**Diary Entry #1 – Week 2**

**Achieved this week:**

Unfortunately I was ill this week, the first week of project development, and hence was unable to make much progress in the practical side of the project. Some things I did achieve this week include:

* research about the system requirements of both Ubuntu and QEMU, to help decide whether I should dual-boot my laptop or my PC. This research and the knowledge that I am more productive on my laptop than my PC led me to the conclusion I should dual-boot my laptop.
* I also began researching some of the key areas of technology associated with the project. Areas I began looking at included:
  + Ubuntu
  + QEMU
  + QEMU with KVM
  + Honeypots
  + Zero-day attacks

Being totally unfamiliar with both Linux and any kind of system emulation, I began to feel intimidated as I read through the plethora of available online information. I found watching YouTube videos of people explaining the concepts to be much more helpful at this stage than simply reading information from a wall of text; and having gained the foundational knowledge from the videos I found that reading and taking in information from text-based sources became more profitable.

**Diary Entry #2 – Week 3**

**Achieved This Week:**

* Dual booting Ubuntu / Windows 10: this process took longer than expected due to a number of issues that I encountered. I documented my progress through them below.
* Quick course on Linux terminal.

**ISSUE #1: BitLocker Security** – Disabling secure boot meant I was prompted to enter a 48-digit BitLocker recovery code every time I wanted to use Windows.

**SOLVED**: A bit of experimentation led me to the conclusion that I was going to have to temporarily suspend BitLocker and decrypt the drive for the duration of the Ubuntu installation process. I would attempt to re-enable it later.

**ISSUE #2: Ubuntu installer not allowing partitioning** – At the stage of installation where I should have been able to create my partitions for Ubuntu, I was presented with a blank partition table. At first I believed that Ubuntu was failing to recognize Windows, or that Windows was somehow hiding the partition information from Ubuntu.

**SOLVED**:

* *Attempt 1*: Changed UEFI BIOS settings on laptop, following advice from Stack Overflow and Ask Ubuntu.
* *Attempt 2:* Performed Check Disk operation on hard drive and disabled hibernation function.
* *Attempt 3:* Tried to install an earlier version of Ubuntu with the intention of updating post-installation. When trying this, my laptop failed to even recognize my USB as a bootable disk which led to great frustration.

At this stage I became extremely frustrated and had to take a break. I was discouraged that I had encountered an issue so early on in the process that had stumped me. My annoyance reached such heights that I even considered installing Ubuntu as a sole OS for my laptop; however I decided I would return to the issue later with a clearer mind.

* *Successful attempt*: Researched terminal commands to check that Ubuntu was recognizing any of the hard drive partitions, and discovered that Ubuntu wasn’t detecting any of my drive partitions; not just Windows. This led me to an Ask Ubuntu post that advised I changed the SATA mode on my laptop from RAID to AHCI. After this the issue was resolved and I could install Ubuntu normally.

Upon successfully overcoming the issue I felt a wave of relief, as well as encouragement. Although slightly annoyed that the process had taken so long (around two days) I was glad that I could now progress further with the project.

Another note I made from this process was a feeling of discomfort I got when using the terminal to discover the drives, as I was simply copying what I read on the screen without understanding it.

**ISSUE #3: Wi-Fi** – Having successfully installed Ubuntu, it was failing to detect the Wi-Fi adapter on my laptop. Some research showed that it was likely because there were no drivers for the adapter present. One suggested solution from an Ask Ubuntu page was to simply reinstall the OS, as there is a check box to automatically install these drivers for you in the installation process (i.e., had I been paying more attention during the installation process, I could have checked it and avoided this issue altogether).

*Successful solution*: I swapped to the Windows side to discover the exact specification of my Wi-Fi adapter, and then googled how to install drivers for it on Ubuntu. I used Bluetooth tethering to my phone to access the internet on the Ubuntu side. Again I felt the aforementioned discomfort as I followed the terminal commands for the installation. I decided I should find a quick course on the basics of Linux terminal.

I found a quick free course on Udemy, and went back and reviewed the commands I used previously to better understand them. This made me feel more comfortable with using the terminal.

This week I also watched numerous more videos and read more articles on QEMU, KVM and virtualization in general. I am beginning to feel more comfortable with these topics, however the networking portion of the project is still a daunting prospect to me.

**Diary Entry #3 – Week 4**

**Brain Dump**

This week I felt particularly motivated to get QEMU up-and-running, encouraged by success in the previous week. I decided to go to the library to see if I would work better there, and discovered that my productivity was much higher there than in my house. I was pleased that the installation of QEMU and starting of my first VMs went largely without hitch; however I felt that most of the tutorials lacked in one area or another, and found myself having to use a combination of them to get good results. Whilst my initial plan was simply to do a How-To guide on networking with QEMU, I decided I should include the installation process with a combination of all the information I gathered from different tutorials, as I think this will be helpful to others who were in my boat.

