



Cyber Range Management Framework for Cyber Warfare Training

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
*Bachelor of Science in Computing in Digital Forensics and Cyber Security***

**Supervisor:
XXXXXXXXXXXXXXXXXX**

20th May 2019

Plagiarism Declaration

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
DEPARTMENT OF INFORMATICS
BACHELOR OF SCIENCE IN DIGITAL FORENSICS AND CYBER SECURITY
LECTURER: xxxxxxxxxxxx

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 XXXXXXXXXXXXXXXXXXXX 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: 13/04/2019

Abstract

To begin with, this paper endeavours to explore and research the current landscape in the area of Cyber ranges, Capture the Flag frameworks, Cyber security training and Educational systems the purpose of which is to train students, employees and cyber actors to operate in the ways of cyber and knowledgably traverse and manipulate the complex digital eco-systems that exist today.

The paper then goes on to document the project's design, development and implementation of our 'Hostile' Web application software framework and our experimentation with its interactions and potential for interactions with simulated network environments for the purposes of users solving challenges, breaching virtual confines, breaking out of isolated sandbox environments and using their knowledge to exploit the vulnerabilities in the systems and access the areas only accessible through demonstrating their knowledge and understanding of the underlying technologies.

Our motivation in this endeavour is to provide an open source environment in which interested parties can run Capture the Flag like events but with an emphasis on training the users on more real-world networking environments representative of the network systems that exist.

Our system is modular, expandable, and extensible and as such we envision it evolving and encourage growth and expansion into a complex eco-system by way of developers and collaborators creative contributions.

Table of Contents

Plagiarism Declaration	2
Abstract.....	3
Table of Contents.....	4
Table of Figures.....	6
Glossary of Terms and Abbreviations	8
1: Introduction	11
1.1: Overview	11
1.2: Aim and Objective	12
1.3: Motivation.....	12
1.4: Background.....	12
2: Literature Review	14
2.1: History of Cyber Ranges	14
2.2: Literature review	15
2.3: Literature Review Summary.....	30
2.3.1: Overlay.....	30
2.3.2: Emulation.....	30
2.3.3: Simulation.....	30
2.3.4: Gamification	31
2.4: Technology Review	33
2.4.1: CTF Frameworks	33
2.4.2: Virtualisation Technologies	34
2.4.3: Financial Considerations.....	46
2.5: Market Research	50
2.6: Summary	50
3: Research Methodology	51
3.1: Research Questions.....	51
3.2: Rationale	51
4: Design, Development and Implementation.....	52
4.1 SDLC Models	52
4.1.1: Waterfall model.....	52
4.1.2: V-Model	53
4.1.3: Spiral model.....	54

4.1.4 Summary.....	55
4.2: Requirements	55
4.2.2: Functional requirements	56
4.2.3: Hardware and software prerequisites	56
4.3: Development Process.....	56
4.3.1: Environment	56
4.3.2: Version control	57
4.3.3: Progress tracking	57
4.4: Implementation.....	58
4.4.1: Web application.....	58
4.5: Networking.....	61
4.6: Server Instances	61
4.7: Setting up Naumachia	62
4.8: Scripting.....	64
4.9: Automating the installation process	64
4.10: Packaging and releasing	65
4.11: Summary	65
5: Testing.....	66
5.1: Setup	66
5.2: Testing	66
5.2.1: Testing SSH	66
5.3: Testing Challenges.....	68
5.3.1: Testing SQLI	70
6: Analysis of Results.....	72
7: Conclusion.....	73
7.1: Future Continuation	75
References	76

Table of Figures

Figure 1 - Collection of Documents for Review	16
Figure 2 - DARPA NCR Facility Layout from Slides	17
Figure 3 - DARPA NCR Facility Support Wireless and IoT Devices	18
Figure 4 - Photo from PH-Days Conference Cyber Range	19
Figure 5 - Virginia Cyber Range slide on simulations of light builds	20
Figure 6 - Screen Grab from Cyber Range Design	21
Figure 7 - Slide from Northrop Grumman on their Cyber Range	22
Figure 8 - Screenshot of slide of HNS Platform Design	23
Figure 9 - Screenshot of layout of the Network on Cyber Range	24
Figure 10 - A view of Quali's sandbox of their cyber range	25
Figure 11 - Red Team Cyber Range from Pentester Academy	26
Figure 12 - Hack The box landing Page	26
Figure 13 - NiteTeam4 working tree of a System	27
Figure 14 - NiteTeam4 Diagram of the Phone	28
Figure 15 - Full layout of a NiteTeam4 Cyber Range	29
Figure 16 – Design for VMware layout	35
Figure 17- Layout of Virtual Switching environment in VMware	36
Figure 18 – Image of Proxmox Network idea and design	37
Figure 19 - Screenshot of Networking functions in Proxmox	38
Figure 20 - Original Test Cyber Range that was built in Proxmox	39
Figure 21 - Layout of testing machines on our local Proxmox datacentre	40
Figure 22 - GNS3 layout of a working network and software	41
Figure 23 - GNS3 running on a virtual machine on Virtual Box	42
Figure 24 - GNS3 shell via SSH connection	43
Figure 25 - Image of working network in Naumachia	44
Figure 26 - Screenshot of the Pricing and availabilities of ClearVM website	45
Figure 27 - Amazon AWS Storage Pricing	46
Figure 28 - Digital Ocean VPS pricing	47
Figure 29 - Scaleway VPS Pricing	48
Figure 30 - Screenshot of Website that compares VPS pricing online	49
Figure 31 - Waterfall model diagram	53
Figure 32 - V-Model diagram	53
Figure 33 - Spiral Model Diagram	55
Figure 34 - use case diagram	58
Figure 35 - Paramiko Working Code	67
Figure 36 - Paramiko SFTP Code	67
Figure 37 - SSH key Recived	68
Figure 38 - Connection the server with the ssh key	69
Figure 39 - Connection Successful	69
Figure 40 - SSH Log	70
Figure 41 - Login Form for testing	70
Figure 42 - Attempt at SQLI	71

Figure 43 - SQLI Successful.....	71
----------------------------------	----

Glossary of Terms and Abbreviations

API - Application Programming Interface

ARM – Advanced RISC Machine

AWS – Amazon Web Services

Bare Metal - Server that contains no installed operating system.

Blue Team – Professional team dedicated to the defence of networks and security.

Commit - Change made to a Git repository.

CPU - Central Processing Unit

CR - Cyber Range

Cron - Timed task scheduler for UNIX based systems.

CTF - Capture the Flag - Collection of challenges that aim to test a person's knowledge and understanding of specific areas of cyber security.

Cyber Range – Realistic simulation of a network of systems, built for the purpose of training security departments such as Red Teams and Blue Teams.

CVE - Common Vulnerabilities and Exposures

DARPA - Defense Advanced Research Projects Agency

DHCP - Dynamic Host Configuration Protocol – Network service used to assign IP addresses and manages hosts on a network.

DoS - Denial of Service - Attack that stops a service by using all its available resources

DDos – Distributed Denial of Service simulate to the attack above but delivered by multiple slave systems.

DoD – US Department of Defense

DNS - Dynamic Name Service - Protocol used to translate FQDNs such as google.com into usable IP addresses.

EternalBlue – Windows SMB v1 exploit developed by the NSA leaked by the shadow brokers

ERD - Entity Relationship Diagram

ESXI - Software sold by VMware for the purpose of virtualisation on a bare metal server

Framework - Centralised collection of utilities working together towards a common goal.

FQDN - Fully Qualified Domain Name - domain name that specifies the exact location of a webpage

GitHub - Popular online web platform for hosting Git repositories

Hosted Server - a Hosted server is a server where the software is installed on top of a current running operating system

HTML - Hypertext Mark-up Language

IDS – Intrusion Detection System

IoT – Internet of Things

IPS – Intrusion Prevention System

Jinja2 - Template engine written in Python

NSA – National Security Agency

Proxmox - An open Source virtualisation server operating system

Python - Popular programming and scripting language.

Python module - Additional package to add functionality to Python

Pythonic - Widely acknowledged way of writing Python when carrying out specific actions.

Raspberry Pi - A raspberry pi is a small portable computer system designed using an arm processor, and is designed to be small portable and low power usage.

Red Team - Professional team dedicated to offensive security and finding vulnerabilities

Repository - Main directory of a project that is managed with a version control system

SDLC - Systems/Software Development Life Cycle

SSH - Secure Shell protocol

SQL – Structured Query language

SQLi - Structured Query language Injection

Ubuntu - Popular Linux operating system distribution

UI - User Interface

UX - User Experience

Virtual Box - Popular open source operating system virtualisation software

VLAN - Virtual LAN - commonly used for segmenting off a network for security and functionality.

VM - Virtual Machine

VMware - Virtualisation Company specialised in designing software for virtualising operating systems

VoIP – Voice over IP an internet protocol for vocal communications

VPS - Virtual Private Server

WAN – Wide Area Network

Wi-Fi - a communication device to allow for communications between systems over radio style communications where traditionally via cable

1: Introduction

1.1: Overview

This project aims to fill a niche in the realm of cyber security training. One of the most effective ways to teach people ethical hacking is by simulating a real-life scenario in a controlled environment, which could be physical or virtual, and allowing them to tackle obstacles and find weaknesses, just as they would when conducting penetration testing as part of a job.

A solution already exists to achieve that, and in fact, there are two different examples of how cyber security training can be taken to a whole new level: Capture The Flag — one of the most popular activities among those who wish to specialise in information security, its goal is to challenge people in different areas in the cyber security field while also enjoying a game-like experience. In most cases these events take place online, which in turn makes them very accessible to anyone at any time.

Cyber Range, a relatively new practice, it tries to recreate realistic setups and challenges people to exploit weaknesses as if it were not a simulation, which might sound slightly similar to Capture The Flag, but in many cases it is a much more accurate representation of the real world.

Many different implementations already exist for Capture the Flag training, whereas Cyber Ranges are not quite as popular due to their nature of being complicated to set up and being open to attacks from anyone who could potentially cause vulnerabilities to escape their isolated environment and damage the underlying systems.

With this project, the team wanted to be able to breathe some life into Cyber Range training by building a platform or framework that lets people run events involving one or multiple Cyber Ranges. This could be implemented in the same way as capture the Flag platforms to make it more accessible.<sup>[P]
[SEP]</sup>

1.2: Aim and Objective

This project aims to fill a niche in the realm of cyber security training. One of the most effective ways to teach people ethical hacking is by simulating a real-life scenario in a controlled environment, which could be physical or virtual, and allowing them to tackle obstacles and find weaknesses, just as they would when conducting penetration testing as part of a job.

A solution already exists to achieve that, and in fact, there are two different examples of how cyber security training can be taken to a whole new level: Capture The Flag — one of the most popular activities among those who wish to specialise in information security, its goal is to challenge people in different areas in the cyber security field while also enjoying a game-like experience. In most cases these events take place online, which in turn makes them very accessible to anyone at any time.

Cyber Range — a relatively new practice, it tries to recreate realistic setups and challenges people to exploit weaknesses as if it were not a simulation, which might sound slightly similar to Capture The Flag, but in many cases it is a much more accurate representation of the real world.

Many different implementations already exist for Capture the Flag training, whereas Cyber Ranges are not quite as popular due to their nature of being complicated to set up and being open to attacks from anyone who could potentially cause vulnerabilities to escape their isolated environment and damage the underlying systems.

Taking inspiration from a past project of a similar genre (Thomas Gibbons, 2017), The team wanted to create a web-based front-end interface that would allow users to set up virtual networks with simulations of vulnerable devices and software, which could be used as part of Red Team (Rouse, 2017) security assessments.

1.3: Motivation

From personal experience, as regular participants in CTFs and other related competitive activities, through research the team has noticed a shortage of open source technologies that specialise in cyber ranges and simulation training. There was a need to build a platform that can design networks and inherits the features of a CTF framework, maintaining a minimal approach while not giving up important features like modularity and scalability.

For these reasons, this project is a crucial step towards an open source solution to the scarcity of such projects on the web. The team also strongly stand by the notion of FOSS (Techopedia). As such, the team will publish all the complete work to a public git repository for review and scrutiny by other developers, in hopes of not only improving the software itself, but also stimulating people's interest in these kinds of training methods.

1.4: Background

The cyber security market has grown considerably in the last decade (Shields, 2014), with more and more people gaining interest and training to become the front line in modern

cyber defence forces. Military organisations have always strived to be at the cutting edge of cyber warfare training.

With the development and understanding of Cyber Ranges the US Military and many other organisations have been funding development and research heavily in the past years, with the most recent development being by DARPA “Defence Advanced Research Projects Agency” this project is called “National Cyber Range” a huge development to build a scale version of the Internet for cyber war gaming (BBC Technology, 2011)

2: Literature Review

When researching this project, there was one major issue in the researching process, that issue was finding material that is citable and understandable, when researching Cyber Ranges and Cyber Range controllers, the team was in many cases only able find presentations, brochures and conference talks on YouTube, what was found via searching on services such as Google scholar where unfortunately in many cases belonging to countries that would require translations or the document themselves were behind pay walls meaning a membership fee is required to view them. Unfortunately, a lot of this kind of material is un-citable.

2.1: History of Cyber Ranges

Cyber warfare is a term that is heard significantly more often than ever in recent years. Any term that contains the word 'war' is loaded and should probably be used with careful consideration and not thrown about lightly. However, governments and politicians refer to cyber war and cyber operations more frequently. Cybercrime is another term that has become more relevant and often heard as the nature of crime and how it is committed has changed along with the evolution of the internet and computer systems. Cyber-espionage could be considered as well as the nature of how corporations and businesses operate changes along with everything else moving online.

Massive state sponsored cyber operations like Stuxnet and Flame are becoming the way that governments and regimes attack each other and spy on each other. Since the leaks by Edward Snowden the public now know a lot more about the level to which the NSA goes to infiltrate global systems and spy on other countries as well as its own citizens. China is believed to have on-going sophisticated state sponsored campaigns to infiltrate the political and economic organisations of the west and both gather valuable intelligence and steal intellectual property from corporations.

Viruses, Trojans, worms, malware, ransomware and botnets are all threats to the stability of the global systems that businesses and governments depend on more and more for smooth operation. Individuals also have become more dependent on their laptops, smartphones and devices. People use online banking and e-commerce is a major factor in most people's purchases. With the advent of IoT (Internet of Things) the future scenarios one can envisage are all of us being even more vulnerable. ID theft, fraud, online scams, malvertising, election tampering are all enabled and enhanced by the internet.

Cyber-Terrorism is another potential threat with groups like ISIS using social media platforms to radicalise and recruit new members.

"Our daily life, economic vitality, and national security depend on a stable, safe, and resilient cyberspace. Cyberspace and its underlying infrastructure are vulnerable to a wide range of risk stemming from both physical and cyber threats and hazards. Sophisticated cyber actors and nation-states exploit vulnerabilities to steal information and money and are developing capabilities to disrupt, destroy, or threaten the delivery of essential services." (HomelandSecurity, 2018)

It is for all these reasons and more that 'Cyber' has for some time been considered by the military to be the fifth domain of warfare along with 'Land', 'Sea', 'Air' and 'Space'. The American military in considering future threats to the stability of their country and its businesses realised that there is just as much need (if not more) for training personnel to operate both offensively and defensively in this fifth domain, and in order to train and develop the skills needed for this they began to create training infrastructures known as 'Cyber Ranges'.

In recent years there has been a huge increase in the number of cyber ranges being built by cyber security companies in co-operation with military and law enforcement. Most of these are in America however the European Union, Australia and Israel also have official programs in action to train up personnel in the cyber skills needed for the current and future threat landscape.

As research for this project each member of the team will look at the different cyber ranges that are in development and currently being used for training. It would be useful to examine the different types of cyber ranges such as simulation systems and emulation systems, are they hardware based or software based, virtual or a large-scale infrastructure encompassing a mixture of all elements. The research conducted by the team will examine the advantages and disadvantages of these different systems.

2.2: Literature review

researching for past projects online, the team came across a mass list of past project PDF and presentation documents that could be found, the first thing to do was make a collection of these on the GitHub repository so others who are interested in researching the projects can themselves find the content that will help their research and will allow them to check the projects GitHub. This also benefited the team as this material would be used for documentation and research.
























 2017-FIE-lessons-learned-exercises-cyber-range-presentation.pdf	Spell-check	7 days ago
 2017-ICSOF-kypo-cyber-range-design-presentation.pdf	Spell-check	7 days ago
 ADA62747711.pdf	Spell-check	7 days ago
 Baltimore-Cyber-Range-07-25-2018.pdf	Spell-check	7 days ago
 CloudShell-Cyber-Range-Orchestration.pdf	Spell-check	7 days ago
 Cyber-Range-Build-vs.-Buy-White-Paper.pdf	Spell-check	7 days ago
 Cyber-Range-Buyers-Guide-for-Higher-Education.pdf	Spell-check	7 days ago
 Cyber-Range-brochure.pdf	Spell-check	7 days ago
 Documentation of the Standoff Network Setup - RULES AND SETUP.pdf	Spell-check	7 days ago
 Documentation of the Standoff Network Setup.pdf	Spell-check	7 days ago
 JYVSECTEC-cyber-range.pdf	Spell-check	7 days ago
 NG MS Federated-Cyber-Range DS.pdf	Spell-check	7 days ago
 PHDays Cyber Range.jpg	Spell-check	7 days ago
 PHDays Cyber Range2.jpg	Spell-check	7 days ago
 Tender-Briefing-Cyber-Range.pdf	Spell-check	7 days ago
 Virginia+Cyber+Range+David+Raymond.pptx	Spell-check	7 days ago
 asf-cyber-range-large.pdf	Spell-check	7 days ago
 br-cyber-range-training-services.pdf	Spell-check	7 days ago
 cyber-range-150630140941-lva1-app6891.pdf	Spell-check	7 days ago
 cyber-range-brochure-new-outlined-ilovepdf-compressed.pdf	Spell-check	7 days ago
 cyber-range.pdf	Spell-check	7 days ago
 cyber-range.pptx	Spell-check	7 days ago
 cyber_acre_2018.pdf	Spell-check	7 days ago
 standoff_phd8_1.jpg	Spell-check	7 days ago

Figure 1 - Collection of Documents for Review

Above you can see documentation of past projects or products that are about Cyber Ranges or contain information related to Cyber Ranges. From reading all of the documentation the team found on Cyber Ranges that are listed above the team decided to instead of writing about all of them, to pick the top few and write about the top past documentation found, these include documentation that gives us the best reference points and information to further research.

National Cyber Range

While researching cyber ranges with governmental offices, an unclassified presentation was found wrote by Jinendra Ranka of DARPA that unfortunately doesn't have much in the way of documentation but what it does have is tones imagery that is well commented for example a picture of the layout of the actual cyber range offices, that can be seen below

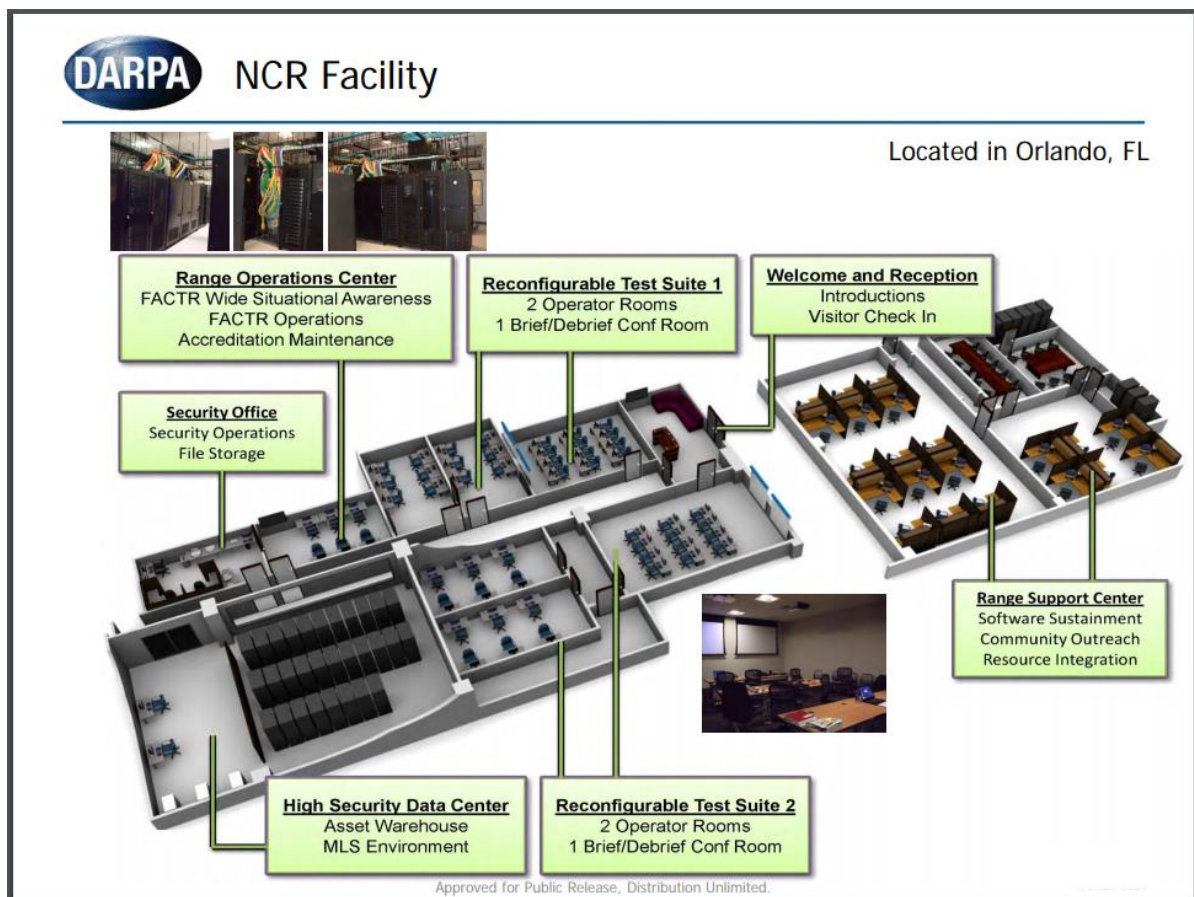


Figure 2 - DARPA NCR Facility Layout from Slides

This as you can see is a huge facility located in Orlando FL with a large datacentre, with multiple server rooms, training rooms, meeting rooms, class rooms and briefing rooms. This location is the housing of the National Cyber Range; this is a large US Government incentive to develop a scale replica of the Internet to allow for training of cyber warriors and cyber security personnel. From more in-depth research into the NCR, more unclassified and released documentation was found, to best describe the cyber is to use a quote from a DARPA memo and factsheet for US President Obama during his time in the Whitehouse around 2009, "DARPA is creating the National Cyber Range to protect and defend the nation's critical information systems. Leveraging DARPA's history of cutting-edge research, the NCR will revolutionize the state of the art for large-scale cyber testing. The NCR will provide fully automated range management and test management suites to test and validate leap-ahead cyber research technologies and systems and provide vision for iterative and new research directions." (DARPA, The National Cyber Range, 2016)

That quote best describes the goals that DARPA has for their cyber range, in a later document also released and declassified talks about the ability for them to add Wireless devices on to the range as they have a secure facility for testing these devices on the cyber range. This can be seen in the bellow slide



Facility Overview: Support for Wireless Testing



- **Wireless environment that supports classified testing (TS/SCI)**
- **Support for mobile computing: iOS, Android, Windows 8 on tablets, cell phones, and multimedia devices**

Distribution Statement A – Cleared for Open Publication by OSD on February 24, 2015 SB Case Number 15-S-0994

2014 Lockheed Martin Corporation. All Rights Reserved. 13

Figure 3 - DARPA NCR Facility Support Wireless and IoT Devices

Later on in the document another quote can be describe another purpose of use for the cyber range “Requirements to test advanced cyberspace tactics, techniques, and procedures that require isolated environments of complex networked systems (e.g., movement on the Internet)” (DARPA, National Cyber Range Overview, 2015) this single quotes demonstrate that the development of the cyber range is also being used for testing of offensive tactics that could have been for example the Eternalblue exploit and other systems. These are some grate use case examples of why a cyber-range is relevant in today's Cyber Space. This network was also demonstrated on a public TV show documentary on the history channel, that was based on Cyber Warriors in 2010, called “Cyber Defence: Military Training for Cyber Warfare” where groups such as the NSA, US Navy, US Army and a civilian based contracting group competed on the network for the top prize.

