:5823:39f2 3. fe80::b0cf:84b5:43c0:83e4 4. 192.168.1.9
VMware Tools	Tools is outdated - you should upgrade Tools inside this guest
Storage	1 disk
Notes	Edit notes

Hardware Configuration

CPU	2 vCPUs
Memory	4 GB
Hard disk 1	60 GB
USB controller	USB 2.0
Network adapter 1	VM Network (Connected)
Video card	64 MB
Sound card	HD audio
CD/DVD drive 1	ATAPI CD/DVD drive 1
Others	Additional Hardware

Performance summary last hour

● Consumed host CPU ● Ready
● Consumed host memory...

Resource Consumption

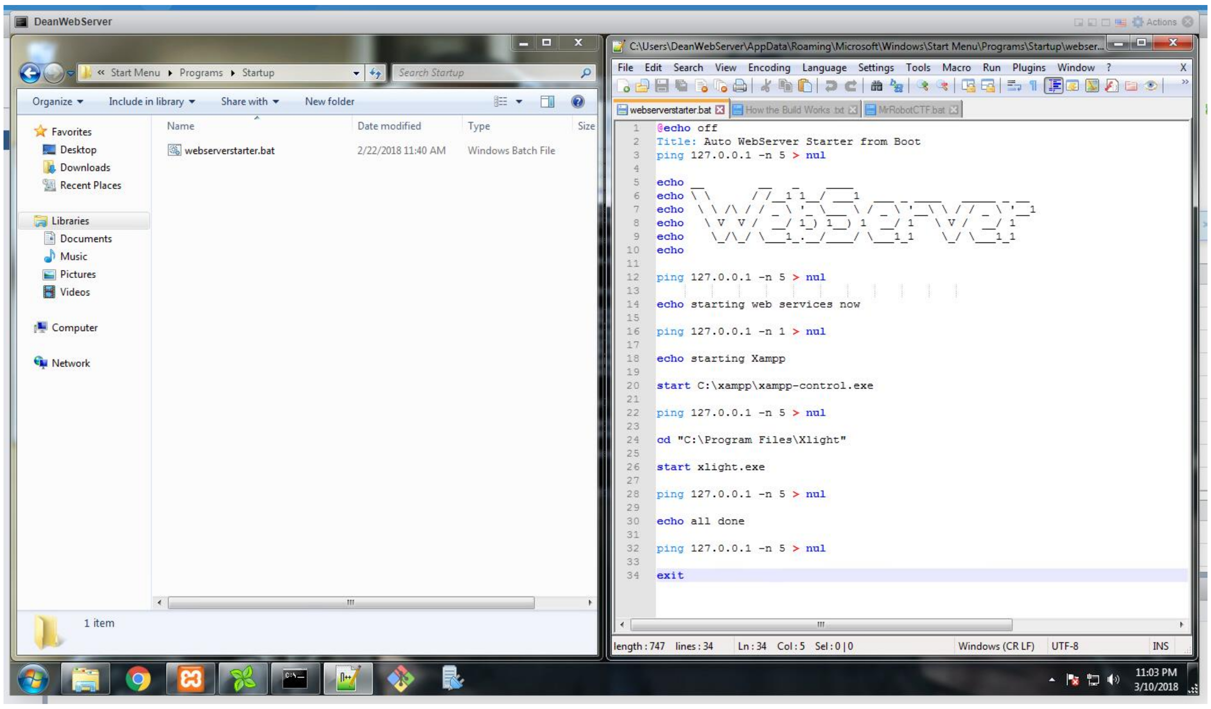
Consumed host CPU	95 MHz
Consumed host memory	4.14 GB

Recent tasks

Task	Target	Initiator	Queued	Started	Result	Completed ▾

This is an Image of the Webserver Running in VMware with its hardware configurations.

To automate the server’s start-up software in case the system automatically reboots for any reason such as windows updates required writing the below script and placing it in the start-up folder so when the system reboots the script will automatically run and start the required programs Xlight a very light open source FTP server with logging, and Xampp with its built in Apache2 webserver which would allow me to run the website simply and effectively.

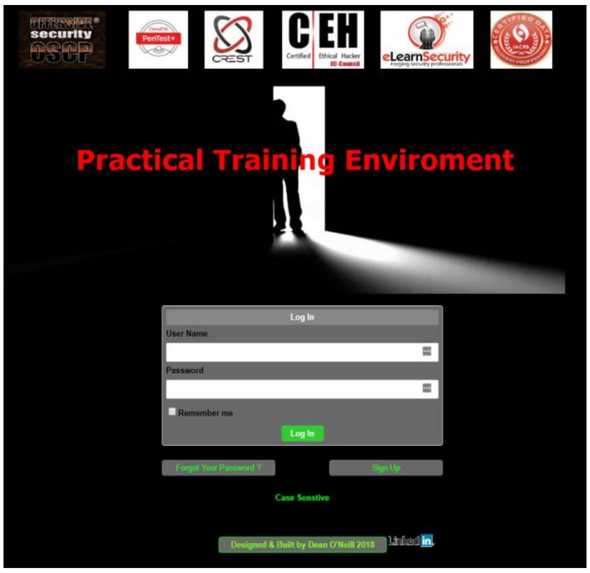


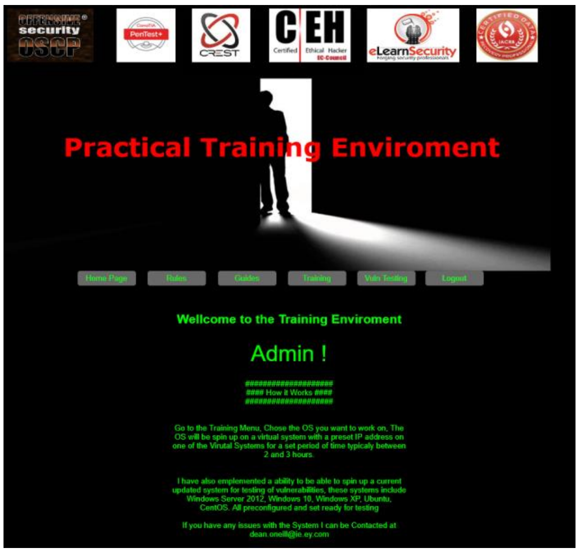
```
@echo off
Title: Auto WebServer Starter from Boot
ping 127.0.0.1 -n 5 > nul

echo __        __   _     ____
echo \ \      / /__1 1__ / ___1  ___ _ ____   _____ _ __
echo  \ \ /\ / / _ \ '_ \\___ \ / _ \ '__\ \ / / _ \ '__1
echo   \ V  V /  __/ 1_) 1___) 1  __/ 1   \ V /  __/ 1
echo    \_/\_/ \___1_.__/1____/ \___1_1    \_/ \___1_1
echo

ping 127.0.0.1 -n 5 > nul

echo starting web services now

ping 127.0.0.1 -n 1 > nul

echo starting Xampp

start C:\xampp\xampp-control.exe

ping 127.0.0.1 -n 5 > nul

cd "C:\Program Files\Xlight"

start xlight.exe

ping 127.0.0.1 -n 5 > nul

echo all done

ping 127.0.0.1 -n 5 > nul

exit
```

8.3 -Building the Website

For ease of use for the team it was important to build a landing page for the user with an email authenticated login. When they go to the URL they are greeted with a Login, this way it is possible to assure that only users from within the company have access to the system by using their email verification.