When I started doing some research into networking with QEMU, I became somewhat intimidated by the information available, as much of it seemed to conflict with itself (which I soon discovered was due to outdated or deprecated documentation still being available, even from QEMU’s own website!). It was during the process of trying to get networking working that I encountered a major problem. Any online documentation I found was EITHER for User-Mode Networking OR TAP device networking, and nigh on NONE of it stated outright that TAP bridging is incompatible with wireless host NICs (something I thought would be very important to mention). This lack of information meant I spent a few days digging in the completely wrong place to find a solution to my networking problem, and I became very frustrated. Due to the fact that I didn’t know about the wireless incompatibility, I thought for certain that I was doing everything right and it just wasn’t working, and I became quite discouraged. Furthermore, this area of the networking was a particularly niche area that I had only really delved into as I recognized the lack of good documentation and tutorials, and wanted to create one to help – it wasn’t necessarily particularly relevant to the project. This led me to include right from the offset of the networking part of my tutorial that there were three different sections to networking; hopefully this will save someone else the time it cost me.

However once I encountered some documentation that mentioned the fact that wireless NICs aren’t compatible with bridging, things started to look up. I spent another few days searching through how to use alternative methods, and again failed to find one source that included *all* the information I needed; I had to alter the steps in many processes to try and suit VMs rather than linking real-life hardware. I was somewhat glad of this discovery however, as once I got it working I knew that the How-To guide I created was going to be valuable as I could yet again combine the information I’d found into one coherent source.

**Diary Entry #5 – Week 6**

**Brain Dump**

**Monday:**

Have mostly recovered from illness, so motivation levels are good. However I have a project due for another module at the end of this week, and am expecting my attention to be diverted mainly towards it.

Began exploring the real-world use-cases of VMs connected to VLANs. Took a look at the Argos tool, and did some research into Honeypots and other forms of network security. Found the concept of a honeypot very interesting, and came across the concept of high-interaction honeypots that make use of multiple virtual machines. Also came across the concept of deception technology, that I will look into later.

**Wednesday:**

Attempted to get the Argos tool running on my machine, but encountered an issue with setting up the VLAN. Unfortunately it is not working how I expected it to, and will need to spend some time getting that part up-and-running before attempting to go any further.

Tried following some tutorials and walkthroughs online, but to no avail as of yet. I also realized that I will need to edit my tutorial, as this area is covered within it and it may not work for others as they expect either.

Was unable to get it working today.

**Thursday:**

Tried again for a little while to get the virtual network running as expected in preparation for using Argos, and again was unable to get it working. I began to get quite frustrated again, as it seems as though I am doing everything right but it simply won’t work. I had a feeling it was to do with the assigning of IP addresses, so I did some more research into that area however I will need to do some more in order to better understand it.

Doing work for my other project took up the bulk of my working day today.

**Diary Entry #6 – Week 7**

**Brain Dump**

**Tuesday:**

Today I attempted to get the private isolated network running (in order to make progress towards the project goal). I attemtped using a network bridge and TAP devices in order to connect the devices together, without connecting them to the actual network. Whilst I managed to get the devices successfully connected to the bridge, they failed to communicate with one another across the virtual network. I tried looking through and following the tutorials of many websites, but to no avail.

**Wednesday:**

Today I made another attempt to get the private isolated network functioning. Unfortunately however, after following a certain online tutorial the OS on my laptop began to fail to recognise the presence of a network card on my laptop. This was an issue that I had encountered at the beginning of the project process, and so I was able to go back to my documentation and fix the issue with few problems, however to have such a strange problem at this stage was quite discouraging. Tomorrow I will continue to attempt to get the isolated network to work.

**Thursday:**

Today I am very relieved to say that I managed to get the private isolated networking functioning properly. I tried a few more last-ditched attempts at self-configuring the network through the terminal, however discovered that some GUI tools would provide the correct networking configuration tools for you. I installed software known as Virt-Manager, which allows the handling of VMs with a GUI front-end. Using Virt-Mangager and following a few YouTube tutorials, I was able to get the private isolated network running. I intend to look into the source code of Virt-Manager in order to understand the inner-workings of what goes on with its network wizard, so that at a later date I could perhaps attempt to make it work manually again.

Having got the private isolated network functioning and tested, I have begun updating my How-To guide with the appropriate steps to follow, so that others may have an easier time of completing this process than I did.

**Diary Entry #7 – Week 8**

**Brain Dump**

**Monday:**

Wrote initial script for blog post, and began working on new sections for my how-to guide. The new section will be about isolated networking with QEMU, as mentioned in last week’s diary entry I found it very difficult to find useful tutorials and walkthroughs in this area – I feel that a more clear and complete How-To guide will be useful to other people who wish to undertake this task in the future.