PHdays Conference in Moscow Russia

PHdays built a CTF “Capture the Flag” event, in which the team of security professionals developed multiple challenges, built a physical miniature scale city that contained multiple challenges and locations such power plants, military stations and homes. They were designed in such a way that an attacker could exploit the entirety of the city trough post exploitation and pivoting through multiple entries such as for example exploiting a home user who has administration control over a router at a local ISP “internet service provider” that then leads to exploitation of the local power plant trough IOT exploitation. Below is a photo of the city full active at the conference and being exploited by attackers at the

conference, the CTF “capture the flag” itself was called Standoff with the idea of having a standoff between both blue team and red team members.

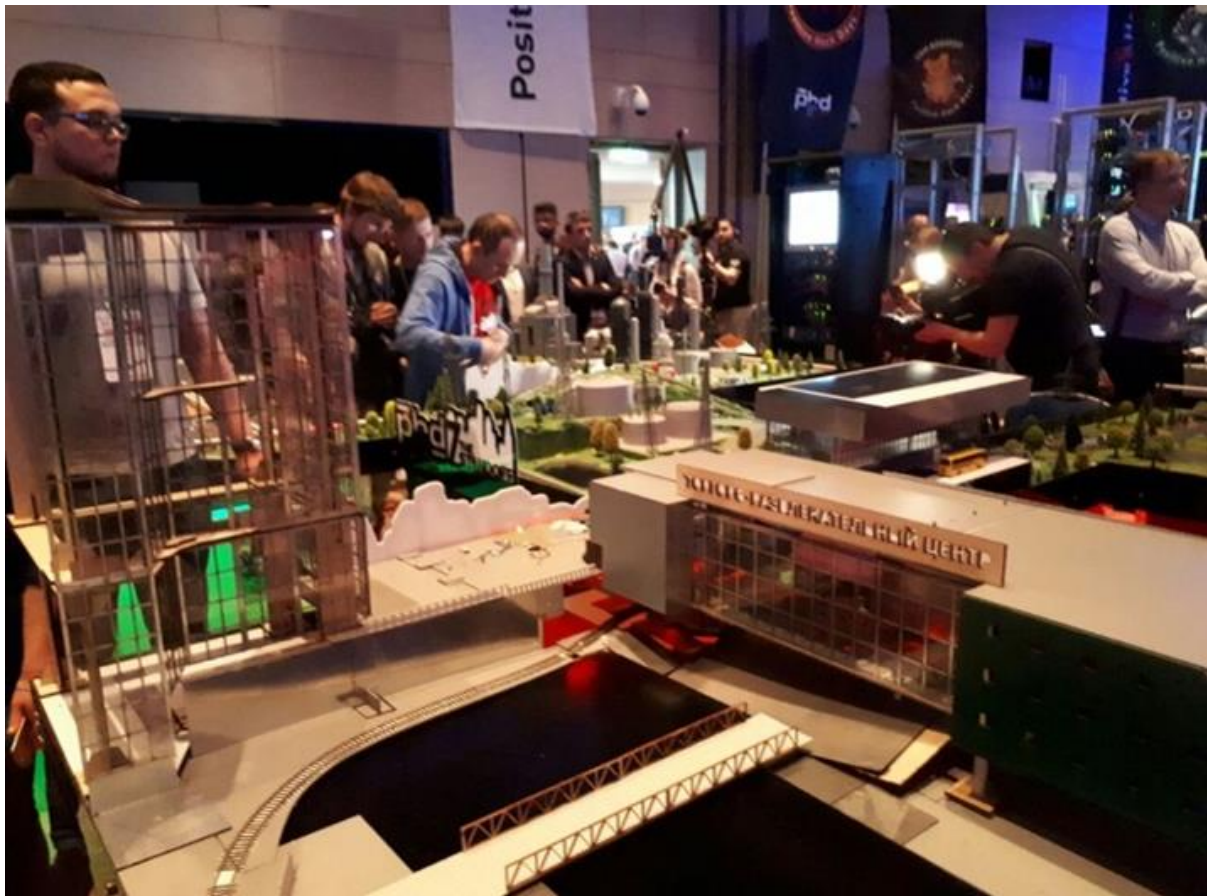


Figure 4 - Photo from PH-Days Conference Cyber Range

Virginia Cyber Range by David Raymond

This was a proposed to the governor of Virginia under the biennial budget; the project was part of the cyber security initiative to increase both cyber security inspirations and awareness trough development of a cyber-range that would take into account Virginia based services such as power stations and police stations as well as IOT devices such as light bulbs and radios. Unfortunately, after further research, it was found that the project was never taken on board and was not developed any further. The figure below represents a slide from the presentation that was used to demonstrate a possible IOT based simulation that attackers could interact with.

Our Simulation (1)

- Virtual machines are used to replicate this scenario
 - One VM emulates lighting system (wireless bridge and bulbs)
 - Another VM simulates an authorized user of the system

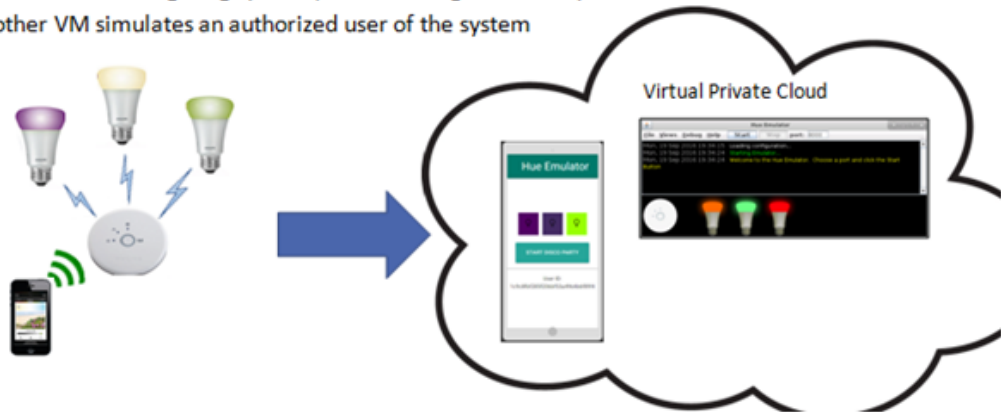


Figure 5 - Virginia Cyber Range slide on simulations of light builds

The above image is a demonstration of how the device would replicate IOT light bulbs for simulation and attack by the user.

Tender Briefing Cyber Range

This document was a leaked document related to a company called MCMC where the topic was the planning stages of building and configuring a full cyber range for simulating offensive attacks against platforms such as traffic lights systems and IOT devices. The platform design was nothing special but what it had was a idea of a time frame for setting up a physical cyber range and the costs involved in setting up, even down to the floor plan for the space where the cyber range training will be held. Bellow I a picture of the slide where the developer talks about how the Cyber Range network will look logically and how it will function.

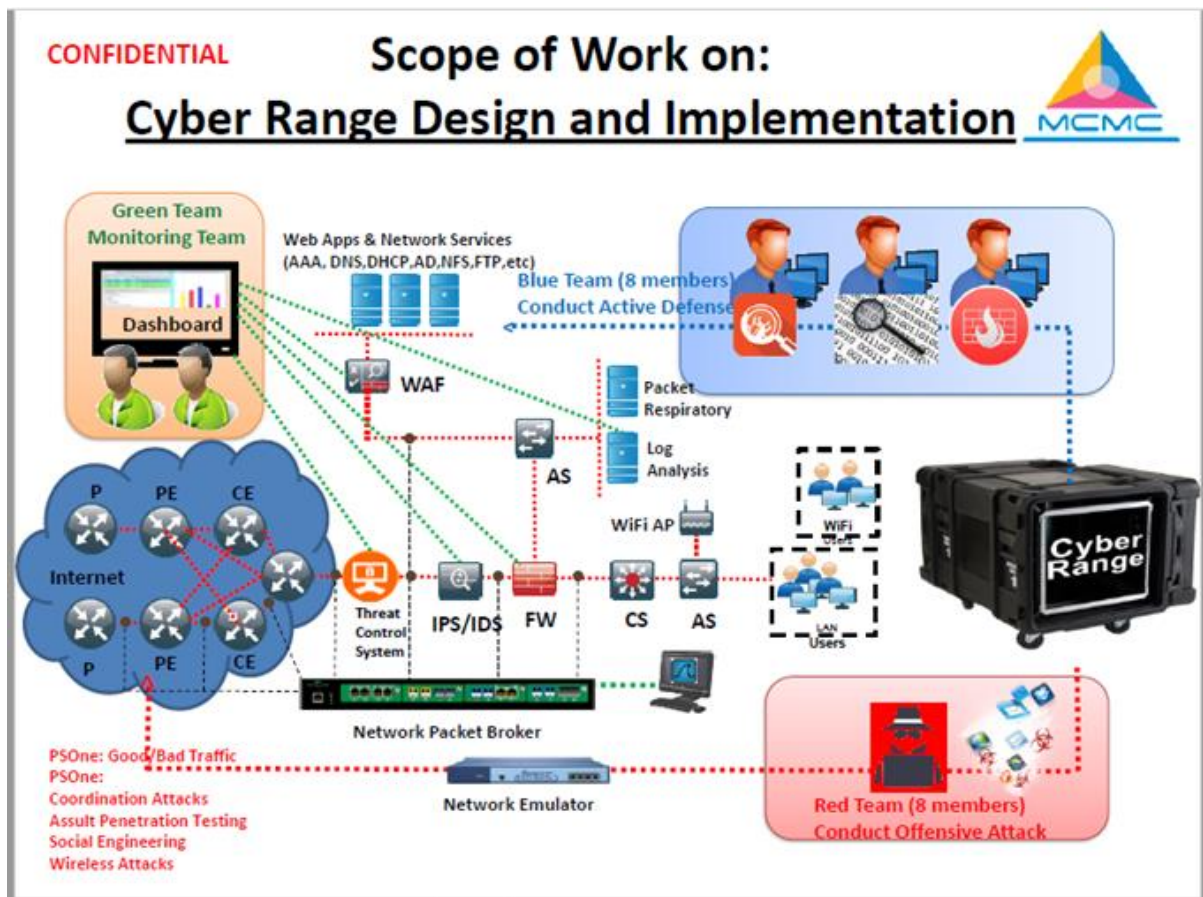


Figure 6 - Screen Grab from Cyber Range Design

NG MS Federated Cyber Range DS

This document is a brochure for services provided by Northrop Grumman; they provide a service that is separate from the conventional idea of a cyber-range. The brochure itself is not very extensive with its content, but it does explain their interpretation of a cyber-range, in which they would configure a platform for both Blue Team and Red Team members to test the application to see if there are any vulnerabilities.

Although this is not exactly related to the current project, is a good reference of other possible uses of a cyber-range.

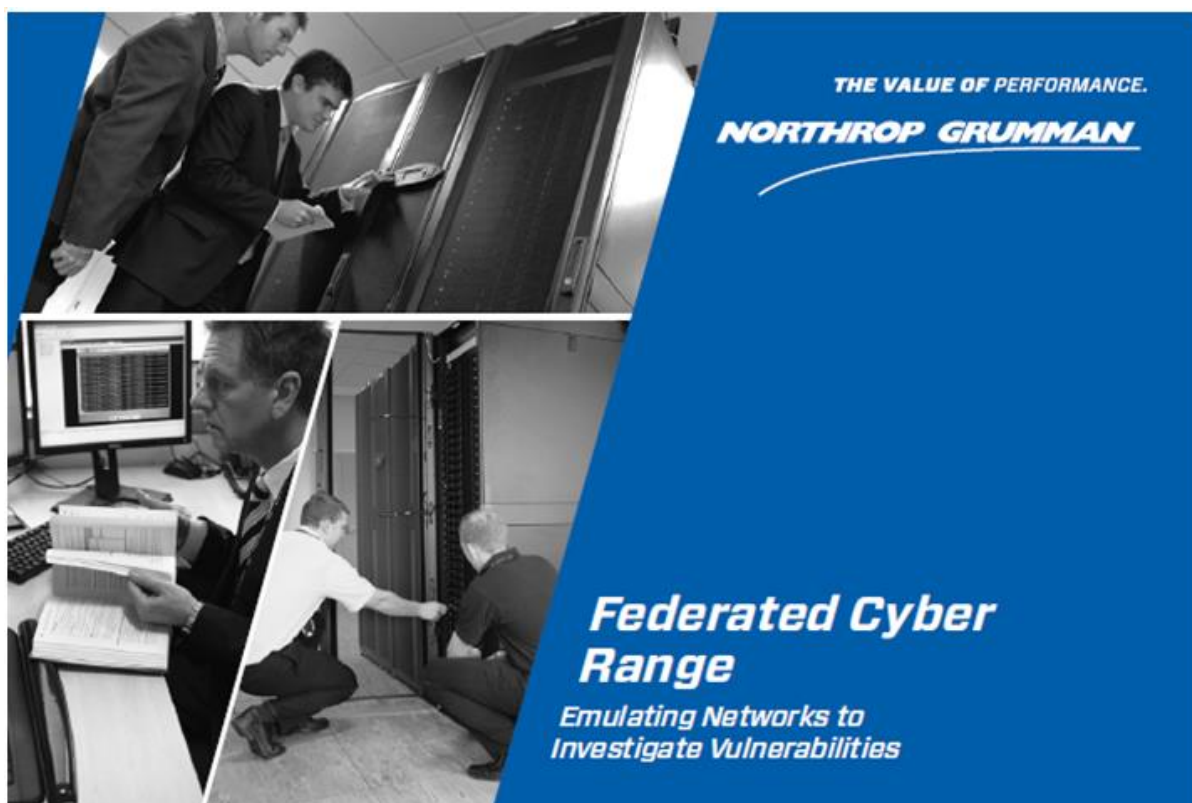


Figure 7 - Slide from Northrop Grumman on their Cyber Range

HNS Platform

This recently came across are pages as its being promoted as an advertisement on many social media sites, nearly all the material is in French unfortunately so following the information is hard. But what can be found from the imagery and the videos they have hosted on YouTube, shows that is a platform developed by the French Military for training cyber warriors, and Capture the flag events. This system is works very similar to a normal Cyber Range but their platform allows for more jeopardy style challenges to be used, which extended out past the original challenge. For example, a simple web application that has an SQL injection, one you have access to the webpage you then can go on an exploit other system such as the webserver, DNS and so on. This platform also has the capabilities to host defensive challenges that opposing teams can attack and your team needs to be able to find the exploit and patch it. This platform seems interesting but unfortunately due to language restrictions, the team was unable to find more information to extend is knowledge of this platform.

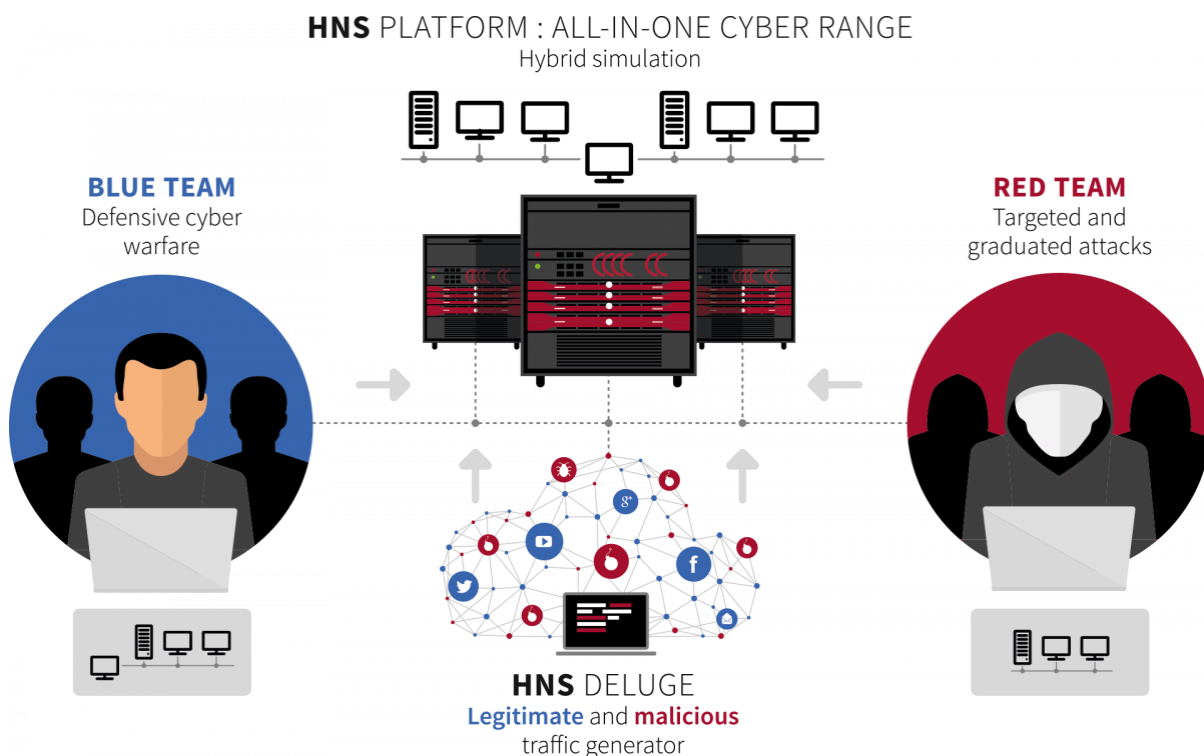


Figure 8 - Screenshot of slide of HNS Platform Design

Jan Vykopal - KYPO Cyber Range: Design and Use Cases (2017)

In this paper Jan Vykopal, a Masaryk University student, describes the development process of his team's newly proposed cyber range called KYPO, which intends to provide complex cyber systems and networks in a virtualised and monitored environment. Jan notes that a survey was conducted by the Australian Department of Defence to showcase publicly available state of the art cyber ranges (Davis and Magrath, 2013). Many other cyber ranges likely exist that are funded by governments and militaries, so the survey is somewhat limited in scope.

In one of the sections, Jan covers several online resources and test beds that aid in the creation of realistic, network based cyber security training frameworks. The research deals with recent advances and innovations through a comprehensive literature review conducted from 2013 to 2017.

A considerable portion of the paper revolves around the logistics of a cyber-range design. From monitoring, to administrator management, to how the data is managed in the back end and front end, as well as data visualisation, these are all very important considerations when working on a practical project like the one described.

The KYPO cyber range gives an impression of a sophisticated framework built for extensibility and modularity, which is why it is a great project to take inspiration from in order to build the project simple and effective platform.