For the purpose of building the website it was important to do a re-fresher of my HTML and PHP knowledge. For this it was easy to find an udemy course. Udemy is an online education platform the best for this; with their micro courses such as learn php in an hour. Since this service is to be running long after the end of the employment, it was important to use a platform that won't fail. To this end the site would use a flat php file called user_db.php so there was no need to worry about failure of the MySQL, and that any possible failures could be resolved by reverting to an earlier snapshot or rebooting the windows system, whereas the MySQL server may need a few extra configurations to return to full working order.





Wellcome to the Training Enviroment

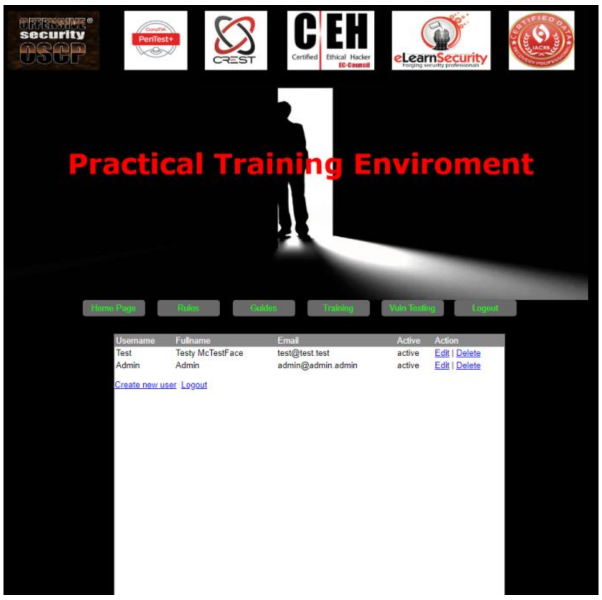
Admin !

```
####################
#### How it Works ####
####################
```

Go to the Training Menu, Chose the OS you want to work on, The OS will be spin up on a virtual system with a preset IP address on one of the Virutal Systems for a set period of time typicaly between 2 and 3 hours.

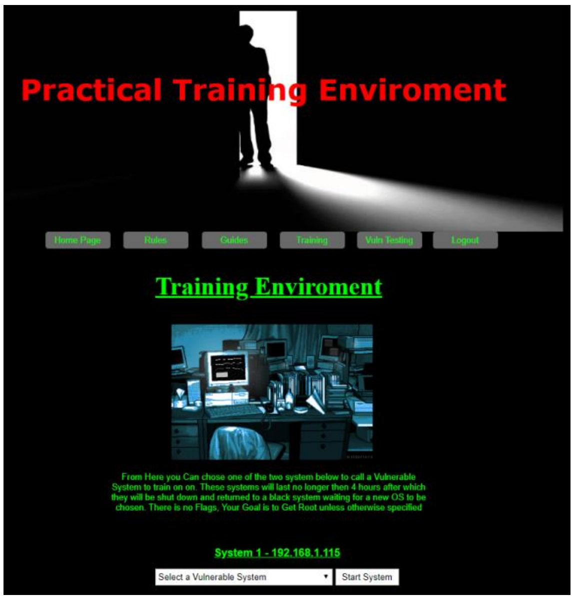
I have also emplemented a ability to be able to spin up a current updated system for testing of vulnerabillities, these systems include Windows Server 2012, Windows 10, Windows XP, Ubuntu, CentOS. All preconfigured and set ready for testing

If you have any issues with the System I can be Contacted at dean.oneill@ie.ey.com



Username	Fullname	Email	Active	Action
Test	Testy McTestFace	test@test.test	active	Edit | Delete
Admin	Admin	admin@admin.admin	active	Edit | Delete