**Wednesday:**

Completed the blog post write-up. I have written the blog post with the intended audience of someone like myself who is completely new to the whole area of virtualization (and even Linux). The post is written to be light and easy-to-read, in an attempt to contrast the dull and official nature of reading ordinary walkthroughs and documentation. The post is also not written to be a stand-alone walkthrough – readers will be directed towards my How-To guide(s) to get the technical information that is being discussed in the post; this helps improve the posts readability as it is therefore not filled with technical jargon. The main purpose of the post is to help someone who may wish to advance my project to get caught up on how I reached the stage I am at, and I will continue to update it as the weeks progress.

**Thursday:**

Converted the blog post from its’ word format to a singular HTML document for the purposes of submission. I followed the previously provided walkthrough on how to make an HTML file to complete this process.

**Diary Entry #9 – Week 10**

**Brain Dump**

**Monday/Tuesday: Bank Holiday**

**Wednesday:**

Began work on updating the script for the section in my How-To guide regarding the private networking. I began looking into the source code of the Virt-Manager software, however I found it quite daunting to observe as studying others’ code is not something that I am familiar with. Information regarding tips and tricks for helping people to read other delevopers’ code appears to be scarce – however I understand this as it is quite a subjective activity and there is likely no simple way to do it for everyone. Perhaps I could offer some advice on my blog / social media post to those looking into source code?

**Thursday:**

Finalised the updates to the HTML for the section in my How-To guide. I am now realising that the most useful things I can likely achieve are to update my blog post and How-To guides, as delving into source code is proving to be a more complicated process than anticipated. These things will be of the most possible use to others. Created an Instagram account for the project @netulate\_project.

**MARKET ANALYSIS**

I started off my market analysis with a simple google search to try and identify existing system for emulating (rather than simulating) whole networks. The two most prominent pieces of tech that I discovered were the [Common Open Research Emulator (CORE)](https://www.nrl.navy.mil/itd/ncs/products/core) and the [Extendable Mobile Ad-hoc Network Emulator (EMANE);](https://www.nrl.navy.mil/itd/ncs/products/emane) both open-source projects managed by the US Naval Research Laboratory. However, these two pieces of software, whilst powerful, were not designed to be used on a quick, plug-and-play daily basis of a system administrator in a smaller environment.

This was an area of the market that I found to be fairly devoid of competition, as most of this software was designed for high-scale operation the market for software for “regular working folks” is definitely not as thoroughly explored. Therefore, I decided the aim of the project should relate more to those people; system administrators in small-to-medium sized businesses, schools, colleges and other institutions.

In order to reach people in this category, I turned to some popular subreddits that revolve around people working or hobbyists in this area; [r/sysadmin](https://pf.reddit.com/r/sysadmin/) and [r/homelab](https://pf.reddit.com/r/homelab/), to name two. A social media post to either of these subreddits would directly address the target recipients of the final version of the system, and good feedback could come on what areas of the software would be most vital. Therefore, I have written my social media post (linked [here](https://github.com/bconnor270/ComputerScienceChallenges/blob/master/40260011_RedditPost.docx)) in such a way to inspire readers to see how powerful the final piece of software could really be, and to encourage them to give feedback and perhaps even become involved in development.

However, since I have not made much technical progress with the project, the other content I have produced such as the Blog Post and How-To guide have a different audience to that of the final version of the software. This is because although my initial audience was system administrators, once I began working on the project I observed that the need for more help and documentation at the beginner’s end of the scale was vast – and so the other mentioned areas of my work are focused on them.

**POTENTIAL TESTING STANDARDS**

As I have not made much technical progress towards the commercial end-goal of the project, the information in this section can be counted as little more than speculation as to what standards it would have to meet. With that in mind, I shall discuss a few of the things I feel would be necessary to make this a commercial standard system.

One main area I would have concern for is accuracy. Should the system become complete to the standards discussed, it could well become a depended-upon program for many establishments. This means that the accuracy of the system in two main areas needs to be calculated. Firstly, the systems accuracy in scanning and replicating the user’s network. This is likely going to be a fairly low accuracy, as network scanning alone cannot attain enough information to emulate a whole workstation. A feature would certainly have to be included to allow network admins to “fill in the blanks” with regards to missing information about the network. The second area of accuracy that is arguable even more critical is that of tests on the VLAN vs. tests on a real network. A vast number of comparison scenarios would need to be carried out on multiple different real-world test subjects in order to verify that the system was producing accurate enough results to be considered for use in commercial environments.

Another area of the project that I feel is important is usability. As mentioned before, I started the project with ordinary system administrators in mind – users should not have to go and spend time researching emulation or any other areas outside of what they need to know for their current occupation. This means that the system needs to be reasonably user-friendly, and built with the consideration that people who are perhaps not accustomed to emulation are going potentially going to become dependent on this software. That said, usability is largely a subjective issue and therefore the only potential testing that can be done is issuing the system to its potential target audience pre-release and asking users to complete a survey on their experience. This kind of testing would also be valuable in regards to potentially adding or removing features that real-life users would like to see in the system.