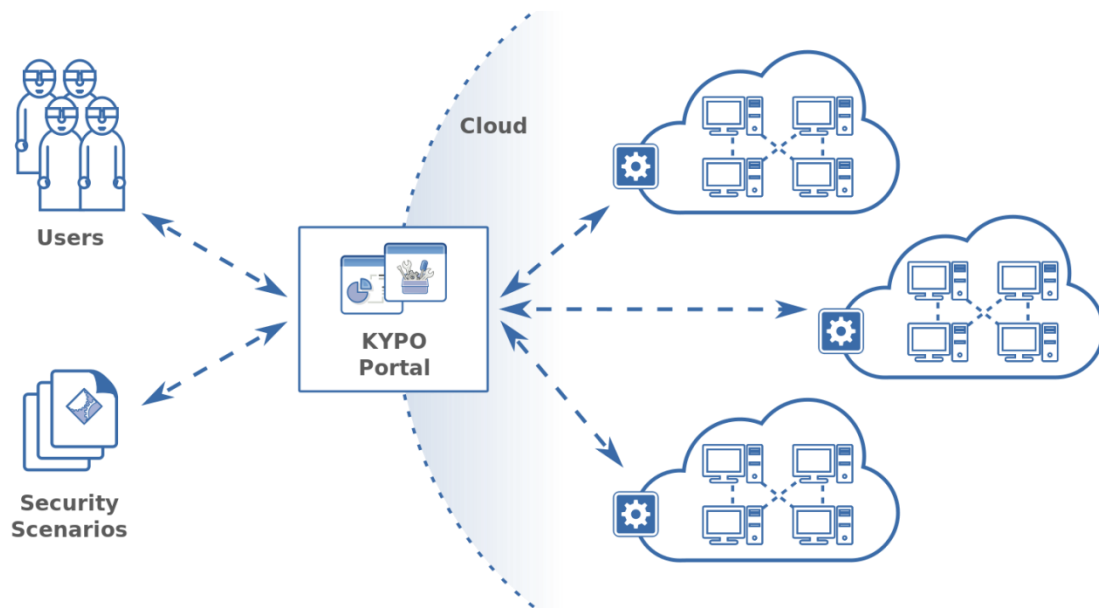


Figure 9 - Screenshot of layout of the Network on Cyber Range

Quali - Cloud Sandbox for Cyber Range Training

At the RSA Conference in 2017, two organisations partnered together to create a project that could provide a full-stack cloud based cyber range on the fly. The goal was to simplify the user experience and to automate as much of the process as possible, leading to a much lighter and more portable framework.

The first company is called Quali, which specialises in cloud-based computing involving intelligent automation and orchestration. Their Cloud Shell service provides on-demand sandbox environments, exactly what a cyber-range would need in order to be portable.

The other company is Ixia, a provider of security, testing and visibility solutions for applications across physical and virtual networks (IXIA, IXIA, 2018) Their industry leading security testing platform Breaking Point (IXIA, BreakingPoint, 2018) allows both regular and malicious traffic to be generated for the purpose of testing networks. This can range from stress testing, such as Denial of Service (DoS) attacks, to launching exploits against critical services.

The collaboration between those two organisations is a great fit for cyber range creation, as they each deliver a unique and necessary aspect in order to make the software more functional and flexible. (Joly, 2017)

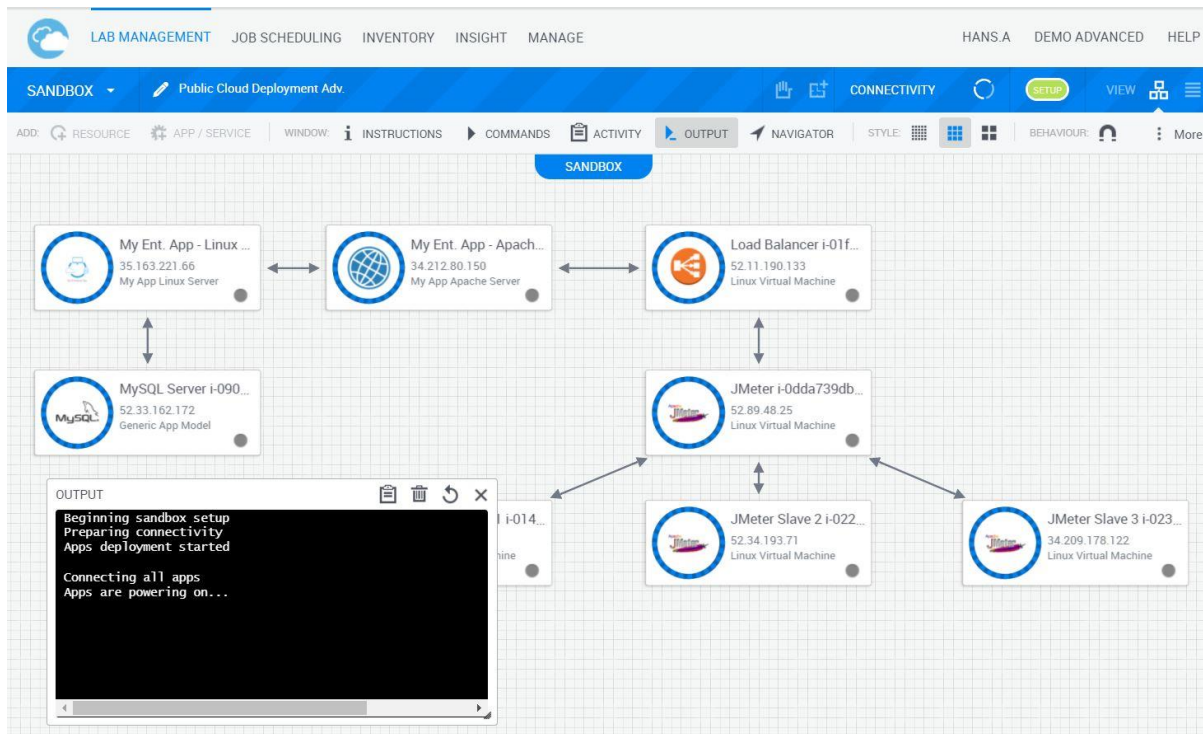


Figure 10 - A view of Quali's sandbox of their cyber range

Public Attack Defence

The Public attack defence network is a system designed by Pentester Academy is an online training environment designed to replicate services of a cyber-range as need and currently hosts over 800 challenges that allow at will request of a vulnerable system such as an IOT device with unsecured login. At request this will start the service for testing by the user, the huge defence between this platform and others is the ability to have such a huge variety of challenges and also its unique system of instead of using a VPN file to connect to the system it gives each user a CLI connection to a kali Linux operating system through a the web application, Making it easier for non-skilled users to work on systems immediately. This platform allows for both offensive and defensive training through multiple software and training utilities.

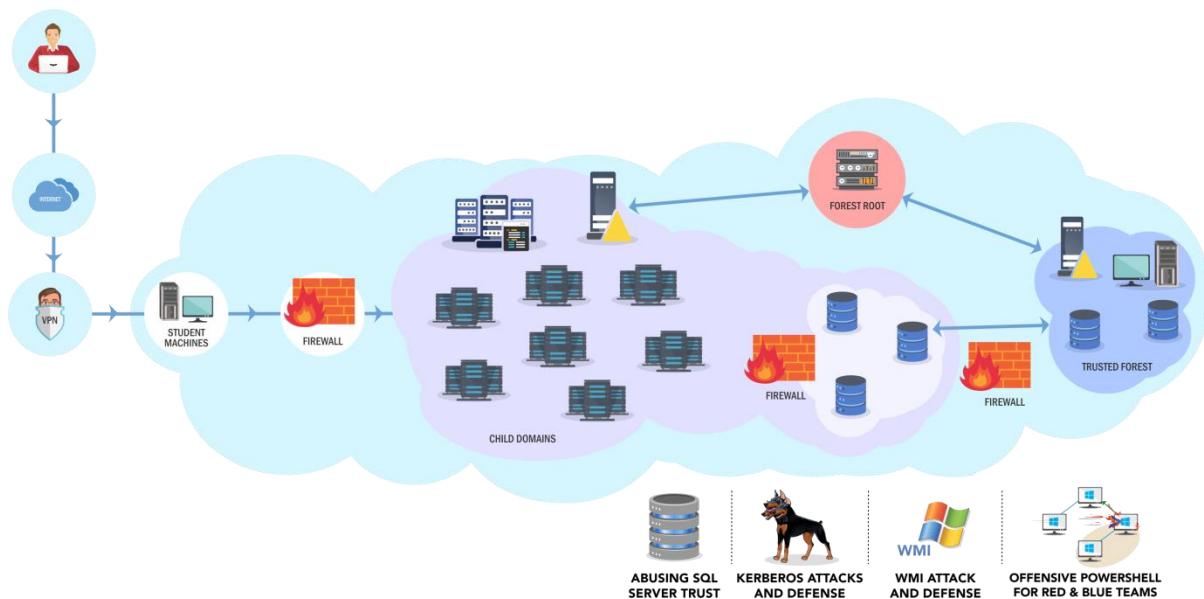


Figure 11 - Red Team Cyber Range from Pentester Academy

Hack the Box

This is a platform designed for learning offensive security mainly, it works by having a set number of virtual machines online with pre-set vulnerabilities that multiple users can work on to exploit; connection to this box is through your own Kali Linux operating system with an open VPN configuration file. This process takes knowledge in understanding the configuration and setting up on an open VPN file, on the system and getting it to connect. The major downside of Hack the Box is the fact that its community based, since its community based there is an issue with people knocking each other off each other's ports, braking services or restarting the box on each other by accident or even on purpose

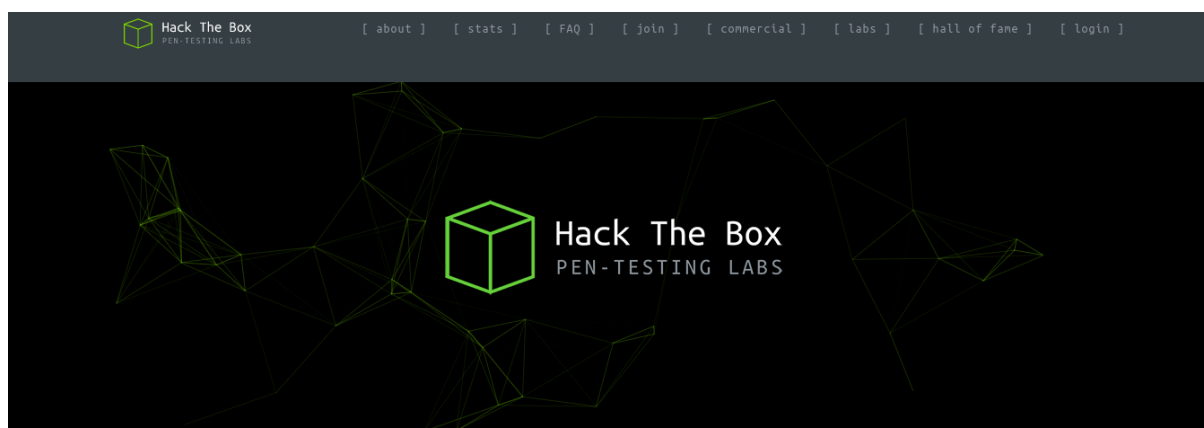


Figure 12 - Hack The box landing Page

Nite Team 4

While researching the functionality and gamification of cyber ranges, the team came across a video game on Steam a gaming platform called "Nite team 4", the game is a Military

Hacking Simulator the game itself has no real bearing on real world hacking but what it does is knowledge and high detailed simulated tools used for simulating the attacks.

within this game for an extra price of €50 there is a feature called network administrator, that allows members of the game who buy the extra content to build a full functioning simulated cyber range that other users of the game to attack with the games mechanism's, this allows for features such as Windows Device, Linux Devices, Mobile devices and tons more using leaked exploits of many government's agencies.

Below is an image of a simple network breakdown in the administrator mode that shows the host `ctf_server.hackersoc.hvm` and shows the services running and the exploits that can be run on the services.

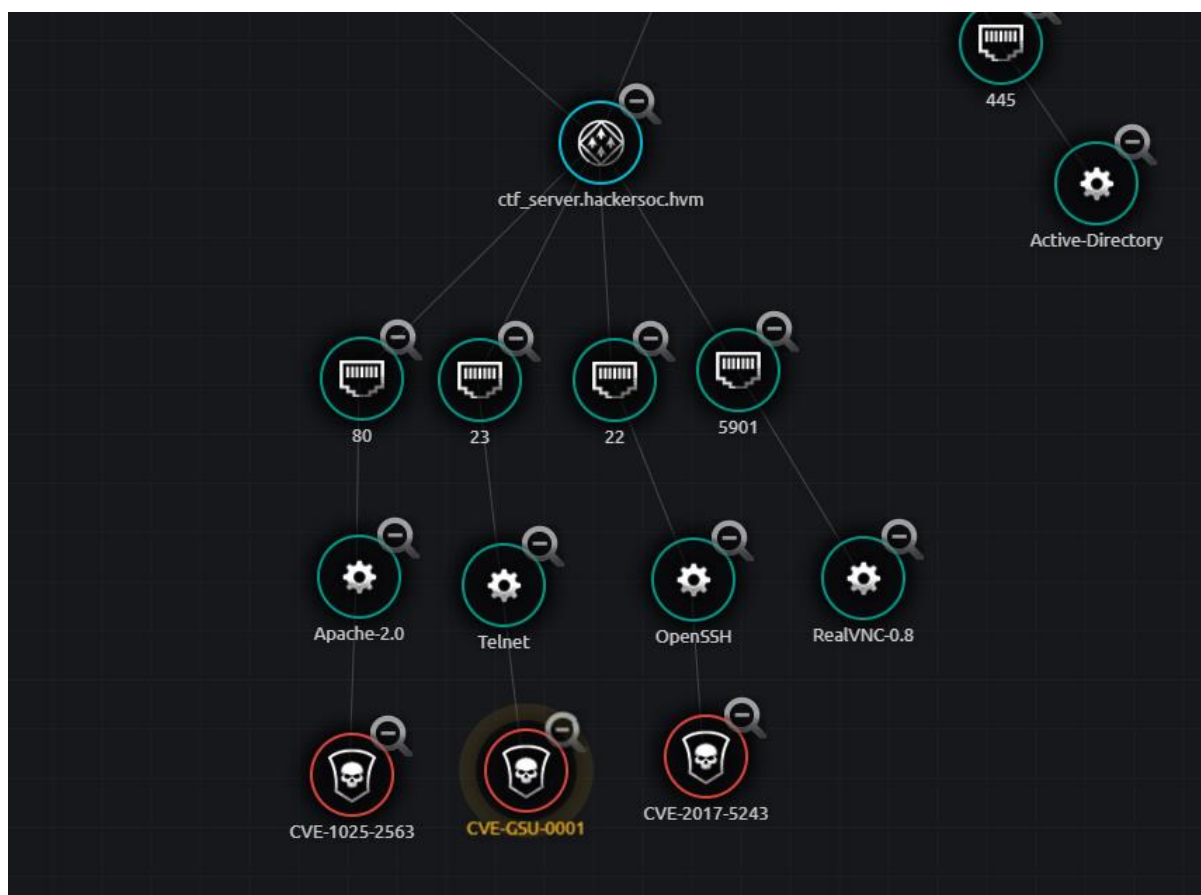


Figure 13 - NiteTeam4 working tree of a System

The Team Managed to get access to the advanced features called "Network Administrator" after contacting the development team on Discord, they offered a discounted copy of the game and network administrator downloadable content. This feature allows the "administrator to build a very detailed network with infinite numbers of Public facing domains, private domains all branching off a single Public host name. These domains can branch off into more detailed areas such as individual hosts, active directory, network attached storage units, phones such as iPhone, Android, Blackberry and multiple other mobile devices, these can branch off to attacks that would include exploiting the camera, microphone, private SMS message and even the ability to exploit the wireless card on the phone to see what public access points are around the device, for further exploitation.

Below is simple diagram of how the phone system works in administrator mode, here you can see all the services and features that can be exploited by the attacker as discussed above

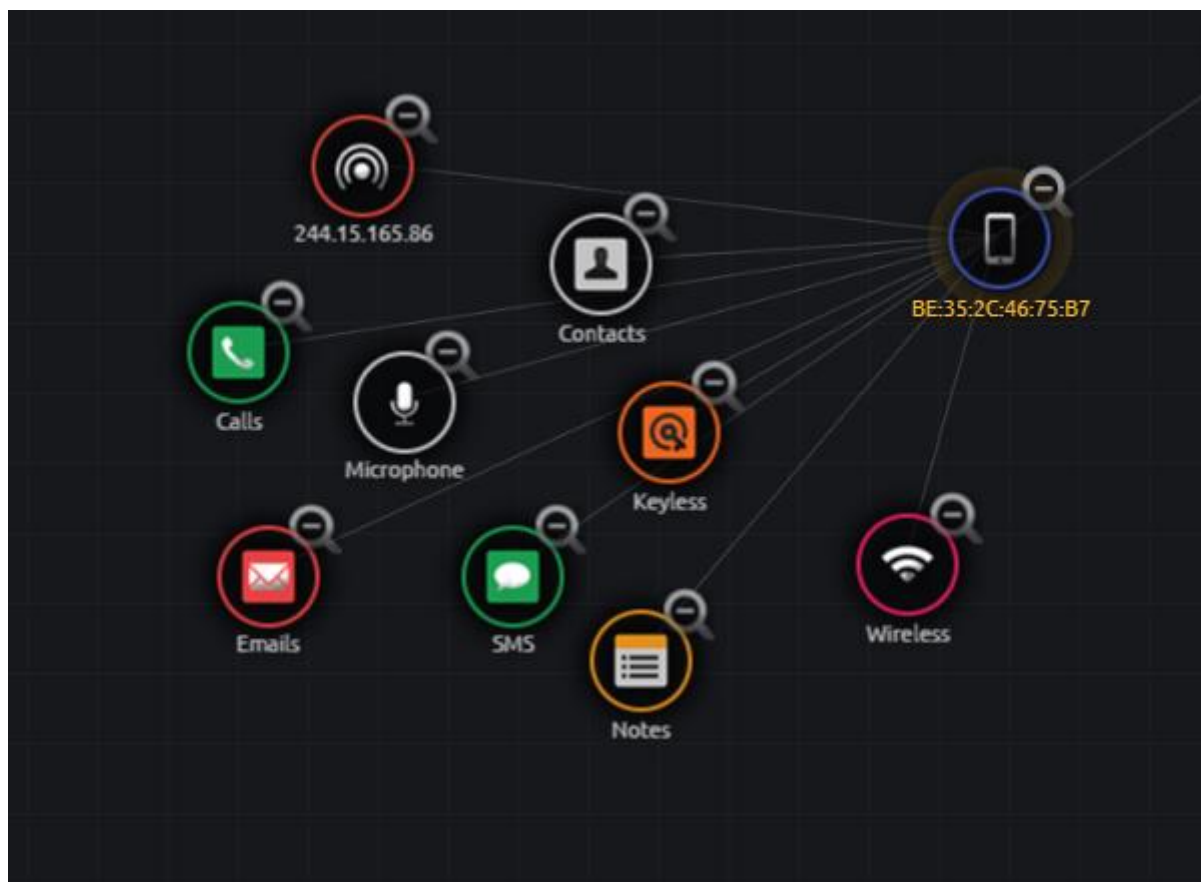


Figure 14 - NiteTeam4 Diagram of the Phone

Below is a photo of a test Cyber Range that was developed and would allow members of the Ethical Hacker Society in college to test the network and the capabilities of the software in explanation of the functionality and uses of a cyber-range.

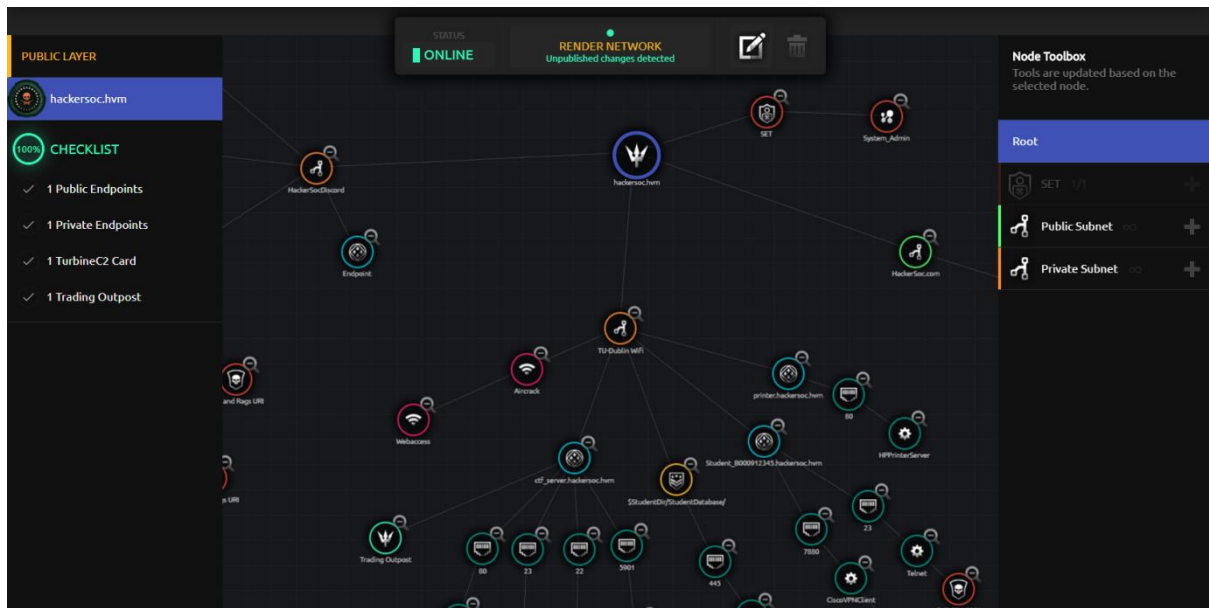


Figure 15 - Full layout of a NiteTeam4 Cyber Range

2.3: Literature Review Summary

Throughout the research on existing cyber ranges the team came across a paper called "A Survey of Cyber Ranges and Test beds" (Magrath) written for the 'Australian Government Department of Defence - Defence Science and Technology Organisation'.

As the title suggests this paper is a review of currently existing Cyber Range's and training infrastructures or test beds. In the paper they categorise each Cyber Range by type, in which they class each one as either 'Simulation', 'Emulation' or 'Overlay' and then also note their supporting sector as either 'Academic', 'Military' or 'Commercial'.

This latter part of their classification on supporting sector is not relevant to the current projects research as it more about the economic and political aspects of each Cyber Range and the team is more interested in the technical aspects. Pure Simulation and total hardware Emulation are 2 extremes on a spectrum or continuum of Cyber Range's, Most Cyber Range's fall somewhere on this spectrum.