Create new user Logout



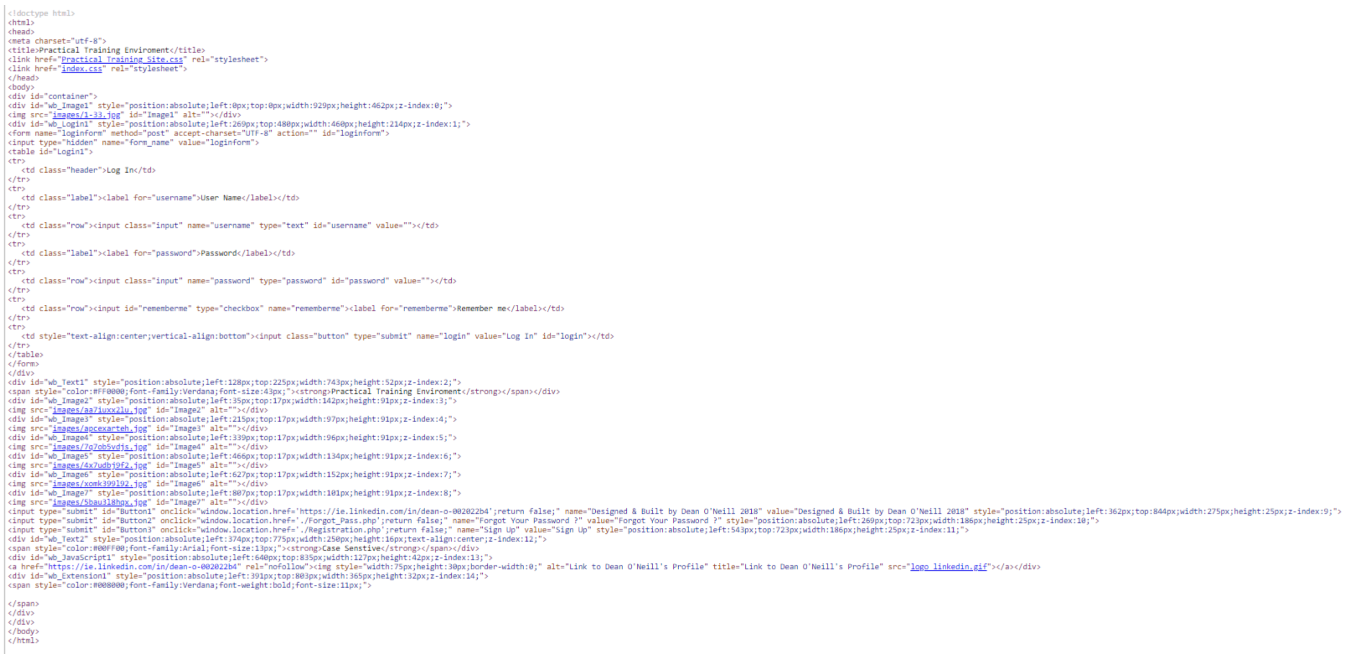
Training Enviroment

From Here you Can chose one of the two system below to call a Vulnerable System to train on on. These systems will last no longer then 4 hours after which they will be shut down and returned to a black system waiting for a new OS to be chosen. There is no Flags, Your Goal is to Get Root unless otherwise specified

System 1 - 192.168.1.115

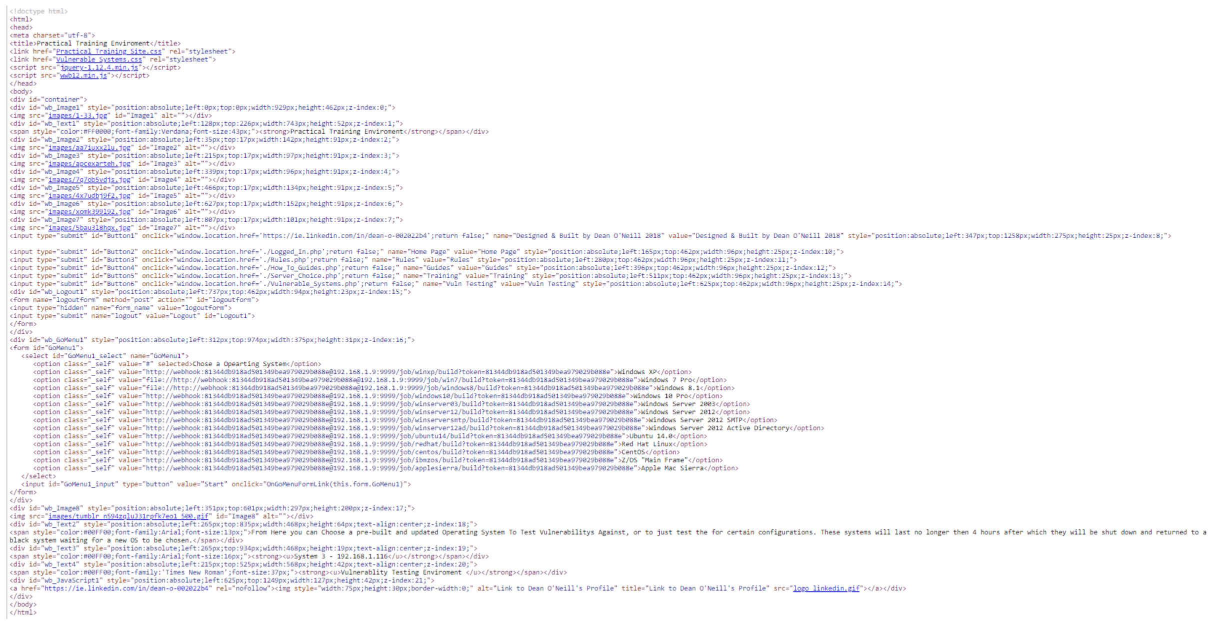
Above are pictures of the current website as it stands, Top left image is the user Login, bottom left picture is of the admin panel with view of the database, top right is the landing page when the server is logged into. Bottom right is the vulnerable system selection menu.

This is an image of the Code of the main login page,



```html
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Practical Training Enviroment</title>
<link href="Practical Training Site.css" rel="stylesheet">
<link href="index.css" rel="stylesheet">
</head>
<body>
<div id="container">
<div id="wb_Image1" style="position:absolute;left:0px;top:0px;width:929px;height:462px;z-index:0;">
<img src="images/1-33.jpg" id="Image1" alt=""></div>
<div id="wb_Login1" style="position:absolute;left:269px;top:480px;width:460px;height:214px;z-index:1;">
<form name="loginform" method="post" accept-charset="UTF-8" action="" id="loginform">
<input type="hidden" name="form_name" value="loginform">
<table id="Login1">
<tr>
   <td class="header">Log In</td>
</tr>
<tr>
   <td class="label"><label for="username">User Name</label></td>
</tr>
<tr>
   <td class="row"><input class="input" name="username" type="text" id="username" value=""></td>
</tr>
<tr>
   <td class="label"><label for="password">Password</label></td>
</tr>
<tr>
   <td class="row"><input class="input" name="password" type="password" id="password" value=""></td>
</tr>
<tr>
   <td class="row"><input id="rememberme" type="checkbox" name="rememberme"><label for="rememberme">Remember me</label></td>
</tr>
<tr>
   <td style="text-align:center;vertical-align:bottom"><input class="button" type="submit" name="login" value="Log In" id="login"></td>
</tr>
</table>
</form>
</div>
<div id="wb_Text1" style="position:absolute;left:128px;top:225px;width:743px;height:52px;z-index:2;">
<span style="color:#FF0000;font-family:Verdana;font-size:43px;"><strong>Practical Training Enviroment</strong></span></div>
<div id="wb_Image2" style="position:absolute;left:35px;top:17px;width:142px;height:91px;z-index:3;">
<img src="images/aa7iuxx2lu.jpg" id="Image2" alt=""></div>
<div id="wb_Image3" style="position:absolute;left:215px;top:17px;width:97px;height:91px;z-index:4;">
<img src="images/apcexarteh.jpg" id="Image3" alt=""></div>
<div id="wb_Image4" style="position:absolute;left:339px;top:17px;width:96px;height:91px;z-index:5;">
<img src="images/7q7ob5vdjs.jpg" id="Image4" alt=""></div>
<div id="wb_Image5" style="position:absolute;left:466px;top:17px;width:134px;height:91px;z-index:6;">
<img src="images/4x7udbj9f2.jpg" id="Image5" alt=""></div>
<div id="wb_Image6" style="position:absolute;left:627px;top:17px;width:152px;height:91px;z-index:7;">
<img src="images/xomk399l92.jpg" id="Image6" alt=""></div>
<div id="wb_Image7" style="position:absolute;left:807px;top:17px;width:101px;height:91px;z-index:8;">
<img src="images/5bau3l8hqx.jpg" id="Image7" alt=""></div>
<input type="submit" id="Button1" onclick="window.location.href='https://ie.linkedin.com/in/dean-o-002022b4';return false;" name="Designed & Built by Dean O'Neill 2018" value="Designed & Built by Dean O'Neill 2018" style="position:absolute;left:362px;top:844px;width:275px;height:25px;z-index:9;">
<input type="submit" id="Button2" onclick="window.location.href='./Forgot_Pass.php';return false;" name="Forgot Your Password ?" value="Forgot Your Password ?" style="position:absolute;left:269px;top:723px;width:186px;height:25px;z-index:10;">
<input type="submit" id="Button3" onclick="window.location.href='./Registration.php';return false;" name="Sign Up" value="Sign Up" style="position:absolute;left:543px;top:723px;width:186px;height:25px;z-index:11;">
<div id="wb_Text2" style="position:absolute;left:374px;top:775px;width:250px;height:16px;text-align:center;z-index:12;">
<span style="color:#00FF00;font-family:Arial;font-size:13px;"><strong>Case Senstive</strong></span></div>
<div id="wb_JavaScript1" style="position:absolute;left:640px;top:835px;width:127px;height:42px;z-index:13;">
<a href="https://ie.linkedin.com/in/dean-o-002022b4" rel="nofollow"><img style="width:75px;height:30px;border-width:0;" alt="Link to Dean O'Neill's Profile" title="Link to Dean O'Neill's Profile" src="logo_linkedin.gif"></a></div>
<div id="wb_Extension1" style="position:absolute;left:391px;top:803px;width:365px;height:32px;z-index:14;">
<span style="color:#008000;font-family:Verdana;font-weight:bold;font-size:11px;">

</span>
</div>
</div>
</body>
</html>
```

Below is a picture of the vulnerable choice System pages code



```html
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Practical Training Enviroment</title>
<link href="Practical Training Site.css" rel="stylesheet">
<link href="Vulnerable Systems.css" rel="stylesheet">
<script src="jquery-1.12.4.min.js"></script>
<script src="wwb12.min.js"></script>
</head>
<body>
<div id="container">
<div id="wb_Image1" style="position:absolute;left:0px;top:0px;width:929px;height:462px;z-index:0;">
<img src="images/1-33.jpg" id="Image1" alt=""></div>
<div id="wb_Text1" style="position:absolute;left:128px;top:226px;width:743px;height:52px;z-index:1;">
<span style="color:#FF0000;font-family:Verdana;font-size:43px;"><strong>Practical Training Enviroment</strong></span></div>
<div id="wb_Image2" style="position:absolute;left:35px;top:17px;width:142px;height:91px;z-index:2;">
<img src="images/aa7iuxx2lu.jpg" id="Image2" alt=""></div>
<div id="wb_Image3" style="position:absolute;left:215px;top:17px;width:97px;height:91px;z-index:3;">
<img src="images/apcexarteh.jpg" id="Image3" alt=""></div>
<div id="wb_Image4" style="position:absolute;left:339px;top:17px;width:96px;height:91px;z-index:4;">
<img src="images/7q7ob5vdjs.jpg" id="Image4" alt=""></div>
<div id="wb_Image5" style="position:absolute;left:466px;top:17px;width:134px;height:91px;z-index:5;">
<img src="images/4x7udbj9f2.jpg" id="Image5" alt=""></div>
<div id="wb_Image6" style="position:absolute;left:627px;top:17px;width:152px;height:91px;z-index:6;">
<img src="images/xomk399l92.jpg" id="Image6" alt=""></div>
<div id="wb_Image7" style="position:absolute;left:807px;top:17px;width:101px;height:91px;z-index:7;">
<img src="images/5bau3l8hqx.jpg" id="Image7" alt=""></div>
<input type="submit" id="Button1" onclick="window.location.href='https://ie.linkedin.com/in/dean-o-002022b4';return false;" name="Designed & Built by Dean O'Neill 2018" value="Designed & Built by Dean O'Neill 2018" style="position:absolute;left:347px;top:1258px;width:275px;height:25px;z-index:8;">

<input type="submit" id="Button2" onclick="window.location.href='./Logged_In.php';return false;" name="Home Page" value="Home Page" style="position:absolute;left:165px;top:462px;width:96px;height:25px;z-index:10;">
<input type="submit" id="Button3" onclick="window.location.href='./Rules.php';return false;" name="Rules" value="Rules" style="position:absolute;left:280px;top:462px;width:96px;height:25px;z-index:11;">
<input type="submit" id="Button4" onclick="window.location.href='./How_To_Guides.php';return false;" name="Guides" value="Guides" style="position:absolute;left:396px;top:462px;width:96px;height:25px;z-index:12;">
<input type="submit" id="Button5" onclick="window.location.href='./Server_Choice.php';return false;" name="Training" value="Training" style="position:absolute;left:511px;top:462px;width:96px;height:25px;z-index:13;">
<input type="submit" id="Button6" onclick="window.location.href='./Vulnerable_Systems.php';return false;" name="Vuln Testing" value="Vuln Testing" style="position:absolute;left:625px;top:462px;width:96px;height:25px;z-index:14;">
<div id="wb_Logout1" style="position:absolute;left:737px;top:462px;width:94px;height:23px;z-index:15;">
<form name="logoutform" method="post" action="" id="logoutform">
<input type="hidden" name="form_name" value="logoutform">
<input type="submit" name="logout" value="Logout" id="Logout1">
</form>
</div>
<div id="wb_GoMenu1" style="position:absolute;left:312px;top:974px;width:375px;height:31px;z-index:16;">
<form id="GoMenu1">
   <select id="GoMenu1_select" name="GoMenu1">
      <option class="_self" value="#" selected>Chose a Opearting System</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/winxp/build?token=81344db918ad501349bea979029b088e">Windows XP</option>
      <option class="_self" value="file://http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/win7/build?token=81344db918ad501349bea979029b088e">Windows 7 Pro</option>
      <option class="_self" value="file://http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/windows8/build?token=81344db918ad501349bea979029b088e">Windows 8.1</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/windows10/build?token=81344db918ad501349bea979029b088e">Windows 10 Pro</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/winserver03/build?token=81344db918ad501349bea979029b088e">Windows Server 2003</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/winserver12/build?token=81344db918ad501349bea979029b088e">Windows Server 2012</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/winserversmtp/build?token=81344db918ad501349bea979029b088e">Windows Server 2012 SMTP</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/winserver12ad/build?token=81344db918ad501349bea979029b088e">Windows Server 2012 Active Directory</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/ubuntu14/build?token=81344db918ad501349bea979029b088e">Ubuntu 14.0</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/redhat/build?token=81344db918ad501349bea979029b088e">Red Hat Linux</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/centos/build?token=81344db918ad501349bea979029b088e">CentOS</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/ibmzos/build?token=81344db918ad501349bea979029b088e">Z/OS "Main Frame"</option>
      <option class="_self" value="http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/applesierra/build?token=81344db918ad501349bea979029b088e">Apple Mac Sierra</option>
   </select>
   <input id="GoMenu1_input" type="button" value="Start" onclick="OnGoMenuFormLink(this.form.GoMenu1)">
</form>
</div>
<div id="wb_Image8" style="position:absolute;left:351px;top:601px;width:297px;height:200px;z-index:17;">
<img src="images/tumblr_n594zqluJ31rpfk7eo1_500.gif" id="Image8" alt=""></div>
<div id="wb_Text2" style="position:absolute;left:265px;top:835px;width:468px;height:64px;text-align:center;z-index:18;">
<span style="color:#00FF00;font-family:Arial;font-size:13px;">From Here you can Choose a pre-built and updated Operating System To Test Vulnerabilitys Against, or to just test the for certain configurations. These systems will last no longer then 4 hours after which they will be shut down and returned to a
black system waiting for a new OS to be chosen.</span></div>
<div id="wb_Text3" style="position:absolute;left:265px;top:934px;width:468px;height:19px;text-align:center;z-index:19;">
<span style="color:#00FF00;font-family:Arial;font-size:16px;"><strong><u>System 3 - 192.168.1.116</u></strong></span></div>
<div id="wb_Text4" style="position:absolute;left:215px;top:525px;width:568px;height:42px;text-align:center;z-index:20;">
<span style="color:#00FF00;font-family:'Times New Roman';font-size:37px;"><strong><u>Vulnerablity Testing Enviroment </u></strong></span></div>
<div id="wb_JavaScript1" style="position:absolute;left:625px;top:1249px;width:127px;height:42px;z-index:21;">
<a href="https://ie.linkedin.com/in/dean-o-002022b4" rel="nofollow"><img style="width:75px;height:30px;border-width:0;" alt="Link to Dean O'Neill's Profile" title="Link to Dean O'Neill's Profile" src="logo_linkedin.gif"></a></div>
</div>
</body>
</html>
```