2.3.1: Overlay

An Overlay network is essentially a network that operates logically on top of another network. In the context of cyber ranges, it is not really necessary to go into any depth about what an overlay network is. It simply refers to the architecture as being a special purpose logical or virtual network running on top of the supporting infrastructure which could refer to Ethernet or could refer to the internet itself.

2.3.2: Emulation

Full hardware emulation Cyber Range's run real software on real hardware, they are therefore the most realistic type of cyber range and are considered to be the most accurate and high resolution in terms of results of experiments and scenario's that are executed on this type of system. So, the major advantages are realistic results and high fidelity concerning the results of tests that are run. For example, this type of system would be much more suited to testing Distributed Denial of Service (DDoS) attacks due to the limitations on resources being more representative of real-world systems. However, the major disadvantages of such a system are high cost and lack of both scalability and flexibility.

2.3.3: Simulation

Simulation type cyber ranges on the other hand virtualized the systems on them logically, this could mean virtualized Operating Systems and virtualized network infrastructure. Considerations for this type of cyber range are that a disadvantage may be that there could be some instances in which completely virtualized systems might not respond the same as

the real-world systems which they are meant to represent. With testing the scenario of a DDoS attack for example in which limitation of resources are relevant to the outcome these types of systems wouldn't always reflect accurately how such systems would respond. Advantages of a simulation Cyber Range however are low cost, flexibility, scalability and portability.

The best cyber ranges would be the ones that take the advantages of each type and use them in combination. In the case of the project however the team would not have the project resources for an emulation type Cyber Range, and this would mean needing the aspects of the simulation type Cyber Range such as low cost, flexibility, scalability and portability.

2.3.4: Gamification

This section on gamifications covers the most important area that is currently in the industry using basic psychology and the rewards system so that the users feels joy from completing tasks while learning complex items in a fun way.

2.3.4.1: Gamification in Education

The concept of gamification in learning is about applying game-like elements and game design structure and mechanics to a training environment such as classroom educational media, workplace training, online courses etc. In the traditional school and college system the GPA (Grade Point Average) can be likened to a high score in a video game and therefor encourages students to perform better because this score and other rewards encourage friendly competitive spirit.

Gamification in general will boost user engagement and motivation as feedback and rewards for participant's achievements will allow them to view their progression through their training and provide them with a dopamine hit in the same way killing any enemy in a video game and levelling up or getting lots of likes on a social media post. The concept of gamification has been shown to boost user engagement, motivation and loyalty but most of all has been shown to increase learning and retention of information significantly more than traditional methods of learning.

The military have always used for example flight simulators to train pilots, and other gamified simulators for tank drivers etc. NASA will use simulators to train astronauts to pilot spacecraft and also to study the effects of long-term isolation that will be experienced on space missions.

2.3.4.2: Gamification in Cyber Security Training

Gamification in Cyber Security has evolved into the form of CTF (Capture the Flag) competitions of which computer science and cyber security students can participate and learn new skills while enjoying the game like competitive spirit of a contest. The Team have explored various CTF's elsewhere in various locations while looking into multiple studies.

One exemplary example of gamified learning that was assessed in this research is called 'Nite Team 4'. This is essentially a game but completely based on how a cyber-operator would infiltrate a target network using known exploits and techniques, social engineering and CVE's (Common Vulnerabilities and Exposures). So as a player progresses, they are learning cyber security training.

The project platform 'Hostile' attempts to provide isolated network simulation instances that users can access and attempt a challenge. If the user completes the challenge, they can have that sense of achievement of a gamified system and move on to the next challenge/level.

2.4: Technology Review

This section describes and explains the different technologies that are being used, could be used and inspirations for uses. It also explains the reasons for why each section solution is good for the project which includes pros and cons of each.

2.4.1: CTF Frameworks

- CTFd: Open source platform, designed to be fully modifiable as needed by administrators, which provides an improved interface and controllability for participants of Capture the Flag events, making it more user friendly and faster for setting up CTF events. The downside of CTFd is that is designed more to host Jeopardy style challenges than to simulate real world systems. This platform is most commonly used in varsity level CTFs across Ireland, such as Zero Days Security (Mark Cummins), Hack Trinity (Rory, 2018) and TU Dublin Ethical Hacker Society (HackerSoc, 2018) CTF's due to its versatility and friendly appearance without the learning curve for new users.
- Facebook CTF: This platform was designed and originally intended for use in Facebook's internal CTF events; this project has become open source and has been used by multiple organisations such as Facebook, Deloitte and the International Cyber Olympics. This platform has a little learning curve in usage and with the fleshiness of the platform can be a little overwhelming for new users.
- Nightshade: Similar to the CTFd platform Nightshade opted to make a simple style platform that was welcoming to users, and allowed for lots of customisation in the development and design of the CTF challenges, although these challenge like CTFD are limited to jeopardy style challenges, without modification of the original code.
- PicoCTF: This platform was designed with multiple uses in mind, and primarily designed for use with the PicoCTF that happens annually and is open to all users from around the world, this is a platform that his highly regarding, due to its challenges and ability to hosting multiple variants of programming challenges.
- PyChallFactory: This platform was designed mainly with the functionally of developing and testing challenges to be hosted on other Capture the Flag platforms, it is completely written in Python, meaning it can be easily modified as needed and be hosted in a simple flack "python web server" alongside a front face webpage. This platform being that it is mainly used for testing means it has no major usage in other CTF events.
- SecGen: The SecGen Platform was designed with Cyber Ranges and training in mind, the project is open source, and allows for a administrator to at will build fresh vulnerable operating systems based around Linux and Ubuntu. These machines and built with random data, vulnerabilities and configurations that allows for randomisation of newly built machines. In a training environment and Cyber Range, it allows for fast and rapid deployment of systems.

2.4.2: Virtualisation Technologies

In this section the team look at and give their input on different types of virtualisations Technologies that they have access to for this project, including some well-known utilities and some not so well known.

VMware ESXI

VMware is one of the goes to companies in the many of the cyber industries for virtualisation technologies is VMware, they develop multiple virtualisation utilities for example VMware workstation a hosted virtualisation utility for allowing multiple operating systems to be virtualised off a current operating systems hardware. They also develop utilities for bare metal systems, for example the VMware ESXI utility that is installed on blank slate servers that allow for the ESXI software to use all the available resources, whereas with the hosted you are unable to use all the available resources.

There are benefits to using VMware ESXI as a server base, since it is an extremely well known and widely used platforms by companies, there are multiple external plugins and scripts that can be found on GitHub to allow the platform to be taken even further, a good example of this would be the web application, Jenkins that allows for full server automation trough GUI based scripting for the server.

The major downside of VMware as a company and of the software utility ESXI is that the software code is closed source, meaning the code is not available to be used. One of the big issues with this is the fact that it is unable to be externally controlled or configured past what the VMware Company want you to be able to do with the software. Although they do provide a utility that can be installed over their ESXI server platform, called vSphere the vSphere tool allows for a lot more controllability over the ESXI platform for uses such as more in-depth network controls, the ability to clone and create templates of current working systems. The major downside of this is the cost of the both the ESXI software and the vSphere software. At current for a single license for use of both the VMware ESXI and vSphere tool, the cost is \$4,995 us dollars directly from VMware the company.

Thanks to a student program through VMware education for the VCP “Virtualisation Certified Professional” certification program, one member of the team has access to the license to use both the ESXI and vSphere, software. Thanks to the features in the software it would make a very easily controllable and configurable platform to be able to build a full working Cyber Range for multiple purposes, as with the with a few configurations and setting up with other open source automation tools such as Jenkins web UI or other, It can be turned into a one click configuration tool trough a web application.

When the project was original chosen and the original design phases was taught of the original Idea was to utilise the abilities of the VMware ESXI and vSphere technologies, by developing a front end that would allow for a user to build a Cyber Range for training purposes with little to no prior knowledge of the software or the configurations.

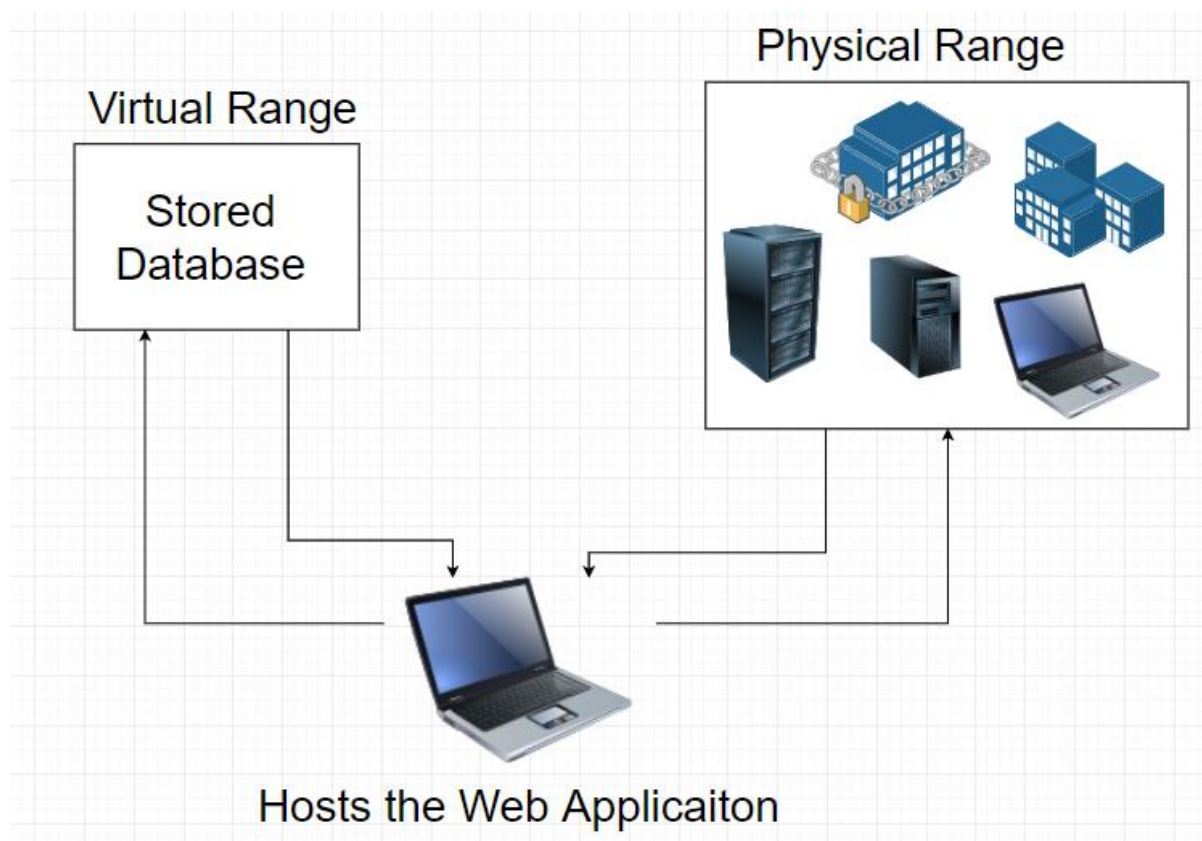


Figure 16 – Design for VMware layout

The above diagram was one of the first set up ideas that was designed as a group in a group meeting, that was wanted to be developed know what the configuration would be but configuration wouldn't be but it was known the team wanted it to be built and to use what resources where available to the team, which was at the time, access to a VMware ESXI server running the vSphere tool.

For the networking portion the downfall of the VMware system as the fact that it does not allow for virtualising of routers within its configured utility's, if a user wanted to use a virtual router they would need to install a Router Operation system such as OpenWrt an open source lightweight operating systems that routes network traffic. The upside is that the team had the ability to virtualise as many switches as wanted and this would then allow us to use VLAN's "virtual local area networks" so in theory segment of the network that would replicate the need for post exploitation to get access to a system further down the line, and to not have that system available to the main body of the network.

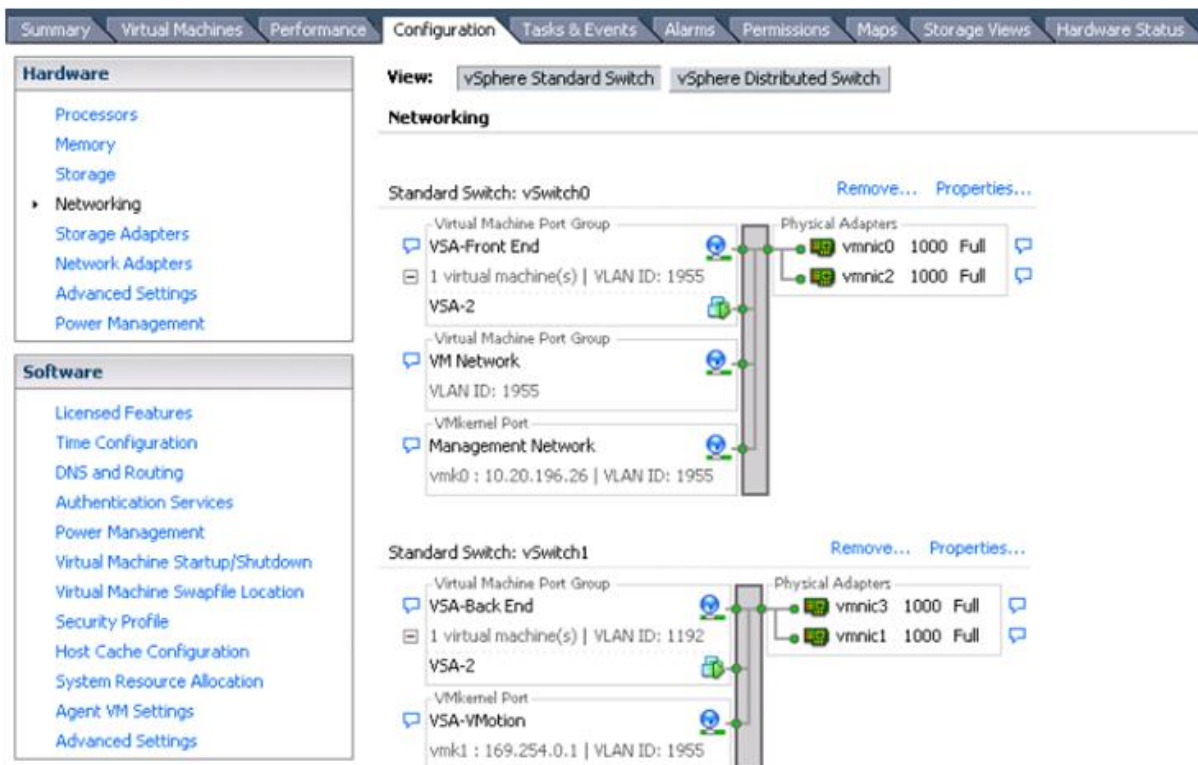


Figure 17- Layout of Virtual Switching environment in VMware

Above is an example of a virtual switching configuration within vSphere, you can see the different virtual switches configured, along with individual VLAN's configured segmenting different portions of each switch off while at the same time linking them back to the physical adapters available so they can connect outside of the server. A feature that was beneficial to vSphere is the ability to import and export templates, where this system could build a ton of vulnerable machines, and turn them into templates which are relatively small files and upload them to a service like GitHub. This would allow anyone with access to the repository of templates to download a template and spin up a vulnerable system in a matter of seconds, speeding up the whole processes, of setting up a cyber-range for testing on. One of the biggest advantages was the ability to be able to set up virtual machines to map to USB ports via a hub, which meant there was an ability to be able to add a physical side to the platform down the road, which would allow a user to have wireless cards that are connected to virtual machines. Mixing this feature with the ability to segment a virtual machine completely off from the others with VLAN's and network configurations meant the possibility of simulating air-gapped systems with only access via USB port. In the end the team decided against using VMware EXSI as the cost of using the software is out of the way of most users' budgets, it doesn't allow for operating system side changes or configurations that is closed source and proprietary software. Since the EXSI software is designed for bare metal system this requires a user to have a server they can install the EXSI software on to be able to run the virtual machines.

Unfortunately while researching this project it was found that their virtualization of systems was done on a proprietary closed source operating system that is installed over the bare metal server, this fact was a massive downside for the project, as the team wants to develop

something that is open-source, this also did not give us command line access to the backend of the server to do any modifications that would be needed outside of the application.

Proxmox

While researching open source alternatives to the closed source VMware software, during research the team came across Proxmox with a lot more documentation available. Proxmox is a bare metal virtualisation platform contains the same features as the paid vSphere software and more, such as allowing for a user to be able to import templates from their own locations; it allows for a user to use docker containers and has more detailed network configurations and settings, and similar to VMware ESXi it is fully controllable via a web application. The biggest difference between proxmox and many other virtualization platforms is the fact that it comes with all possible features pre allowed default and gives your access to the shell of the system running. Whereas with other platforms such as VMware ESXi, you only get half the functionality without getting an extension such as vSphere, which still does not give a user access to the internal system shell to make changes.

Since the options for configurations were very similar to the VMware system that the team could easily move over to the original idea and design to be able to design the following layout

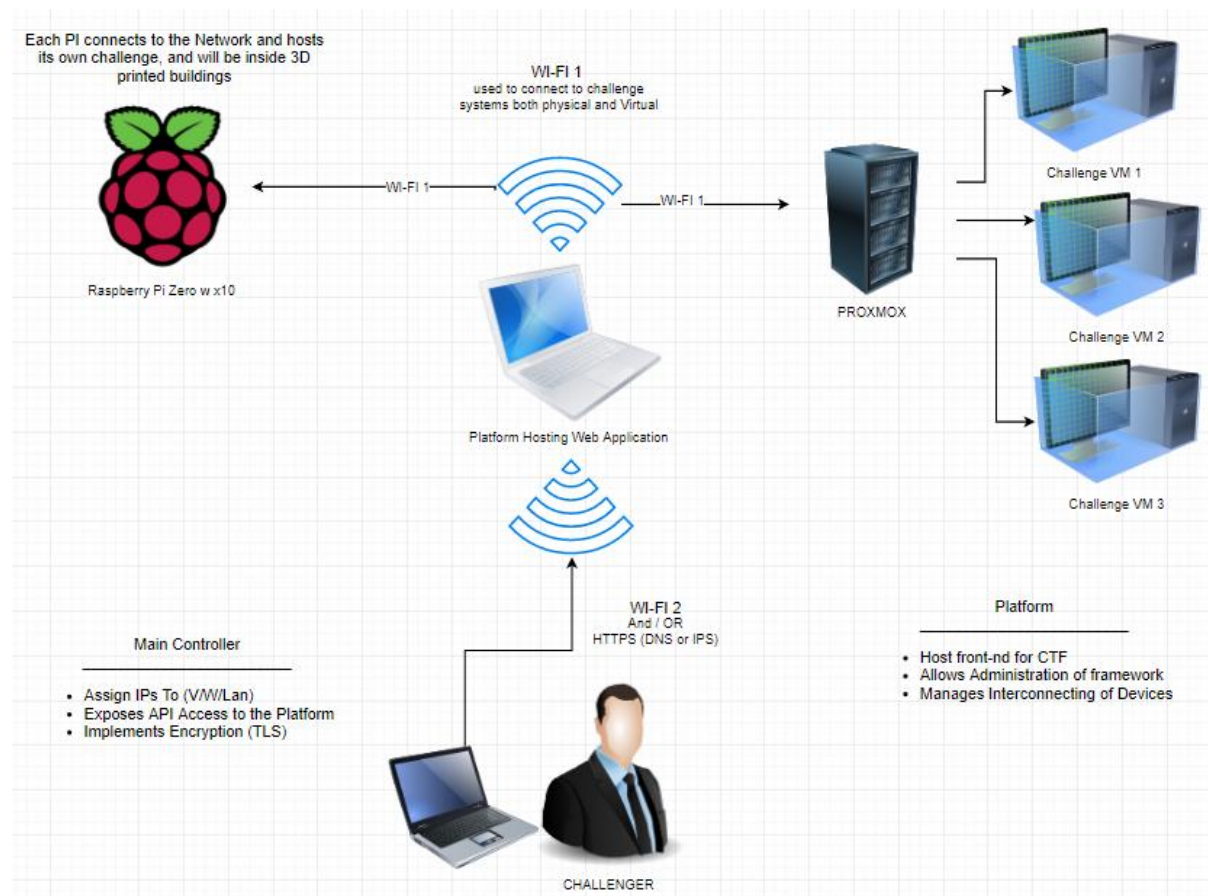


Figure 18 – Image of Proxmox Network idea and design

This is a very crude diagram but lays out the idea of how it could not control the server and the virtual machines built as needed.

The Proxmox allows for in-depth network-based configurations, it allows for setting up an internal DNS, Certificates, VLAN's, DHCP and virtual switching allows for a lot of more in-depth configurations. Another major benefit was that that Proxmox has a built-in firewall that allows a user to configure there on firewall rules, simply and fast and these rules can be configured on one machine or across an entire VLAN, or even an entire subnet allowing for an in-depth configuration on systems.