As you can see from the code when an option is selected the link redirects to use the API token for a lower privilege user account called Webhook and its API token to request the server to build which then follows the stages of the present configurations on Jenkins build.

8.4 -ESXI Connection

For the connection to work between the website and the API being run on the JenkinsUI that is running on the webserver a server side script was needed to be written in batch, which can be seen below. When the option is selected on the website, a JavaScript function will run the .bat file on the server that will request a build through the Jenkins API user token. This is very inefficient as a script needs to be created for each of the builds taking up storage space.

```
@echo
title WebServer To JenkinsUI
echo.
echo Text only for Bug Detection
start chrome http://[IP ADDRESS]:[PORT]/job/[BUILD NAME]/build?token=[API TOKEN]
ping 127.0.0.1 -n 59 >NUL
TASKKILL /IM chrome.exe /F
ping 127.0.0.1 -n 9 >NUL
exit
```

Since building this script a better option has become available, instead of requesting the .bat file to be run on the server, there is a better way to request the build via direct URL link with the API token set by the app, as can be seen below.

http://webhook:81344db918ad501349bea979029b088e@192.168.1.9:9999/job/mrrobot/build?token=81344db918ad501349bea979029b088e

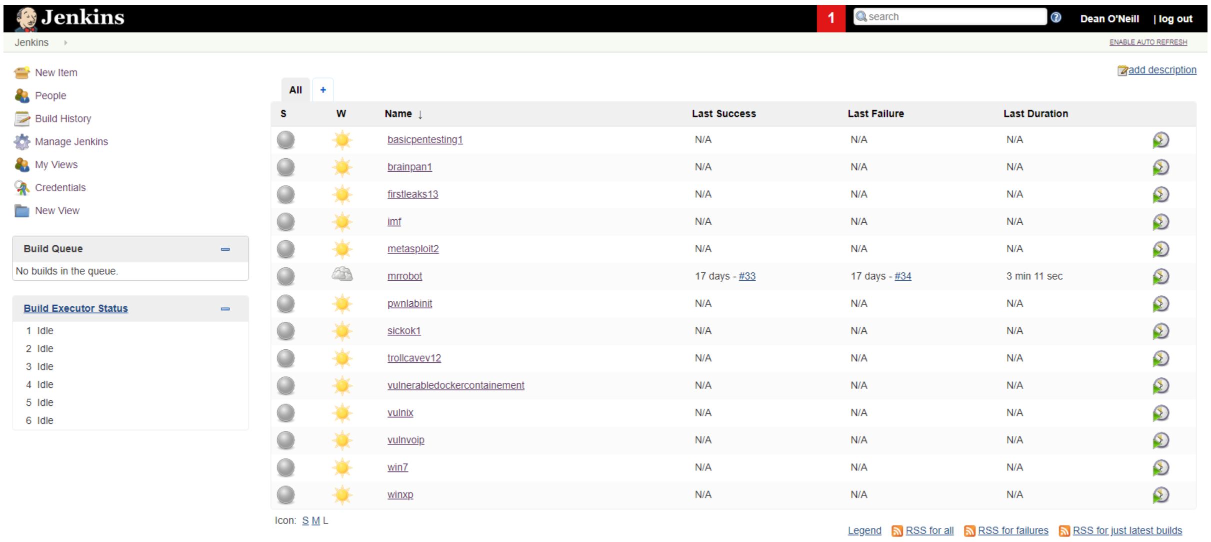
http://[USER]:[API TOKEN]@[SERVER IP]/job/[BUILD NAME]/build?token=[API TOKEN]

This puts less strain on the webserver by having multiple scripts through command prompts at the same time, although it decreases the security of the website as it displays the servers IP & port, the username and the API token. However it is more efficient for the user and since the system is on the Lab network which is air gapped from the outside world it wasn't much of a concern.

The above URL makes the server more efficient by not having to generate a new script for each system but instead only requires a change to one section of the URL, the build name, and it will work the same.

8.5 Server Automation

Server Automation in this project is highly important because this system had to be 100% stand alone and be able to function without intervention of the system administrator. The image below is what the Jenkins WebUIplatform looks like with the configured vulnerable OS builds.