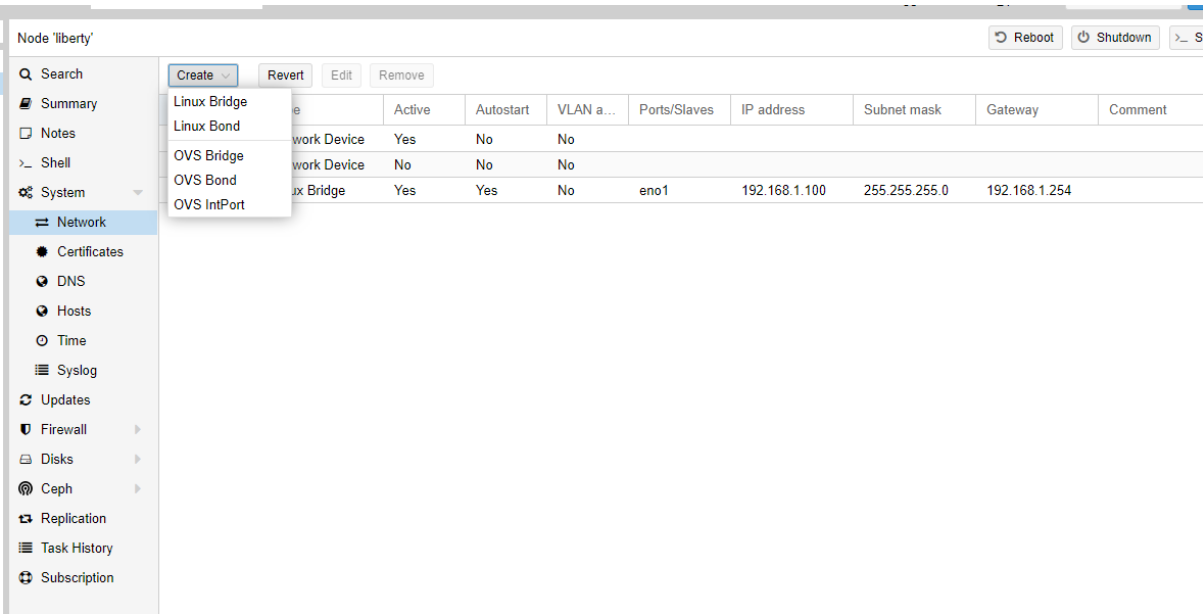


Figure 19 - Screenshot of Networking functions in Proxmox

To test the configurations, the team decided upon and built a test network on a server that was running Proxmox, to see how the virtual network works, Using the known network configurations and a couple of vulnerable operating systems.

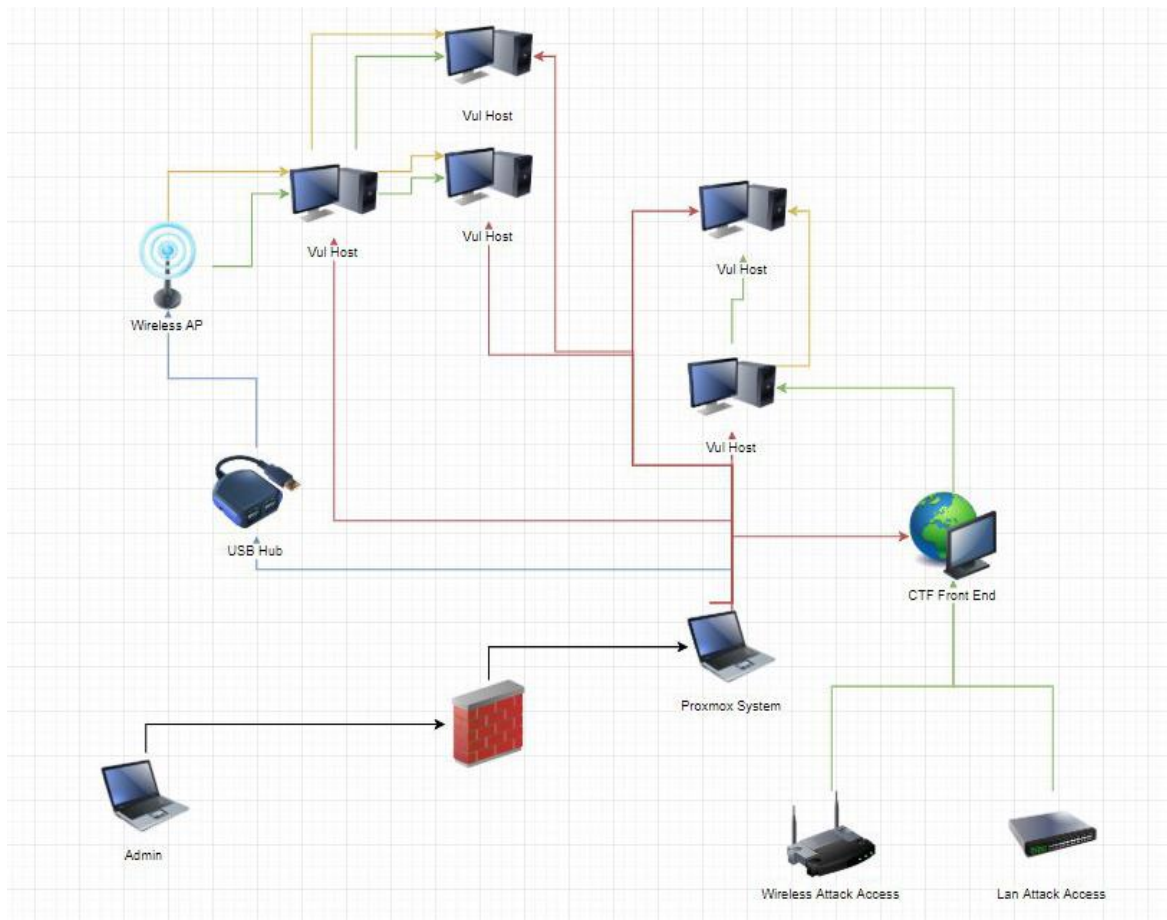


Figure 20 - Original Test Cyber Range that was built in Proxmox

Above is a logical diagram of how the network works, the green lines are connection routes to each machine from location, the Blue line is a physical connection via an USB device in this case a Wi-Fi access point replicated with a Netgear USB antenna. The yellow is a connection route from the physical access. The red lines are just to show that these machines are virtual machines sitting on the Proxmox server and should be disregarded. Through the build the team was able to test the idea of using VLAN's to segregate the network.

With one of the features of the Proxmox server, the group has the ability that they can add more storage and replicate multiple different data centres to possibly replicate multiple different locations in different areas such as offices in different countries or different floors and connect via a virtual configured switch.

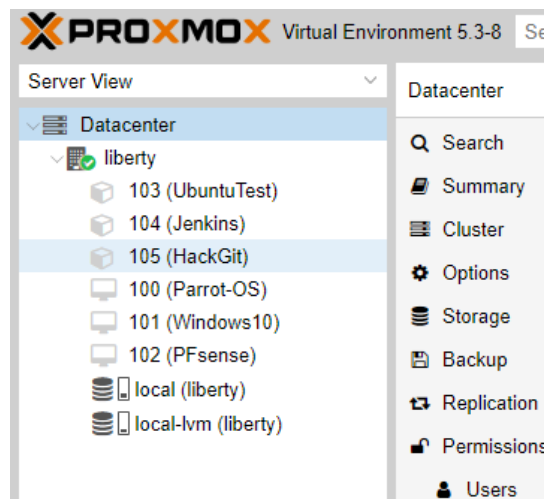


Figure 21 - Layout of testing machines on our local Proxmox datacentre

This was a success in the testing as this allowed the finding out that the network segmentation worked proper and made the network closed off, and logically seemed independent to the network. Although the testing of this build network was a success, The Team decided against using proxmox as even though it is free to use, it still requires a powerful machine such as a server to control and work, meaning that it is not a feasible cyber range for many including ourselves. As this project was to be as lightweight as possible it was important to be looking for something more lightweight and can easily be set up and configured.

While working with Proxmox in the platform testing phases the team came across an issue within the Proxmox system itself with the ability of internal routing, with building virtual routers. Unfortunately, with the capabilities of virtualization technologies this is not currently usable, although there is the ability to use virtualized switching this does not simulate working routers.

One way that found while researching is that this could be accomplished was using router emulating operating systems such as OpenWrt an open source router in the form of an OS or by using modified firewall rules and the VLAN functions that are available to simulate network paths.

Other issues that was discovered in the building phase of the test network was the storage requirements, one of the team members had a physical server with 1 terabyte of storage available for testing this environment. With each new challenge environment needed their storage would increase exponentially. The issue with this is that storage was limited on the server, and the only fix was to acquire more storage drives or arrays, this is costly and currently outside the budget of this current project.

Graphical Network Simulator 3 is an open source software framework designed to emulate complex networks and was released in 2008 by Solar Winds. It achieves this network emulation by virtualising both desktop operating systems and router and switch OS images. For emulating Cisco, it uses the Dynamic IPs engine and it also supports among other vendors technologies such as:

- Cisco IOS routers and switches
- Juniper network devices
- Arista
- Palo Alto Networks
- F5

It can integrate with Oracle Virtual Box, VMware, QEMU, Parallels and other virtualisation technologies and it also incorporates an array of components for virtualising other types of devices and operating systems and monitoring appliances.

Its purpose is for testing small to large scale network topologies and monitoring their operation. It is also used for education and training of network engineers.

In considering GNS3 regarding its relevance and usefulness to the project it was installed using virtual box in Windows 10 and setup a simple network configuration using various desktops, a switch and a cloud component.

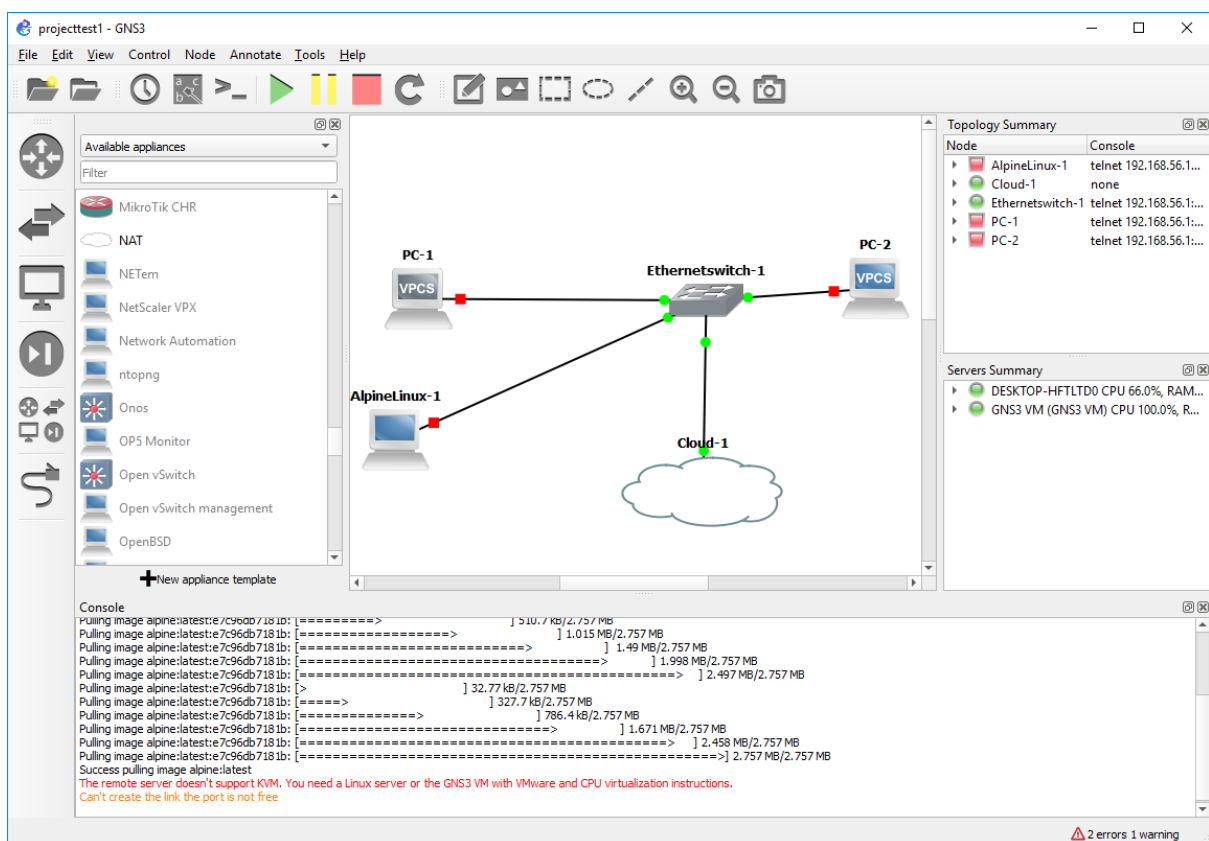


Figure 22 - GNS3 layout of a working network and software

Using GNS3 with Virtual box

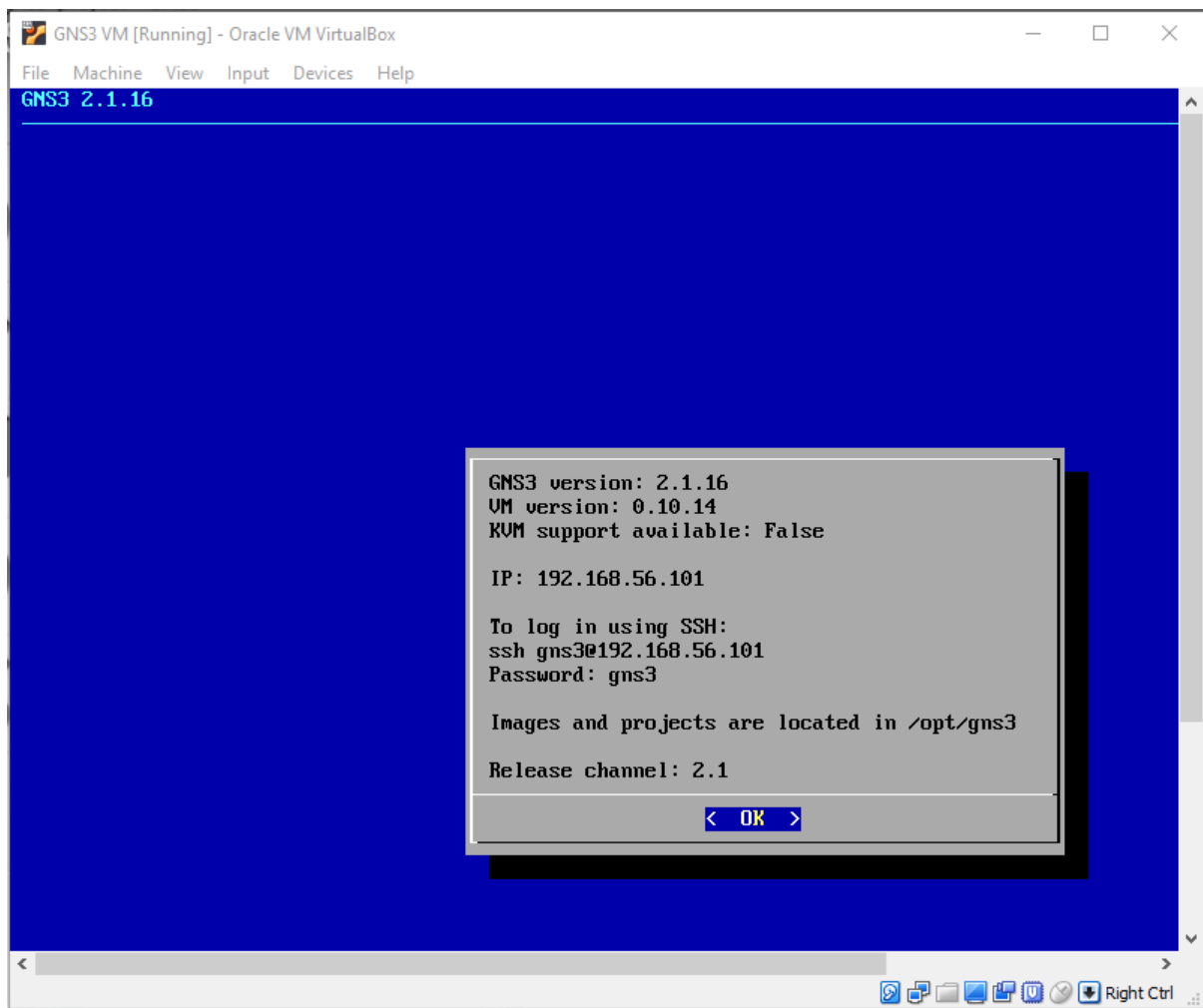


Figure 23 - GNS3 running on a virtual machine on Virtual Box

SSH into the GNS3 VM

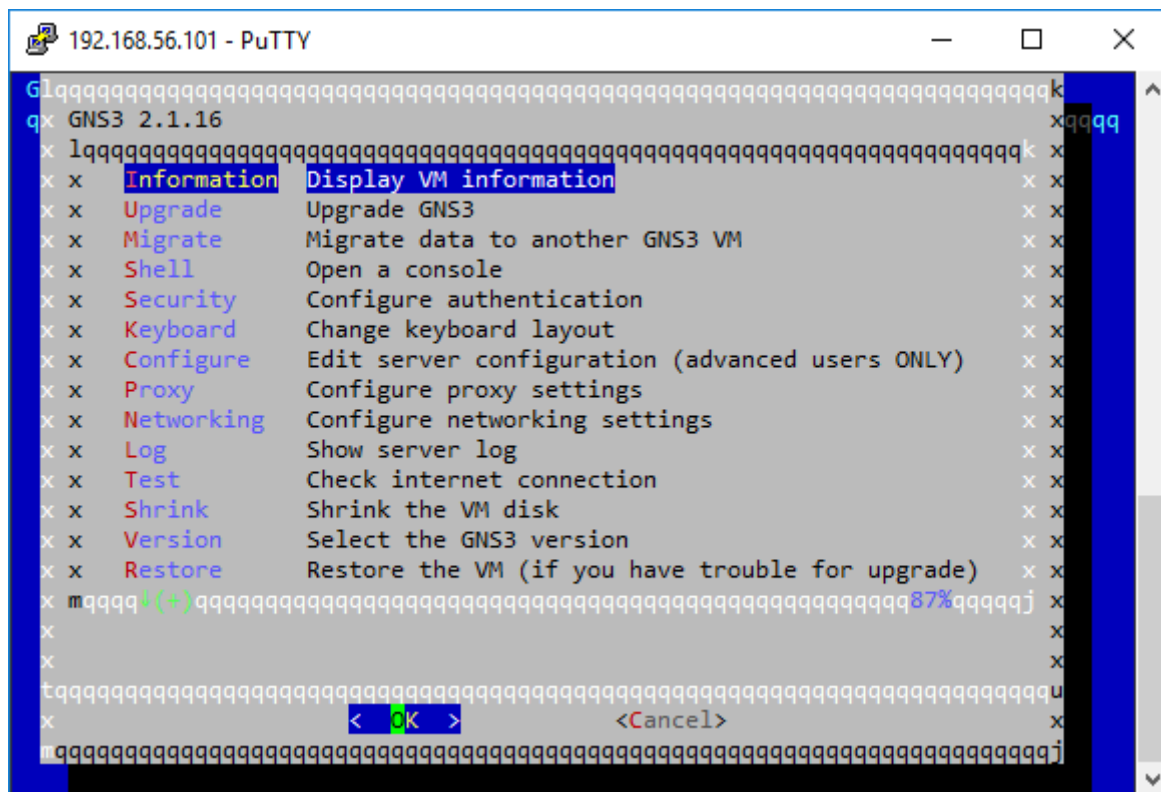


Figure 24 - GNS3 shell via SSH connection

On the surface GNS3 has a lot of similarities to Cisco Packet Tracer in terms of User Interface but that's where any resemblance ends. It is considerably more powerful in terms of complexity, range of supported network devices, an array of virtualisation components, to support virtualising almost any type of operating system image with network capabilities.

Whereas Packet Tracer is a simulation and simply provides an environment to learn router and switch commands and experiment with network topologies by connecting devices together, the devices are simulated so they do not contain the actual operating system image from those devices. GNS3 is an emulator rather than a simulator and it uses the OS image from the network device it is representing in its network topology. For example, it uses the Dynamic IP's emulator to support Cisco IOS images.

However, for the project it was concluded that GNS3 is not suitable for what the project was attempting. It is more suited to testing large scale network topologies and to training on specific vendor's network equipment. It doesn't provide the flexibility and portability that is needed and doesn't allow for the type of web app management interface that is needed.

Having said that it could potentially be used to implement a challenge into the existing framework where the challenger is tasked to exploit a CVE (Common Vulnerability and Exploitation) on a Cisco or other vendor's router or switch. This may however reduce the portability of the project setup.

Naumachia

Recently this software came into the teams focus after competing in the Inter-Varsity CTF, hack trinity hosted by Trinity College Dublin. This software was designed to be open sources and allows you to building individual challenge for each user or cluster of users, this could possibly allow for functionally such as development of individual challenges dedicated to each sector of security, Red Teaming and Blue Teaming for example. This software also allows us to visualize a service, such as certain ports and configurations being able to do large amounts of challenges for multiple users being able to build their interpretation of a Cyber Range. The Major benefit is this project is 100% open-source and free for all to use under a MIT license on GitHub.

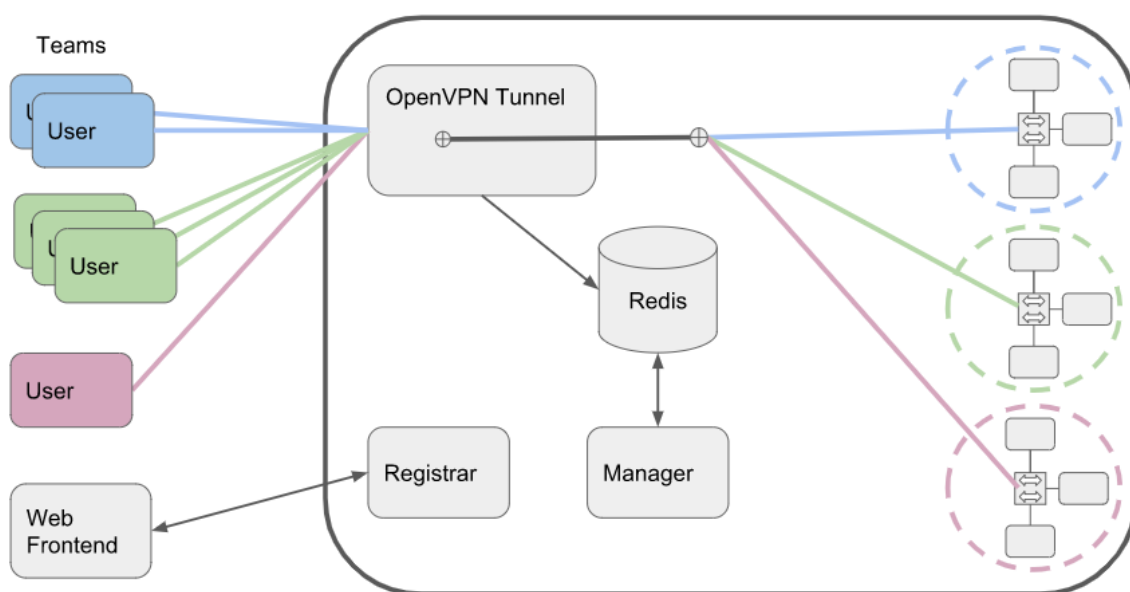


Figure 25 - Image of working network in Naumachia

ClearVM

ClearVM is a service that was found while researching alternative virtualisation platforms, the way in which it works is like Proxmox and VMware but is built and designed to 100% used through the web application. They have prebuilt and designed images that are installed on virtual machines as needed for a price. They maintain and deal with all the needed resources for the systems so there is no need for worrying about systems crashing or not having the needed resources.

ClearVM allows you to simply login via the web application choose a pre-built template operating system and the service will start up a virtual machine for you. The downside of using this service is there is no ability to upload your own virtual machines; you must use only the images that they provide which are very limited by their pre-set list of the most common operating systems.

[OVERVIEW](#)
[HOW](#)
[WHAT](#)
[WHY](#)
[PRICING](#)
[ROADMAP](#)
[CUSTOMERS](#)
[DOCUMENTATION](#)
[FORUMS](#)
[CONTACTS](#)
[LOGINS](#)

\$0/month

FREE

Up to: 2 Physical Servers
Up to: 8 Virtual Servers
Up to: 8 CPU Cores
Up to: 2 Teammates
Up to: 2 Geo-Locations
Community Support
[Sign Up Today](#)
No credit card required

\$40/month

STANDARD

Up to: 10 Physical Servers
Up to: 40 Virtual Servers
Unlimited CPU Sockets
Unlimited Teammates
Unlimited Geo-Locations
-
[Sign Up Today](#)
No credit card required

\$140/month

PREMIUM

Unlimited Physical Servers
Unlimited Virtual Servers
Unlimited CPU Sockets
Unlimited Teammates
Unlimited Geo-Locations
ClearCARE Support
[Sign Up Today](#)
No credit card required

\$0+/mo

ClearOS Community

\$4+/mo

ClearOS Home

\$8+/mo

ClearOS Business

\$16+/mo

ClearOS Hosted

\$0+/mo

CentOS 7

\$0+/mo

Arch Linux

\$0+/mo

Debian 7.1.0

\$0+/mo

Debian 7.3.0

Figure 26 - Screenshot of the Pricing and availabilities of ClearVM website

2.4.3: Financial Considerations

This section mentions some important points regarding finance, which someone who wishes to try this project should consider in advance.

Public Hosting

Since the web application will be connecting to multiple servers that will each host the challenges, a hosting service, or personally owned hardware with the appropriate power, would host the instances for testing the web application.

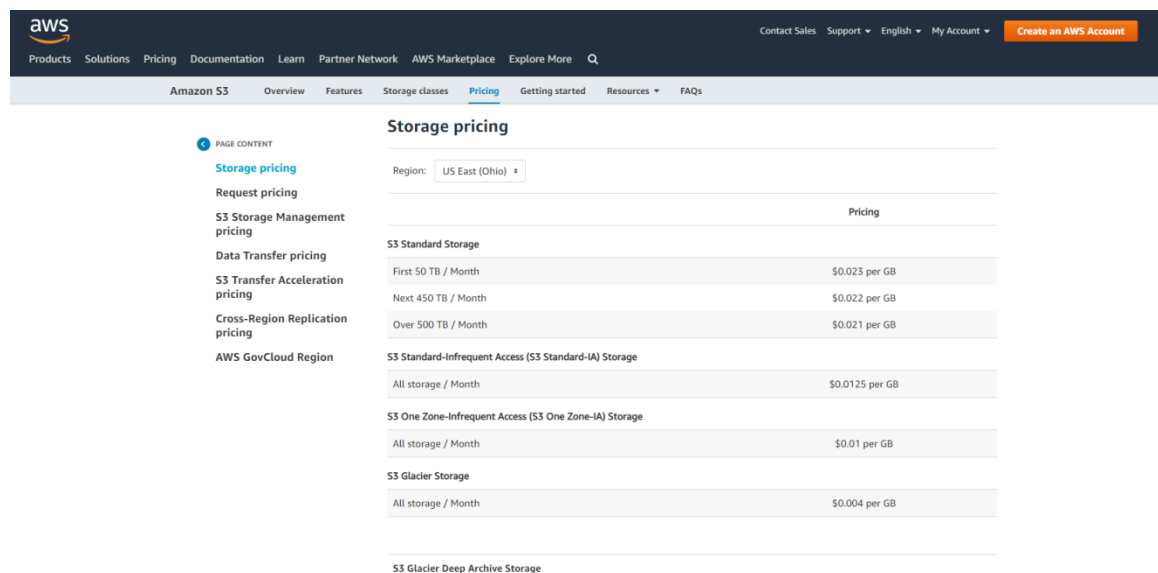
Numerous different options exist for hosting the server instances on a worldwide scale, including some of the following.

AWS

Amazon offers a limited selection of servers for free for up to 12 months, which includes default Ubuntu installations, meaning that an instance compatible with this project could be set up in under an hour, and for no cost.

This option would be well suited to running Hostile at cyber security events or remote training sessions, as it is a portable and globally accessible service.

AWS website - <https://aws.amazon.com>



The screenshot shows the AWS Storage pricing page for the US East (Ohio) region. The page lists various storage options and their corresponding prices per GB per month.

Storage Class	Pricing
S3 Standard Storage	
First 50 TB / Month	\$0.023 per GB
Next 450 TB / Month	\$0.022 per GB
Over 500 TB / Month	\$0.021 per GB
S3 Standard-Infrequent Access (S3 Standard-IA) Storage	
All storage / Month	\$0.0125 per GB
S3 One Zone-Infrequent Access (S3 One Zone-IA) Storage	
All storage / Month	\$0.01 per GB
S3 Glacier Storage	
All storage / Month	\$0.004 per GB
S3 Glacier Deep Archive Storage	

Figure 27 - Amazon AWS Storage Pricing

DigitalOcean

DigitalOcean has a variety of applicable services which, just like AWS, includes options to install the latest version Ubuntu Linux. Although it does provide the necessary services, there are no options to set up a free server.

DigitalOcean website - <https://www.digitalocean.com>

MEMORY	VCPUS	SSD DISK	TRANSFER	PRICE
1 GB	1 vCPU	25 GB	1 TB	\$5/mo \$0.007/hr
2 GB	1 vCPU	50 GB	2 TB	\$10/mo \$0.015/hr
3 GB	1 vCPU	60 GB	3 TB	\$15/mo \$0.022/hr
2 GB	2 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr
1 GB	3 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr
4 GB	2 vCPUs	80 GB	4 TB	\$20/mo \$0.030/hr
8 GB	4 vCPUs	160 GB	5 TB	\$40/mo \$0.060/hr
16 GB	6 vCPUs	320 GB	6 TB	\$80/mo \$0.120/hr

Figure 28 - Digital Ocean VPS pricing

Scaleway

Scaleway has a wide range of servers available, custom built for different purposes, just like AWS. Although it lacks an option to create free servers, the service is more affordable than DigitalOcean, and would be well equipped for running medium to large scale events in Europe, since that is where the servers are physically located.

Scaleway website - <https://www.scaleway.com>

Cloud Instances

Cloud instances delivered in seconds with backup, network and security options.

Featured Offers

New

DEV1-S
Development Instances

2 vCPUs
2 GB memory
20 GB NVMe SSD disk
1 flexible public IP
100 Mbit/s unmetered bandwidth
99.95 % SLA

€2.99
per month

Discover full range

New

GP1-XS
General Purpose Instances

4 vCPUs
16 GB memory
150 GB NVMe SSD disk
1 flexible public IP
400 Mbit/s unmetered bandwidth
99.99 % SLA

€39
per month

Discover full range

New

RENDER-S
GPU Instances

10 dedicated cores
1 dedicated P100 GPU
45 GB memory
400 GB NVMe SSD disk
1 Gbit/s unmetered bandwidth
99.99 % SLA


€500
per month




Discover full range


Figure 29 - Scaleway VPS Pricing

VPS comparison

A service called VPS Comp was used to compare the prices of different VPS providers, based on what was needed for the testing phase of this project.

VPSCOMP

Add provider   

I am looking for VPS from  Earth with some bandwidth.
Minimal CPU: 1 core, RAM: 128 MB, HDD: 1 GB











		↑ Cores	↑ Memory	↑ Hdd	↑ Network	↑ Transfer	Updated	▲ Price [EUR]
	Time4VPS XS Plan	🔌 1 core	💾 512 MB	💾 20 GB	🌐 400 Mbit	📶 1 TB	8 months ago	0.99 EUR
	BuyVM OPENVZ 128	🔌 1 core	💾 128 MB	💾 15 GB	🌐 100 Mbit	📶 0.5 TB	8 months ago	~ 1.12 EUR
	BuyVM OPENVZ 128	🔌 1 core	💾 128 MB	💾 15 GB	🌐 100 Mbit	📶 0.5 TB	8 months ago	~ 1.12 EUR
	Ramnode 128MB SVZ	🔌 1 core	💾 128 MB	💾 12 GB	🌐 100 Mbit	📶 0.5 TB	8 months ago	~ 1.12 EUR
	Ramnode 128MB SVZ	🔌 1 core	💾 128 MB	💾 12 GB	🌐 100 Mbit	📶 0.5 TB	8 months ago	~ 1.12 EUR
	Time4VPS S Plan	🔌 1 core	💾 1 GB	💾 40 GB	🌐 400 Mbit	📶 2 TB	8 months ago	1.49 EUR
	Ramnode 256MB SVZ	🔌 1 core	💾 256 MB	💾 25 GB	🌐 100 Mbit	📶 1 TB	8 months ago	~ 1.79 EUR
	Ramnode 256MB SVZ	🔌 1 core	💾 256 MB	💾 25 GB	🌐 100 Mbit	📶 1 TB	8 months ago	~ 1.79 EUR
	HostHatch Package #1	🔌 1 core	💾 256 MB	💾 30 GB	🌐 1 Gbit	📶 1 TB	8 months ago	~ 1.79 EUR
	HostHatch Package #1	🔌 1 core	💾 256 MB	💾 30 GB	🌐 1 Gbit	📶 1 TB	8 months ago	~ 1.79 EUR

Figure 30 - Screenshot of Website that compares VPS pricing online

With the capabilities of systems like Naumachia it allows us to be able to run the cyber range on low power system such as a using a free web server from Amazons AWS “Amazon Web Service” program that allows free hosting of small projects on their platform. This could allow us to potentially host the project for free.

Self-hosting

The option of self-hosting was discussed for this project, including purchasing second hand servers from different sources, such as company liquidations. However, with this consideration, the main worry would be the power consumption and processing power. The more power is used, the higher the electricity costs. Up to date hardware would be more efficient and maintainable, but the initial cost would also be higher.

The main disadvantage with self-hosting is that users will need to have constant access to the system, which means the system would have to do all the following things:

- Consume a reasonable amount of power
- Maintain availability for legitimate users
- Maintain stability for legitimate users
- Protect against various possible attacks

In the testing stage, this project did not involve setting up a self-hosted server. Instead, AWS Free Tier servers were used because of the cost advantage.

Summary

Due to the nature and size of this project, it does not initially require a large amount of resources to run. In fact, the entire web application could be hosted on a small computer running on an ARM based CPU, such as a Raspberry Pi.

In the future, the project may require multiple online server instances to replicate other styles of networks, such as for running larger events, but at present there is no need for this type of scalability.

2.5: Market Research

When first deciding to design and build a cyber-range, we decided it was best to get the opinion of the community of people in the information security industry who would be directly interested in this project. The purpose of market research is to be able to understand the needs and requirements that multiple industries would need from a project such as this project. The collected data can be used to design and develop functionally, features and even the graphical look to best please multiple communities. Based off the projects original design we had we wrote up this question which was given to a large Information security community of professionals,

“What would you say to a cyber-range, that is extremely lightweight and when I say lightweight. I mean runs on a laptop and instead of building virtual machines running on a virtual network with racks of servers and equipment, instead it crates services, that can span over multiple instances meaning mass connection between companies, this can generate traffic for blue teamers or host challenges for red teamers and these services are also individual to each user meaning no cross contamination of results from users on the platform, that is common with virtual machines where one users connection attempts kicks another user”

The replies that where received where wide and varying from multiple industries such as big four companies, government agencies, law enforcement and small corporate security teams.

“This could be revolutionary in training of new and current staff members in our industry, both in cost and amount of time needed to set up a training environment” – Penetration Tester at EY USA

“A system such as this would be amazing for challenging the current staff we have on downtime between projects” – Security Engineer at Government Agency

“We don’t have any need for Red Team training but, we have been looking for solutions for training our forensic investigators and SOC staff in-house, this could be perfect for our needs” – SOC Manager UK Law Enforcement

“Just off the description you gave in the above comment this could be a mind-blowing system for running a CTF, it would make it so much fast and I wouldn’t have worry about other users resetting my box every few seconds” – Community Member

2.6: Summary

From the market research that has been conducted there is a clear and present need for a system such as the one being developed in this project, and from connections and conversations with military, government, corporate and students just by explaining the concepts and working of the project idea, they were extremely interested in finished project and the future capabilities of it.

3: Research Methodology

3.1: Research Questions

1. Q: Can a windows app be run in a docker container that runs on Docker on Linux or other host OS'?

2. Q: Can a Linux app be run in a docker container that runs on Docker on Windows? Or other host OS'?

3. Q: How does a Kubernetes or other container frameworks work and can they be useful to our project?

1. A: According stack overflow (Stackoverflow) the Windows apps cannot be used in containers on Linux and recommends either using virtual box or running that app on a windows VM. However, it also describes 2 more unconventional methods:

A: for simple applications wine might work (in a docker container). And it also describes a method that may work involving running a windows OS inside Virtual box (using Virtual Box Headless mode) so no graphical interface, and then inside a docker container, Using the NAT setup to pass thru any traffic from the docker container.

B: Using the Vagrant environment to run a headless windows docker container that runs on MAC and 'Probably Linux'

2. A: In 2017 it was demonstrated that Linux kit allowed for a Linux Subsystem in Windows which essentially then allowed the running of containers with Linux applications inside. Linux kit provides enough of Linux to allow for the containers to run on MAC OS, Windows, AWS and other cloud platform, and also on bare metal.

3. A: Kubernetes is an open source container storage platform. It written by Google and is for managing multiple containers on a system. The system is thought by some to be overly complex. An example of the abilities and scalability of Kubernetes is the Pokémon Go app. Given this example the implications are that it could be useful to our project and deserves further research

Notes

Docker is not a VM environment it's an isolation container. It takes the isolation boundary of regular VM environments such as VMware and Virtual box for the OS level and down to the application level. Docker containers are way more efficient than using full virtualised OS's running as guests just to run one or two applications. The rest of each of those virtualised OS's doesn't need to be run (everything in the OS except the resources needed to run the app or apps that one needs or their system which is probably most the OS) (Fulton)

3.2: Rationale

The Overall reason for choosing the Naumachia system over the others, was cost and usage abilities, Naumachia can be run off over a laptop with just python, whereas the others needed a dedicate system to be running their own software to be able to do half of the required data

4: Design, Development and Implementation

As this is primarily a software project, a path must be picked that will help most effectively in designing and developing the code base. This involves looking at various types of SDLC models, analysing their strengths and weaknesses, and deciding which one best fits the criteria.

4.1 SDLC Models

This subsection describes some SDLC models, which are considered as popular among software development projects.

4.1.1: Waterfall model

In software development, the Waterfall model is a linear and continuous design approach consisting of phases, where one must be completed for the next one to commence (Royce, 1970).

The phases are as follows:

- Requirements - A clearly defined list of software, hardware and system requirements.
- Design - High level overview of the software and system architecture, which involves creating ERDs and relational models if necessary.
- Implementation - The software is built according to the design specification on a per-function level, meaning that every individual functional unit is created and tested separately on a case-by-case basis, which is known as Unit Testing.
- Testing - Units are combined and tested in tandem, and broken functionality is debugged and fixed.
- Deployment - The resulting software is run and/or distributed on the target platform.
- Maintenance - If issues are reported, they are analysed and resolved, then patches are applied, and the software is updated or re-released.

The flow and progression travels from the top (first) to the bottom (last) stage, resembling a cascading waterfall (SDLC Waterfall Model, 2014). This model would fit the project workflow

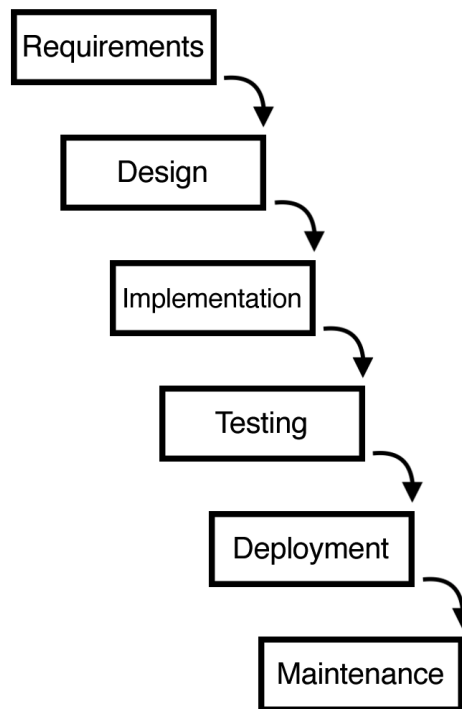


Figure 31 - Waterfall model diagram

4.1.2: V-Model

An extension to the Waterfall model, the V-Model (also known as the Verification and Validation model) is a very structured and disciplined framework that ensures testing is done for each step towards completing the end product. (SDLC V-Model, 2014)

It operates by constructing a design and test phase for each step in the development cycle. Only when the tests pass will the next stage commence. This works well with small scale projects that have a clearly defined set of requirements which are easily testable.

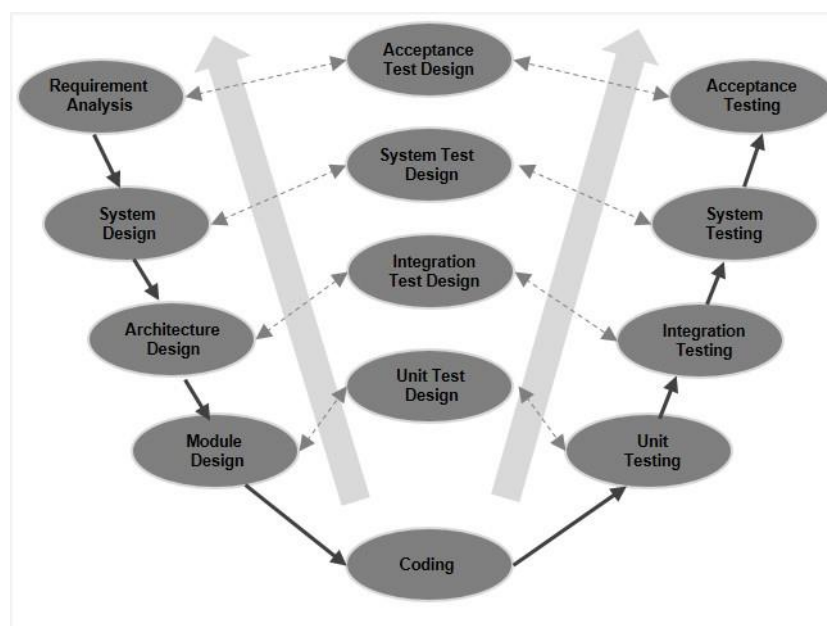


Figure 32 - V-Model diagram

4.1.3: Spiral model

This model revolves around an iterative and sequential development style, strongly emphasising risk analysis and incremental refinement of the product. It is heavily tailored to business related development projects where revenue is largely considered. (SDLC Spiral Model, 2015)

Although the development of this project does not involve business aspects, and thus does not take revenue into account, this model's phases could be a good fit for guiding the project in the right direction, regardless of which model is chosen in the end.

The Spiral model divides a development project into four phases, which are iterated through in sequence and continuously, going around like a spiral. (Boehm, 1986)

Although some title variations exist, the four phases are generally as follows:

- Identification - This starts with identifying business requirements and, in the following spiral iterations, adds system and unit requirements.
- Design - The first step creates a concept, later the architecture and the logic are moulded and ultimately a blueprint of the outcome is formed.
- Construction - This refers to the actual coding aspect of the project. Incremental changes are made with each spiral iteration and, once new information comes to light or is retrieved from testing and user feedback, the development progresses further until the final product is produced.
- Evaluation - Risk analysis and user feedback is heavily relied-upon to help with each phase and spiral iteration of the development cycle, making this phase the pivoting point of the whole spiral.