Jenkins | 1 | search | Dean O'Neill | log out

Jenkins › ENABLE AUTO REFRESH

New Item
People
Build History
Manage Jenkins
My Views
Credentials
New View

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle
3 Idle
4 Idle
5 Idle
6 Idle

add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
●	☀	basicpentesting1	N/A	N/A	N/A	
●	☀	brainpan1	N/A	N/A	N/A	
●	☀	firstleaks13	N/A	N/A	N/A	
●	☀	imf	N/A	N/A	N/A	
●	☀	metasploit2	N/A	N/A	N/A	
●	☁	mrrobot	17 days - #33	17 days - #34	3 min 11 sec	
●	☀	pwnlabinit	N/A	N/A	N/A	
●	☀	sickok1	N/A	N/A	N/A	
●	☀	trollcavev12	N/A	N/A	N/A	
●	☀	vulnerabledockercontainement	N/A	N/A	N/A	
●	☀	vulnix	N/A	N/A	N/A	
●	☀	vulnvoip	N/A	N/A	N/A	
●	☀	win7	N/A	N/A	N/A	
●	☀	winxp	N/A	N/A	N/A	

Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds

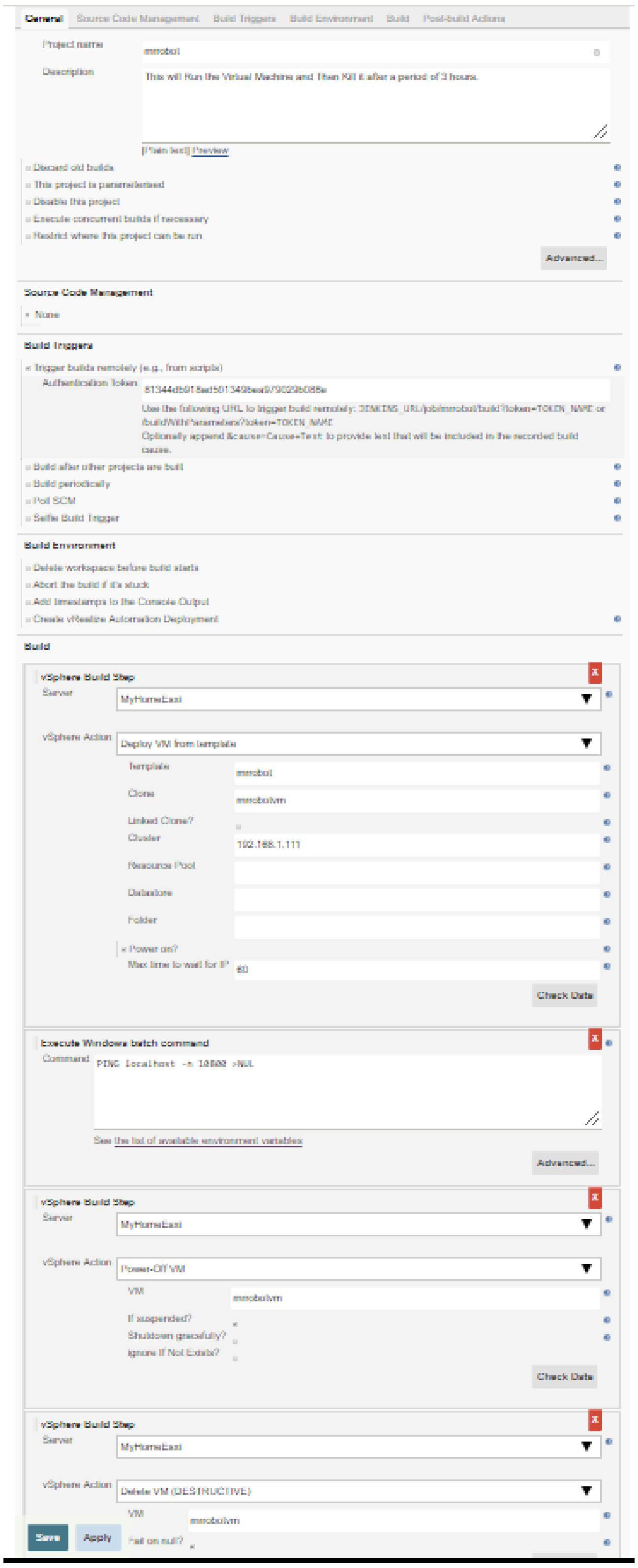
The build works in the following order

1. Website selection requests build on Jenkins via API token
2. Jenkins then Sends the command to build the stated template as an operating System
3. After build the Jenkins shell starts a ping that lasts for 3 hours
4. After 3 hours Jenkins shuts down the Virtual Machine
5. When the machine shuts down it is then deleted

Unfortunately there was no way to find an accurate way to put a timer on the server for the 3 hour delay, but I found sending pings to the local host and giving the output to NULL over the 3 hour period does not create strain on the server unless there is a large amount of Virtual Machines online.

The photo below is the whole processes listed above in JenkinsUI for building the MrRobot CTF virtual machine from a template, it will be similar for all other builds with only a simple name change, you can also see below the pinging script I have on the shell.

The following is how the Jenkins UI builds an operation system after being prompted. First the Jenkins UI receives the API token which corresponds to a job build. It then selects that image which is stored on the ESXI server as a template and builds the system automatically and also turns the system on. After the system is turned on Jenkins initiates a ping portion. This is done as you can see in the image below ping localhost -n 10000 > NULL , after this is complete, Jenkins power down the virtual machine and finally deletes the running VM access waiting for next command to build the next vulnerable system from a template



General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Project name: mrrobot

Description: This will Run the Virtual Machine and Then Kill it after a period of 3 hours.

[Plain text] Preview

- Discard old builds
- This project is parameterised
- Disable this project
- Execute concurrent builds if necessary
- Restrict where this project can be run

Advanced...

Source Code Management

- None

Build Triggers

- [✓] Trigger builds remotely (e.g., from scripts)

Authentication Token: 81344db918ad501349bea9790295088e

Use the following URL to trigger build remotely: JENKINS_URL/job/mrrobot/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME
Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

- Build after other projects are built
- Build periodically
- Poll SCM
- Selfie Build Trigger

Build Environment

- Delete workspace before build starts
- Abort the build if it's stuck
- Add timestamps to the Console Output
- Create vRealize Automation Deployment

Build

vSphere Build Step

Server	MyHomeEsxi
vSphere Action	Deploy VM from template
Template	mrrobot
Clone	mrrobotvm
Linked Clone?	
Cluster	192.168.1.111
Resource Pool	
Datastore	
Folder	
Power on?	✓
Max time to wait for IP	60

Check Data

Execute Windows batch command

Command:

```
PING localhost -n 10800 >NUL
```

See the list of available environment variables

Advanced...

vSphere Build Step

Server	MyHomeEsxi
vSphere Action	Power-Off VM
VM	mrrobotvm
If suspended?	✓
Shutdown gracefully?	
Ignore If Not Exists?	