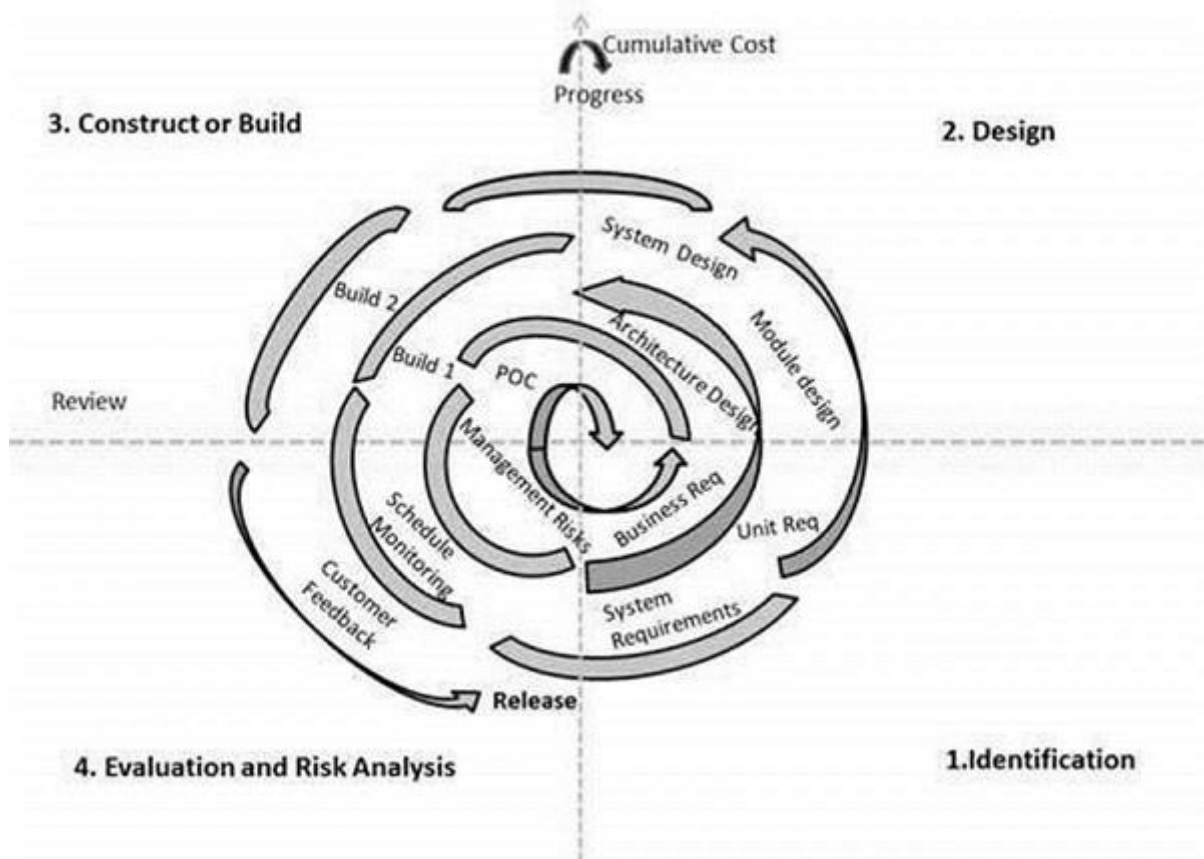


Figure 33 - Spiral Model Diagram

4.1.4 Summary

A few other types of SDLC models exist that could fulfil the project criteria, such as the Spiral model or the Iterative model, however they proved to be too complex for a relatively small project like this one.

In the end, the V-Model framework was used, simply because it is arguably the strictest and most thorough in its ways of operation and is well-suited to dealing with small but complex software development projects.

4.2: Requirements

For the project to function properly, it must be clearly outlined as to what an end-user should be to be able to do with the final software created. There are several different types of requirements, some are mandatory while others are optional, and there are also a set of specifications detailing any prerequisites for the project to function.

This section will list and define each requirement in its respective category, as well as explain what is needed to be accomplished with regards to the functionality and usability of the software, from the perspective of both an administrator and a participant.

4.2.2: Functional requirements

The following requirements define the necessary and additional functionality for the software to operate as expected.

Mandatory requirements include:

- Accessing instances via SSH for debugging
- Adding new server instances
- Administrating user accounts
- Configuring settings related to instances and their services
- Downloading enabled service configuration files
- Easy installation process
- Logging actions taken by the web application's users and administrator
- Point-and-click interaction in a clean interface
- Registering user accounts
- Toggling the availability of services

Optional requirements include:

- Cross-platform compatibility (i.e. Linux, macOS, Windows)
- Hosting challenges on the same server as the web application
- Importing and exporting configurations and SSH access keys
- Load balancing for DoS protection

4.2.3: Hardware and software prerequisites

These are the necessary steps that are needed to be taken prior to engaging with the software:

- Complete shell access to a server running Ubuntu 18.04
- Computer running either Linux, macOS, or Windows
- Python 3.6 or later

4.3: Development Process

This section will go over how the project was developed, including the development environment and the methods used to keep track of the team's progress.

4.3.1: Environment

The main technology that powers Hostile is Python, specifically version 3.6 or later. According to the Python Software Foundation (Python, 2001), the use of Python virtual environments to develop software is a great way to avoid common issues.

Some of the benefits of using a virtual environment include:

- Being able to easily switch between specific versions of Python modules.
- Not needing administrative permissions to install modules.
- Not needing to modify the system installation of Python.
- Portability, meaning that the project will run in the same way on all systems.
- The ability to dispose of the project by simply deleting its main directory.

The use of virtual environments has allowed us to seamlessly test the web application across different platforms, confirming that it functions properly on Linux, macOS and Windows.

4.3.2: Version control

For this project, a method was needed to make sure that any changes made by any group member were recorded and backed up in such a manner, that retrieving older versions of any part of the project would be trivial.

This is commonly referred to as version control. One of the most significant advantages of using it is to prevent the loss of source code and data. There are several different version control systems available such as Git, Subversion, Mercurial and Bazaar, and applying one of them to a project's main directory makes it become known as a repository, where each change made to it is known as a commit.

In the end, Git was chosen, as it is by far the most popular of them all, and thus it is better documented, and there is no shortage of help with Git on the internet.

4.3.3: Progress tracking

Having decided on a version control system, it became easier to view any changes made to the project's files in a detailed fashion. GitHub was used to help comfortably track each team member's contribution and visualise the differences between separate versions of the repository or a specific file.

If at any point a file had been lost, or the project needed to be reverted to an older version, the repository could simply be viewed in the desired previous state and the necessary data could be retrieved, or the repository could even be completely restored to that state.

4.4: Implementation

This section describes the process of implementing the software package on a code level, which technologies were used, and the design choices in implementing the web application. It also details how the software will be packaged and how easily someone could install it, with the help of the included automated setup script.

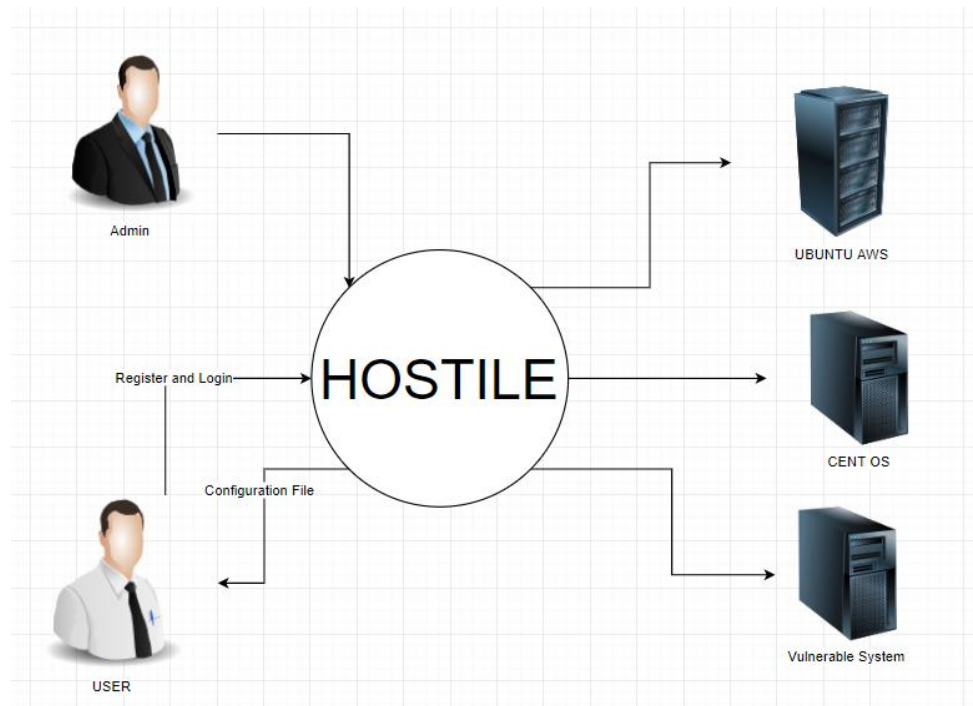


Figure 34 - use case diagram

4.4.1: Web application

The most essential element of this software project is the web application environment. The code base is primarily written in the Python programming language, specifically version 3.6 and above, and the reason for this choice is that these versions of Python are future proof.

4.4.1.1: Language choice

The Python Language Foundation will deprecate version 2 of Python by January 2020 (Guido van Rossum, 2014) and, since the major versions - 2 and 3 - are not strictly compatible with each other, it is a better choice to keep up to date with the language's development, despite the fact that most of the world still uses Python 2.

4.4.1.2: Framework choice

To help create an efficient and lightweight web application, this project employed the use of a popular micro framework of the Python ecosystem called Flask (Ronacher, 2010), which was designed with simplicity in mind (Grinberg, 2018).

Its goal is to provide a minimal setup in the creation of web-based software, by allowing users to pick and add functionality, as and when needed. This lets the project to be extensible, and it means that future contributors to the project will easily be able to modify various aspects of the code and add different features.

As well as that, Flask is highly scalable when it comes to running large projects on limited resources. Coupled with additional Python modules such as Green Unicorn or Celery, requests can be processed asynchronously, and handled with speed and efficiency.

For this software project however, a simple Flask development server setup will be used, which can also be run in production mode.

4.4.1.3: Layout and Integration

The main objective, when building the web application, is to meet the mandatory functional requirements outlined in the Design and Development chapter of this report. To fulfil these design goals, they must be addressed on an individual level. For each distinct feature, there needs to be a clearly defined section of code that is responsible for carrying it out.

This was achieved by constructing the project as a Python package and assigning a file for each logical function of the web application. The package layout is as follows:

- `__init__.py` - This is the main file that, because of its name, has a special meaning and defines the directory structure to be a Python package. It is responsible for importing and instantiating the classes that are defined in the other files, making everything work in synchronicity.
- `forms.py` - To fulfil the requirements regarding logging in or registering user accounts or any other form related activities, this file specifies the form processing aspect of the web application, including the security functionality through the sanitisation of user inputs.
- `models.py` - A vital part of the project involves managing persistent storage so that, if the web server crashes, no important data is lost. To achieve this, a standard and universal approach was used, one that revolves around databases, and was implemented in a purely Pythonic way.
- `routes.py` - The heart of any web application is the part that manages what users will see when they send various requests to the server. This file takes care of what is known as "routes" by managing the paths that need to be served to all the users. It is also the largest file because, for each distinct route, it must implement the necessary forms, database actions, templates, message sending, and page redirections.
- `static` - This is a directory that stores all the unchanging, user-visible, static files. These include any files that the web browser could load from its cache storage when accessing Hostile.
- `templates` - This is a directory that contains dynamic files that are served on a per request basis to users. Files in this directory are written mostly in HTML, with added sections written in Jinja2.

To expand on the routes of the web application, here are the routes that have been defined in the `routes.py` file:

- Route `/` is the home page and the first page that is usually accessed by users. For guest users, it shows a welcome page set up by an administrator. For logged in users, it shows some statistics about their usage of the services and instances. For admin users, it shows some general statistics about the operation of services and instances.

- Route `/info` is an informational page that can be modified by an administrator to inform users of notifications or important messages.
- Route `/register` display the user registration form where guest users can create an account, so that they can have access to the offered services.
- Route `/login` display a login form, allowing guest users to sign into their account if they have one set up.
- Route `/logout` sign a user out of their account if they are logged in, and then redirect the request to the home page.
- Route `/account` shows logged in users their personal page, where they have a limited ability to configure their account details.
- Route `/users` shows a list of all the users registered on Hostile. This route is only accessible from the administrator account.
- Route `/user` is a prefix route that shows any user's account page when appending `/USERNAME` to it, where **USERNAME** is any user ID that is registered on the web application. This is also only available to the administrator.
- Route `/instances` show a list of all the server instances that the web application has been configured to access. This is also another route that is only accessible by the administrator.
- Route `/instance` is a prefix route that shows a configuration page of a specific server instance when appending `/INSTANCE` to it, where **INSTANCE** is an ID of an instance registered on the web application. This is also only viewable by the administrator account.
- Route `/services` show a list of services offered by all the server instances. The administrator also has the option to enable or disable each service, whereas users can only view the currently active or enabled services.
- Route `/service` is a prefix route that, just like the other prefix routes, provides a page for the service specified by appending `/SERVICE` to it, where **SERVICE** is the ID of a service offered by an instance. An administrator can also enable or disable the service, as well as see statistics regarding how many users have tried to access the service.

Each aspect of the web application plays a critical role in making it function correctly, operate smoothly and present an easy way to debug any errors. Due to the ecosystem that was chosen - Python and Flask - there is much greater control over the functionality of the web application.

4.5: Networking

To manage the back end and networking aspect of this project, the Paramiko module from the Python package manager was installed and was used for connecting to server instances using SSH sessions. This kind of setup allows the web application to access any of the instances and configure them through the administrator account interface.

While setting up full shell access through a web browser is not a functional requirement, this project has the necessary tools that open it up to a much greater potential in the area of server administration and remote debugging.

4.6: Server Instances

In this particular case, an instance is defined as being a remote system that meets the following criteria:

- Can be accessed from the internet.
- Can be connected to via an SSH client.
- Has a compatible version of Linux installed, as per the prerequisites outlined in the previous chapter?
- Has a compatible version of Naumachia installed, and preferably running?
- Has the correct ports open to access each of the services?

In order for the project to function properly, each individual server must have the necessary aforementioned software installed and set up. A list of requirements for each instance and an optional setup script are provided to the administrator on the Instances route of the web application, which they can share with whoever manages each instance.

In the current working stage of the project, there are integrations built to interact with Naumachia - a multi-tenant network sandbox for security challenges (Graf V. ", 2017)- which is publicly hosted on GitHub. However, this can be further expanded in the future to support other software as well, beyond just Naumachia.

One example of additional software integration could be GNS3. Connecting to an instance running GNS3 would then entail setting up a method for getting access to the internal and virtualised network and granting that access to users of the Hostile web application. This could either be done with a VPN session, just like Naumachia, or by providing the user with an IP address to connect to.

Although the former method is preferable, a similar effect could be achieved by letting Hostile manage the connections from the web UI itself, making the UX more seamless to the user and less prone to connection faults due to it being taken care of completely on the back end.

4.7: Setting up Naumachia

Naumachia is a software tool written by Nate Graf which he describes as 'A multi-tenant network sandbox for security challenges' (Graf N. , 2017)

To begin with an instance of Ubuntu 18.04 is setup on AWS(Amazon Web Services) and made accessible by allowing inbound traffic to all ports (most of which should be closed later if not used)

In order to setup Naumachia a script was authored called setup-naumachia.sh which takes care of setting up the requirements and cloning Naumachia from GitHub

setup-naumachia.sh installs the following major components:

- * Git
- * Python 3.7.3
- * Docker
- * Docker-Compose
- * Naumachia

Beginning in /home/ubuntu/ run the setup script like so

./setup-naumachia.sh

Note: Naumachia's main directory is now:

/home/ubuntu/Platform/Naumachia/

unless otherwise specified run any commands from here.

If the script is successful the next step is to copy:

/home/ubuntu/Platform/Naumachia/challenges/config.yml

to

/home/ubuntu/Platform/Naumachia/config.yml

and edit this file to reflect the desired challenge to be run and its parameters.

An example config.yml exists at

<https://github.com/nategraf/Naumachia/blob/master/config.example.yml>

so that the format can be followed.

Next run

sudo ./configure.py

Now to create the *.ovpn config file that the user must load into OpenVPN on their local machine in order to connect to the challenge, Do this by typing

sudo ./registrar-cli challenge add user

sudo ./registrar-cli challenge get user

(where challenge is the name of a challenge to run)

The contents of the *.ovpn file are then output to the terminal

Copy this output, paste it into an ASCII text editor

On line 9 which looks like this:

remote challenge 2000 udp

Replace the challenge name with the IP address of the AWS server

And save this file as challenge.ovpn. This is the file the user will use to connect the challenge using OpenVPN.

Finally to run the configured challenge/s on the AWS server

sudo docker-compose up

if all went well the challenge/s should be accessible.

4.8: Scripting

In order to receive information from each server instance regarding the available services on offer, the web application will need to send requests to the instances at regular intervals. This can be done in a few ways, such as:

1. Setting up a cron task on each server that informs the web application of the active services, which then displays the results live in the web UI. This method is more involved because it requires that every instance of Naumachia is modified in a specific way to send messages to Hostile, while Hostile itself would have to run a separate process to interpret those messages.
2. Contacting all the instances via SSH at certain intervals (specified in the administrator settings) and requesting the names of all the active services. This would remove any requirements for Naumachia to be set up in a certain way, making the web application truly modular, as it would adapt to any instance it is given access to.
3. Setting up user accounts for managers of server instances, or assigning roles to users as instance managers, and letting those users manually adjust the selection of services they wish to offer. This would be even more involved than the first two methods because it would not be done automatically, and updates to services may not happen as promptly.

This project implements the second method, as it minimises the complexity of the setup and allows the web application to retain its small size by avoiding the unnecessary creation of more database tables and models.

The way this was achieved is as follows:

1. First, the "Paramiko" and "threading" modules are imported in the source code.
2. Then an SSH client object is created using Paramiko, one for each server instance recorded in the database.
3. Each interaction with any singular instance (i.e. each object) is placed into separate "Thread" objects.
4. To minimise the possibility of conflicts between each individual thread, threading.Lock was utilised to implement mutex locks, preventing any given thread from being interacted with by multiple users at the same time.

4.9: Automating the installation process

To make it as easy as possible to set up Hostile, a script was authored that can automatically set up a working environment to host the web application, provided that the prerequisites outlined in the previous chapter are met.

4.10: Packaging and releasing

To create a piece of software and release it to the public requires a certain selection of standards to be followed, depending on the size or type of project being worked on.

Since Git was already being used to manage the development of the project, it was decided that the web application should be released as a Git repository, purely in code form, with the necessary documentation included. The continued adherence to the GitHub platform allows to keep the development process streamlined, even after the initial release.

Potential users of Hostile will be able to download it either through the GitHub website, or using the project repository link and running a command akin to the following:

```
git clone https://github.com/PATH_TO_REPOSITORY
```

4.11: Summary

This chapter has identified the various SDLC frameworks available to work from, analysed the requirements set out by the chosen SDLC model, listed the prerequisites for the project, and outlined the general development process using the chosen technologies.

The final section covered how the web application was built, which technologies were implemented, the reasons for the implementation choices, the actions of each part of Hostile, the higher level user perspective, an example use case from the point of view of both a user and an administrator, as well as how the project was packaged and released as a whole.

5: Testing

The testing portion of the project is a highly important section as it allows us as a team to evaluate the usage of the system and find any possible bugs or issues when running challenges.

We can also gather feedback data from the users to see what changes they would feel would be best

5.1: Setup

For testing purposes, we decided to use a Free AWS “Amazon Web Services” server for hosting the challenges, as Naumachia is so lightweight it would not take a lot of resources.

For beta tester, we enrolled the help of the Ethical Hacker Society of TU-Dublin Blanchardstown Campus to login and attempt the challenges that we have set up for the testing phase.

As per the Financial Considerations section in the Literature Review chapter of this report, the service provider of choice for hosting Naumachia for the purpose of this project is Amazon. Hosting the Hostile web application was carried out on a laptop.

5.2: Testing

This section covers the tests themselves, including code snippets and images.

5.2.1: Testing SSH

The testing for a successful SSH connection was done using a script that attempts to do the following:

- Load the Paramiko module
- Establish some connection parameters
- Connect to an example server instance

```

1  import paramiko
2
3  # Create an SSH client
4  ssh = paramiko.SSHClient()
5
6  # Don't ask to verify keys for uncached hosts
7  ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
8
9  # Connection parameters:
10 # • Server IP or host name
11 # • Username
12 # • Password
13 # • Path to private key file
14 # • Remote path to Naumachia
15 host = "18.221.3.74"
16 connection = {
17     "username": "ubuntu",
18     "password": "",
19     "key_filename": "/Users/Vicky/Downloads/GroupProjectTest.pem"
20 }
21 path = "/home/ubuntu/Platform/Naumachia"
22
23 # Start the SSH session
24 ssh.connect(host, **connection)

```

Figure 35 - Paramiko Working Code

Having established a successful connection, the rest of the steps involved:

- Running a command on the example server
- Copying a file from the example server to the local machine
- Closing the SSH connection

```

26 # Run commands on the server
27 stdin, stdout, stderr = ssh.exec_command(f"cd {path} && ls -la")
28 output = "".join(chr(x) for x in stdout.read()).strip()
29 print(output)
30
31 # Copy files to/from the server via SFTP
32 sftp = ssh.open_sftp()
33 sftp.get(f"{path}/config.yml", "test-output/config-test.yml") # Receive file
34 # sftp.put("local_file", "remote_file") # Send file
35 sftp.close()
36
37 ssh.close()

```

Figure 36 - Paramiko SFTP Code

5.3: Testing Challenges

An example connection to a challenge was generated manually, to verify its functionality. The commands that were run on the test server instance to generate a user configuration were:

```
cd /home/ubuntu/Platform/Naumachia
sudo ./registrar-cli sticks420 add exampleuser
sudo ./registrar-cli sticks420 get exampleuser
```

The first line changes the current working directory to where Naumachia is located, the second one adds a sample configuration for a user called "exampleuser", and the final command retrieves the newly generated configuration.

```
client
nobind
dev tap
remote-cert-tls server
float
explicit-exit-notify

remote sticks 2003 udp

<key>
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCn7PcQ0BGXLAgn
Vhe4B0aE/w8z2+XCmpl/6CGbaPvkGdBe0I/HoiZP2vuRfK6T9c4TIDrvtpjiXXqm
AssAvVl0A9d9wCvb1Bohaypb/JgzeytV9Y8cLHHWMLEaBLoi4opz6hghmZWNPu/V
pdddbRcKEPy8lCh2d6XYWwOQTzA/e6eDwjKl5QuidtmHVrufi8Kzz3zuByIgwOwe
DwbG6YcNddWjB1kp4DzcCKQSp0Wwh0Pl18fRuDH9M0UhcBx4UbTPlNrKUCIqf8Q
tgYq6eVlTrSzE1YozrrLCmBVYxPA3DnSeGBbmYk29hHWivQYKpIsRYlNlJZ+w/H8
t1pBVAZ7AgMBAECggEAKu+/GHBrHwEmKerKc8Xk/zqMM5FVRRXsi7TZ6vFD+EXV
b1H+CwhfbWmriyQDo1dlUqSaDTLeEnseYhYyu8SufEvhzq8vB+WctcI5K5IaFlk
8jywU97VwJPopKNyyHquNgc0ueSiva3K+xDFaWf1Dv488Sbc1bWwFQrodVyaFJDq
PBEZ3I59k4nxrE+CI027qI/Yc0pYicPzzAPBynGJd95L02DG+tRjdRE00AQyp+ik
Pnsq+FEG1GqTH0IP3lVLxpe3gbJzWrkaIBA5Za+yRWn+1eqrL3MbBA9U8gz1J8ZG
nwBt1p0/w6N3Cth30MuVpTsr7waKeCnYduA3Ac8jmQKBgQDdBtH/qPMiiwEaap5Y
ewUP+PzQ+qUOMQ7w4ve1SGvqCnRXQwCyZDbqUQPQV7ZYmnnnQ5Hpou0n6k0pUTBk
qV/FoZoYpp7m1Tppr/9Y2VRlVE4y04wG8RfotyIa0lta4i5f8v5oF4EsPDGqQ8m0
yXmX7sY3DUPFhQ0vc730N9yHJQKBgQDCfyt7ouk7JSAF302brXKwAYQ9BPR7XHoH
FiCHZ0oFTylr5LbDcCoiqEg3dWoYHEX0Wg0jA7vAZQabfj000MsEUKZvJZzMntuu
Wnv1sCnnUsQpMDT2uvTDNeTns8QSUmmD94G3v+5ItVwBTY6BJenJYvw3kQxlv9Lk
EqBtDIE1HwKBgFlSaUltiJx8Gn0He2h73qi+0cT0liePEd+tibecReg1neMBy8cs
ErlM9kKbP9i8z0j1+K3sFLqrF9frTPw8wBWV4UAYRS2NT4C3ymT4rR0holsYSFp
C2Wdp0e7/SlUYLNr6v48CxrL9LI/Z00HzHjncPm0hZ/sLjDwEGMctD91AoGA0bCc
xjRhN7DgWb2NeIKvBP0+Njc2qSyZ6WYvydkt7Ns5Lw6NA1V0ui7uFL0X9gTU3xCi
2uLtmriciByb0F2wCh0ZVThz10G9yW5/XDWfiEstEiYa/6Aen5RpxHXwz3pwZ3GV
oVolpJ1ep+hIP5huStS2i3dTLQ6b0BiSSyYtYQMCgYEA2K36LKqJGERbyn8XB3jE
rQz1DFGQCHRbBjZwu/z4g/u/b20ezg/JLFrQ02a0TnvWYgNsKuTXFDJosQmdTqaV
il+isL2XEXEkjX2rBb9QdhUUhf5IXEXyVVRhVVKz39LdXJ4rWcGMGB/Df8giIWG
IhTh9ow97szqy0/yvWYzW3A=
-----END PRIVATE KEY-----
</key>
<cert>
```

Figure 37 - SSH key Recived

With the generation

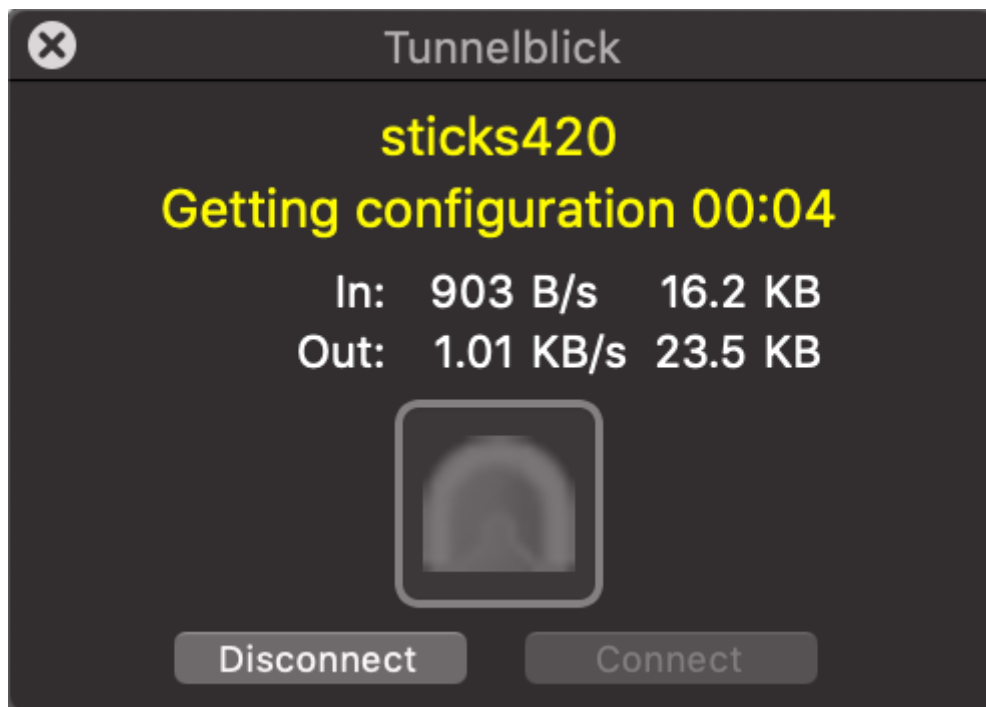


Figure 38 - Connection the server with the ssh key

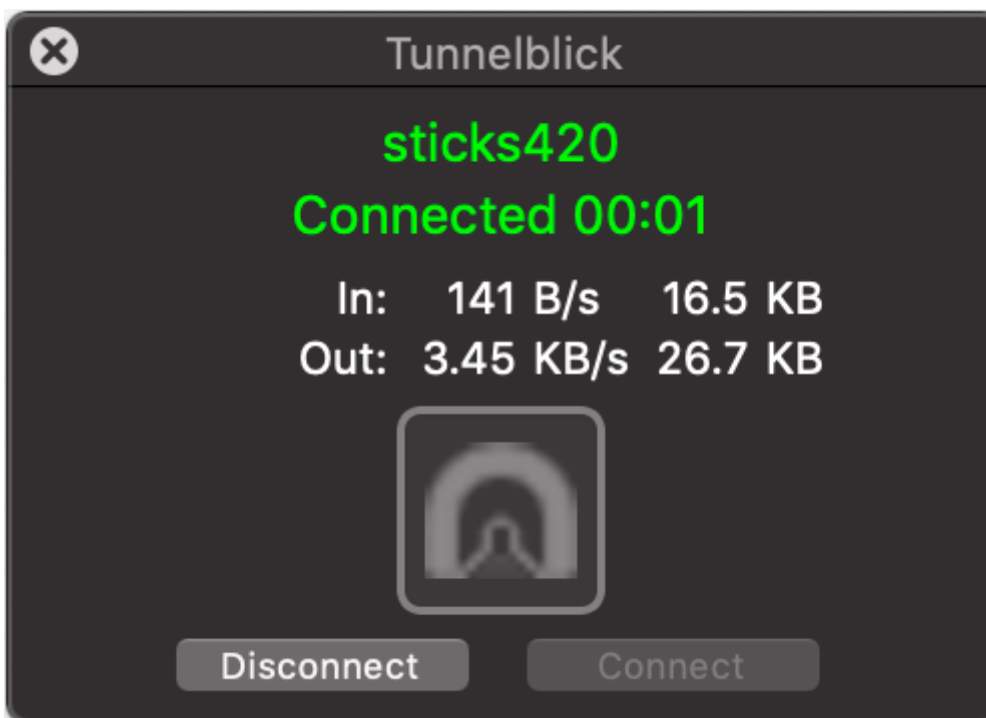


Figure 39 - Connection Successful

```
Last login: Mon May 20 10:41:45 on ttys007
fi→ ~/Desktop/MISC/ITB/Year 3/Group Project/group-project ifconfig tap0
tap0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 2e:f1:ff:73:db:0d
    inet 172.30.0.14 netmask 0xfffffff0 broadcast 172.30.0.15
    media: autoselect
    status: active
    open (pid 54567)
→ ~/Desktop/MISC/ITB/Year 3/Group Project/group-project nmap 172.30.0.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-20 11:33 IST
Nmap done: 256 IP addresses (0 hosts up) scanned in 3.52 seconds
→ ~/Desktop/MISC/ITB/Year 3/Group Project/group-project
```

Figure 40 - SSH Log

5.3.1: Testing SQLI

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/register". The page has a teal header with "Hostile" and "Services" on the left, and "Login Register" on the right. The main content area is light gray and contains two white boxes. The left box is titled "Register" and contains three input fields: "Username" (with a single character entered), "Password" (with a single dot entered), and "Confirm Password" (with a single dot entered). Below these fields is a "Register" button. The right box is titled "Latest logs" and is currently empty.

Figure 41 - Login Form for testing

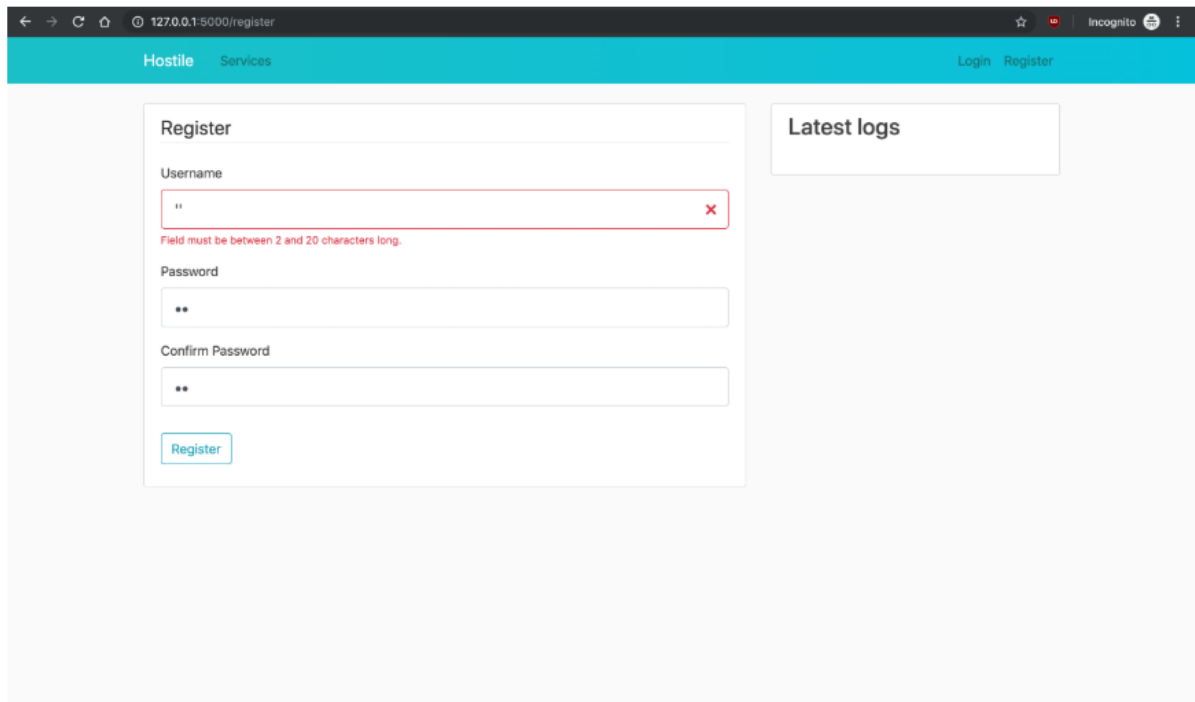


Figure 42 - Attempt at SQLI

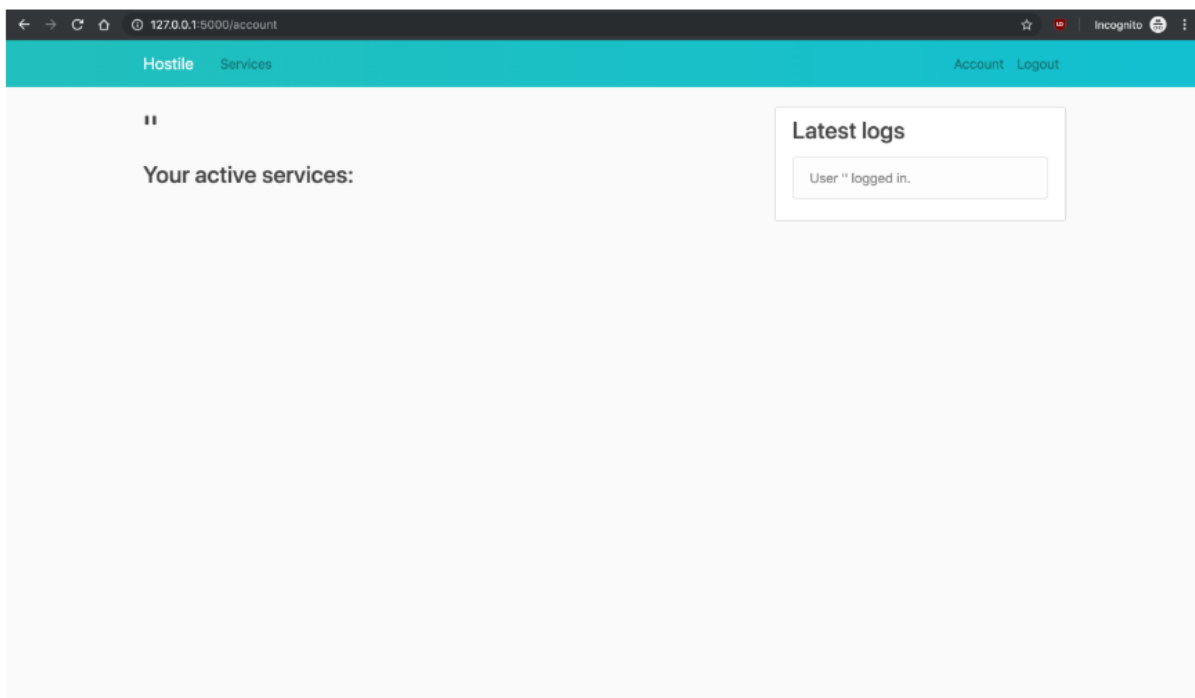


Figure 43 - SQLI Successful

5.4: Results

This section briefly outlines the results of the tests conducted.

5.4.1: Testing SSH

These tests passed without issues. Files could be sent and retrieved, and commands could be run on the server.

5.4.2: Testing a Challenge

Some initial connections failed, but after the author of Naumachia submitted a fix to his GitHub project page, all subsequent connections succeeded.

5.4.3: Testing for SQLi

There were no SQLi vulnerabilities, due to having used the FlaskSQLAlchemy module, which takes care of common issues like these. Queries are not written directly in SQL, and thus the programming layer abstracts the web application from having to deal with this.

5.5: Summary and Analysis

This chapter covered the tests for this project, and the analysis of results. To conclude, the results met the quality requirements and fulfilled the project's security standards.

6: Analysis of Results

Having run several tests against the Hostile web application it was concluded that the software meets the initial goals and criteria. It is functionally competent to perform the administration tasks it is meant to achieve.

7: Conclusion

From the moment the idea was put on the table during a team meeting, the team was intrigued and ecstatic about the idea after having seen cyber ranges before, where all of them had the same issue - they required multiple teams of people working together for months to be able to set up a handful of challenges for a small team of offensive or defensive members.

When the team decided to go ahead with the current project, it was felt that it was best to get people's feedback on whether they would like to see a platform that combines the features of different cyber ranges or cyber security computational events into one central web application, giving users access to everything from a singular point of contact.

From asking this question to multiple industries, levels of knowledge and professions, the team received an overwhelmingly positive feedback from all who were asked, with this came a greater understanding for all who were involved in just what was needed in the industry in regard to training and future development.

When the team first developed the first test network of the cyber range, there was overall excitement but also fear and confusion as to thinking see the amount of details and work that was required to even get just a basic small network off the ground, if from the team who developed the process felt this way then there was an overwhelming feeling that others who even just looked at the functioning of the cyber range would feel the same, and the idea behind the project was to make it lightweight and user friendly and this was the opposite of what the idea was.

Although this was all put to rest after a few group conversations and after researching different systems and technologies the team decided to pivot the idea from a working Cyber Range to a web application that controls multiple Cyber Ranges and allows them to be interconnected across the globe allowing for multiple devices to be connected. This meant that it would only require half of the resources that the first development project idea would. As well as that, it would not only allow for just network based challenges, but also for hosting jeopardy style challenges.

Although there were a few hiccups and moments at which there were some delays and issues in the design and development phase, the team reached as far as it could have with such an ambitiously designed system, and after further discussions at team meetings, there was an overall decision to continue to develop this project a lot further, hopefully to a stage where this project can have it automatically host and generate challenges to be used in

mass team training, changing how colleges and companies both look at educating their future staff and students of the Information Security community.

It is a firm belief of the team that the information security community needs as much open source software contribution as it can get, and this project brings the community a step closer in helping to fulfil this important need. Thanks to open source utilities this worked out better than the team originally envisioned, and now have a useable, understandable software framework that multiple testers believe will fit a niche in the area of cyber security training.

7.1: Future Continuation

Although the projects initial intention was to build a platform that would interact with hardware-based challenges, the team have since pivoted in a direction that would consider being more sophisticated, and vastly more manageable and scalable. Still, the project has retained some of the original goals, most important of which is modularity.

We as a group have decided to continue with the project after; the thesis has been submitted and graded, will continue to work on this project to develop it into its full potential as a cyber-range, hopefully allow it to be eventually be used in Capture the Flag events all over the world.

We already have had interest in this project from other groups who have heard of the Idea through market research this includes multiple companies, law enforcement, Government agencies and even regularly everyday people who just think it would be fun to play with.

We look forward more to after the submission so that the team can get back to working on the web application and future modification of the Naumachia to expand the current working design into the full capability of the Cyber Range.

References

- BBC Technology*. (2011, June 7th). Retrieved April 5, 2019, from BBC.co.uk:
<https://www.bbc.com/news/technology-13807815>
- SDLC V-Model*. (2014). Retrieved from Tutorials Point:
https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm
- SDLC Waterfall Model*. (2014). Retrieved 2019, from Tutorials Point:
https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
- SDLC Spiral Model*. (2015). Retrieved 2019, from Tutorials Point:
https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm
- Boehm, B. (1986). A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4):14-24.
- DARPA. (2015). *National Cyber Range Overview*. Retrieved 2019, from Under Secretary of Defense for Acquisitions, Tech, and Logistics: https://www.acq.osd.mil/dte-trmc/docs/20150224_NCR%20Overview_DistA.pdf
- DARPA. (2016). *The National Cyber Range*. Retrieved 2019, from WhiteHouse Archives: https://obamawhitehouse.archives.gov/files/documents/cyber/DARPA%20-%20NationalCyberRange_FactSheet.pdf
- Fulton, S. M. (n.d.). *TheNewStack*. Retrieved April 2019, from <https://thenewstack.io/finally-linux-containers-really-will-run-windows-linuxkit>
- Graf, N. (2017). *Naumachia*. Retrieved 2019, from GitHub:
<https://github.com/nategraf/Naumachia>
- Graf, V. ". (2017). *Naumachia*. Retrieved 2019, from Github:
<https://github.com/nategraf/Naumachia>
- Grinberg, M. (2018). *Flask web development: developing web applications with Python*. O'Reilly Media, Inc.
- HackerSoc. (2018). *TU-Dublin Ethical Hacker Society*. Retrieved 2019, from HackerSoc: <http://www.hackersoc.com/>
- HomelandSecurity. (2018). *CyberSecurity*. Retrieved 2019, from Department of Homeland Security: <https://www.dhs.gov/topic/cybersecurity>
- IXIA. (2018). *BreakingPoint*. Retrieved 2019, from IXIAcom: <https://www.ixiacom.com/products/network-security-testing-breakingpoint>
- IXIA. (2018). *IXIA*. Retrieved 2019, from <https://www.ixiacom.com>.
- Joly, P. (2017). *Cyber Range 101: A Dangerous Game?* Retrieved 2019, from Quali: <https://www.quali.com/blog/tag/ixia/>
- Magrath, J. D. (n.d.). *A Survey of Cyber Ranges and Testbeds*. Australia.

- Mark Cummins, M. L. (n.d.). *Irish Colleges Cyber Challenge*. Retrieved 2019, from Zerodays.ie: <https://zerodays.ie>
- Python. (2001). *Virtual Environments and Packages*. Retrieved 2019, from Python.org: <https://docs.python.org/3/tutorial/venv.html>
- Ronacher, A. (2010). *Flask is Fun*. Retrieved 2019, from Flask: <http://flask.pocoo.org>
- Rory, E. J. (2018). *Twitter*. Retrieved 2019, from HackTrinityCTF: <https://twitter.com/hacktrinityctf?lang=en>
- Rouse, M. (2017, July). *red teaming*. Retrieved April 2019, from WhatIS: <https://whatis.techtarget.com/definition/red-teaming>
- Royce, W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON*, 26.
- Shields, A. (2014, December 30). *Cyber security presents an opportunity for Symantec*. Retrieved April 2019, from Yahoo Finance: <https://finance.yahoo.com/news/cyber-security-presents-opportunity-symantec-191947768.html>
- Stackoverflow*. (n.d.). Retrieved April 2019, from stackoverflow: <https://stackoverflow.com/questions/42158596/can-windows-containers-be-hosted-on-linux> 07.04.19
- Techopedia. (n.d.). *Free and Open-Source Software (FOSS)*. Retrieved April 2019, from Techopedia: <https://www.techopedia.com/definition/24181/free-and-open-source-software--foss>
- Thomas Gibbons, G. B. (2017). *A Subversive Cyber Range*.