Check Data

vSphere Build Step

Server	MyHomeEsxi
vSphere Action	Delete VM (DESTRUCTIVE)
VM	mrrobotvm
Fail on null?	✓

Save | Apply

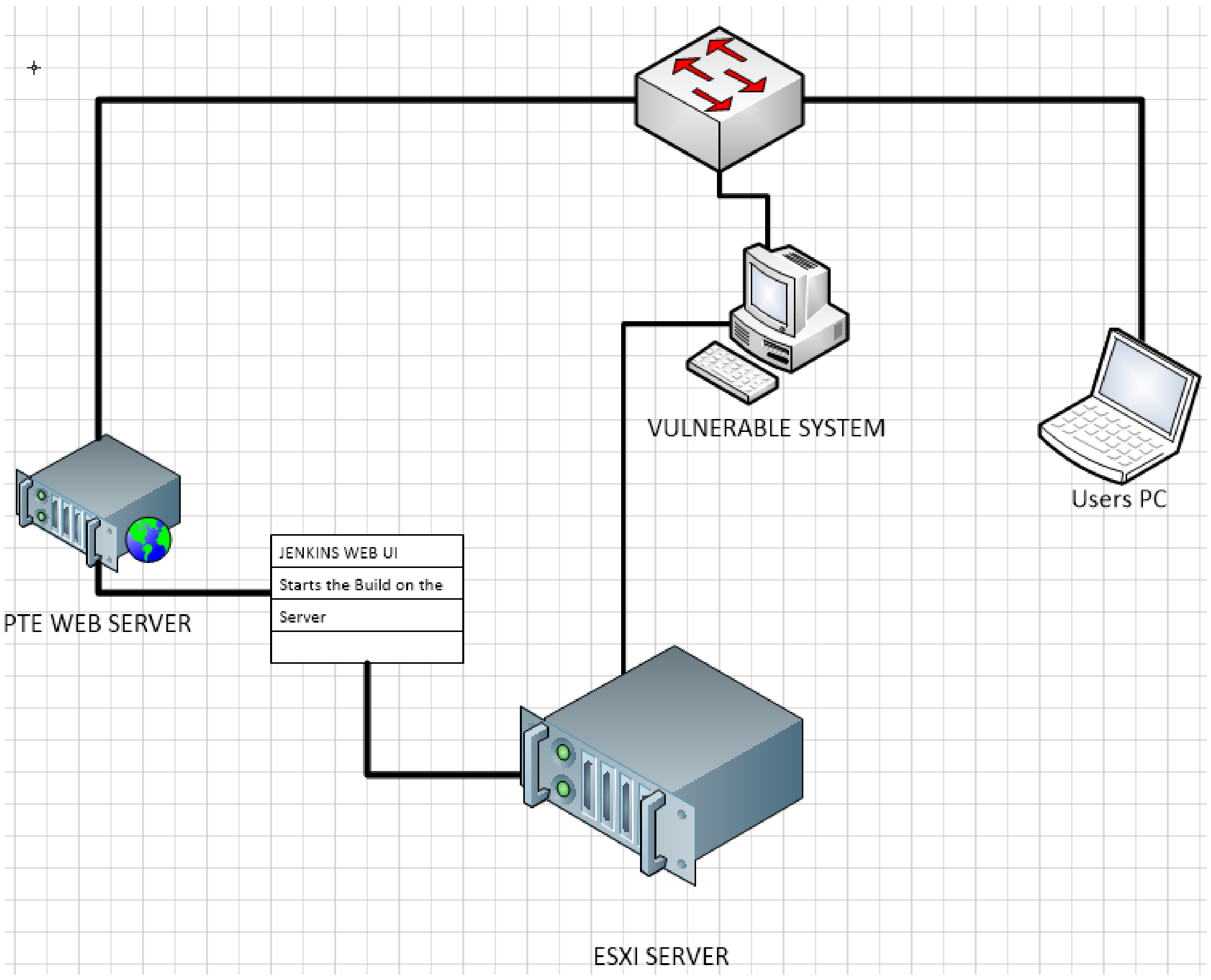
8.6 –Networking

Networking was a complicated process as the vulnerable operating systems we used where from VulnHu and were pre-configured with IP addresses. Having to individually modify each system's IP address could cause conflicts on build or even cause running services and processes to fail.

After having discussed this with the penetration testing team, as a group the conclusion was drawn that it be better to use the normal networking platform that is available as this would force the persons in training to use network scanning tools such as in nmap, Massscan or Netdiscover to find systems on the network, thus mimicking a realistic environment.

This took some strain off the building process and allowed valuable time for other configurations and settings. In the future one possible way would be to configure a DNS server on the network so that a URL for the specific machine redirects the user to the correct IP address without having to modify the virtual machine.

Below is the image of the current layer of the network structure, it shows connected to the switch the development network, which is then connected to the web server and the Jenkins UI. The Jenkins UI then controls the ESXI server to then build the vulnerable operating system which can be accessed through the same switch via IP the address.



9 – Automation of self-repair

Similar to the functions listed above with the building of VM's using the vSphere 6 plugin for Jenkins it will allow you to build systems form images; this is the bases for the automated self-repair system. This build requires two management servers a clone of each other, one will be used to test the other.

9.1 – Failure Testing

Using the similar functions of Jenkins shell option and a python module requests, we can use the shell to curl a hosted webpage and wait for a response that the webpage is still available. Using the requests to read and understand the response, if the response is 200 wait for a couple of minutes and test again.

If the system receives any other response prompt the user with a webpage splash screen stating there is an issue with the service and it will be back soon. The system will then wait 60 seconds then the non-damaged management system will send a request through the vSphere plugin to destroy the broken Image, the request a templated image to be built with a pre-set IP address and start the system.

With a configuration of the DNS, the URL is redirected to both IP addresses, if there is a failure of one system the other system will take over.

9.2 – Normal Workings

The idea of this function is to have both images constantly testing the opposite system, if one system goes offline the script will be called and will rebuild the opposite system, this is a continuous cycle of destroy and rebuild.

10-Testing and Evaluation

The testing and evaluation of this project was done by members of the red team, penetration team and network defence team, any bugs found were easily fixed due to the versatility the Jenkins webUI.

10.1 - Security Testing

The most important test was to make sure that the database file which housed the user logins was secure. This meant securing the platform against SQL injections, directory transversal attacks, buffer overflow attacks and brute forcing. During the rigorous testing play by members of the security team the only vulnerability discovered was the lack of an SSL certificate on the VMware server which was out of the control of the cyber security department that the project was designed for. This security concern was reported to the IT Department and is currently in the process of been repaired.

10.2 - Self-Sufficiency

The design and application of the server is intended to be self-sufficient so it was important to test the system for software and hardware crashes. Thanks to the excellent design of the vsphere application by VMware, they already have diagnosing features on the virtual machine that it can break it down and rebuild a new one from a template. This also included the web server, so if any issue was to arise the server would go offline and within minutes be back online fully functional as a new system.

10.3 - Load Testing

The biggest fear in designing this platform was having the system crash under a harsh amount of load from users. The server disk system was limited to the hardware resources available to it. After extensive testing it was found the system's hardware resources where used up with 48 active virtual machines running at once. When discussing this with the security team they agreed that this would not be a concern as a security team for a company is small and they wouldn't expect to have mass amounts of usage, other than for internal training purposes.

10.4 – Testing for Automation

The idea was to cause a failure of the system, by crashing the web server. Forcing a failure of the system ran the script as was expected and rebuilt the image as was intended by the opposite. This allowed for little down time of the service to the user.

11 -Conclusion and Further Work

From the original plan the design of the platform was modified to match the available resources. Due to hardware storage limitations the amount of operating systems that could be placed on the vsphere virtualisation server was infinitely greater than the available space.

Additionally the security team’s preference for a vulnerability training platform, as opposed to test patching, meant the use of vulnerable operating systems instead of working operating systems such as Windows Server 2012 or the IBM zOS mainframe operating system, which were originally intended for this project. However this preference also lightened the load on the development side of the platform.

In the future hardware can be obtained to increase the storage capacity on the virtualization server allowing more operating systems for development and security patch testing to be setup and configured. The beauty of this platform build is that as many virtual machines as needed can be initialised from a call on the webserver.

Since finalizing the platform and finishing this project, it has come to my attention that a company called Practical Pentest Labs or PPL for short, has assigned a similar system allowing penetration testers to practice and hone their skills for a nominal fee. However unlike my platform the PPL platform is limited to their designed systems, whereas with my project has the ability to allow the security team to add new vulnerable operating systems that with the aid of a system administrator and can be configured in about 5 minutes.

Thanks you for this opportunity and reading my paper.

11 - Appendix: Software Listings

abatchy (2017) *OSCP-like Vulnhub VMs*, Available at: https://www.abatchy.com/2017/02/oscp-like-vulnhub-vms.html(Accessed: 22nd January 2018).

Ar0xA (2015) *FristiLeaks: 1.3*, Available at: https://www.vulnhub.com/entry/fristileaks-13,133/ (Accessed: 22nd January 2018).

Claor (2016) *PwnLab: init*, Available at: https://www.vulnhub.com/entry/pwnlab-init,158/ (Accessed: 22nd January 2018).

D4rk (2015) *SickOs: 1.1*, Available at: https://www.vulnhub.com/entry/sickos-11,132/ (Accessed: 22nd January 2018).

Geckom (2016) *IMF: 1*, Available at: https://www.vulnhub.com/entry/imf-1,162/ (Accessed: 22nd January 2018).

Jason Swager& Eric Lordahl (2018) *Jenkins vSphere Plugin*, Available at: https://plugins.jenkins.io/vsphere-cloud (Accessed: 22nd January 2018).

Jenkins (2018) *Jenkins WebUI*, Available at: https://www.jenkins.io(Accessed: 22nd January 2018).

Josiah Pierce (2017) *Basic Pentesting: 1*, Available at: https://www.vulnhub.com/entry/basic-pentesting-1,216/ (Accessed: 22nd January 2018).

Leon Johnson (2016) *Mr Robot CTF*, Available at: https://www.vulnhub.com/entry/mr-robot-1,151/ (Accessed: 22nd January 2018).

Maleus (2014) *Tr0ll: 1*, Available at: https://www.vulnhub.com/entry/tr0ll-1,100/ (Accessed: 22nd January 2018).

NotSoSecure (2017) *Vulnerable Docker: 1*, Available at: https://www.vulnhub.com/entry/vulnerable-docker-1,208/(Accessed: 22nd January 2018).

Rapid 7 (2016) *Metasploitable 2 - Virtual Machine to Test Metasploit*, Available at: https://information.rapid7.com/metasploitable-download.html(Accessed: 22nd January 2018).

Reboot User (2012) *HackLAB: Vulnix*, Available at: https://www.vulnhub.com/entry/hacklab-vulnix,48/ (Accessed: 22nd January 2018).

Reboot User (2012) *HackLAB: VulnVoIP*, Available at: https://www.vulnhub.com/entry/hacklab-vulnvoip,40/ (Accessed: 22nd January 2018).

superkojiman (2013) *Brainpan: 1*, Available at: https://www.vulnhub.com/entry/brainpan-1,51/ (Accessed: 22nd January 2018).

VMware (2018) *VMware vSphere*, Available at: https://www.vmware.com/products/vsphere.html (Accessed: 22nd January 2018).

VMware (2018) *VMware*, Available at: https://www.vmware.com(Accessed: 22nd January 2018).

Warrior (2016) *Milnet: 1*, Available at: https://www.vulnhub.com/entry/milnet-1,148/ (Accessed: 22nd January 2018).

12 -Bibliography

FarkhodAlisherovA (2009) 'Methodology for Penetration Testing', *International Journal of of Grid and Distributed Computing*

Ang Yang (2006) 'Characterizing warfare in red teaming', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* (1083-4419), pp. 268 – 285

Walter M. Grayman (2006) Locating Monitors in Water Distribution Systems: Red Team–Blue Team Exercise', *Journal of Water Resources Planning and Management*, 134(4)

Redyouch. 2018. Reddit. [ONLINE] Available at: https://www.reddit.com/user/redyouch. [Accessed 1 March 2018]

ITB Logo. *ITB Logo*. Available at: https://www.tennisireland.ie/userfiles/Image/it blanchardstown.jpg [Accessed 1 March 2018]

VSP4 41 vCLIInst Script. (2010). 1st ed. VMware

13 -Plagiarism Declaration

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
DEPARTMENT OF INFORMATICS
HIGHER CERTIFICATE IN SCIENCE IN COMPUTING IN NETWORKING TECHNOLOGIES
LECTURER MICHEAL

DECLARATION ON PLAGIARISM

I declare that the work I/We am(are) submitting for assessment by the Institute examiner(s) is entirely my(our) own work, except where the author or source has been duly referenced and attributed.

I confirm that this material has not been previously submitted for a degree or any other qualification at XXXXXXXXX or any other institution. I further confirm that I have read and understood the Institute policy on plagiarism in assignments and examinations (3AS08.doc) and that I am not, so far as I am aware, in breach of any of these regulations.

Signed :Dean

Date :28/ 05/ 2018