BUILDING AND DEPLOYING A TYPE 2 HYPERVISOR

Network Operating Systems and Services – Assignment 1

Danny Thomas (2004365)

Contents

Hypervisors 2
Type 1 2
Architecture 2
Security and Features 2
Scalability, Hardware Requirements and Use Cases 2
Type 2 3
Architecture 3
Security and Features 3
Scalability, Hardware Requirements and Use Cases 4
Hypervisor choice 4
IP addressing 5
Setting up the network 6
Testing and Troubleshooting 7
Troubleshooting Firewall 7
Pinging on the Network 8
Bibliography 12
Appendix 13

Hypervisors

A hypervisor is software that is used to manage virtual machines in relation to networking concepts and virtualisation, in this report we will cover type 1 and type 2 hypervisors as they are the commonly used architectures [1].

Type 1

Architecture

The first hypervisor we're going to discuss is Type 1, also known as a "bare-metal" hypervisor; this hypervisor sits directly on top of the physical server and acts as an operating system as seen in Figure 1, communicating with the hardware directly instead of having another operating system between it and the hardware resources; this has a few effects on the security, reliability and efficiency of a type 1 [2].

Security and Features

Any additional operating system or device will always influence security, therefore not having a middleman between the hypervisor and the hardware already decreases the chance of an attack to happen on the hardware operating system. Security in hypervisors is also inherently good because everything is compartmentalised meaning if it is set up correctly and intelligently there won't be any effect on the end user. This is because type 1 has mobility and resource allocation features that ensure servers can stay active if the resources are available, [3] for example if a virtual machine is being overloaded and requires more CPU power than the current hypervisor is able to give, the hypervisor can detect this and move it on to a different server with more resources available, this is seamless and wouldn't have any effect on the services.

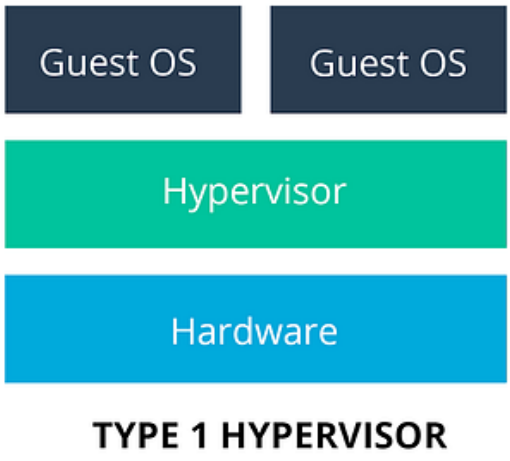

Figure 1: Type 1 Hypervisor Architecture [9]

Another aspect of resource management in type 1 hypervisors is that you can set up your virtual machines to be allocated with more CPU cores, RAM or storage than is currently available on the hardware, this is because the virtual machines only take the resources that they need automatically, unlike type 2 which require you to allocate hardware to a virtual machine and have those resources limited to only that virtual machine.

Scalability, Hardware Requirements and Use Cases

Type 1 hypervisors are easily scalable because of the aforementioned resource management features, because the only thing required is to install more or better hardware into the machine the hypervisor is running on, or to create another hardware environment for another hypervisor to create a backup system for the running virtual machines.

Hardware requirements for hypervisors are always dependent on what they are being used for, type 1 is intended for heavy use because of its scalability and features, and requiring a dedicated machine means that enterprises benefit from having dedicated servers and services running meaning it can support dozens to hundreds of virtual machine environments.

On the Microsoft website they list some requirements to setting up their hypervisor software “Hyper-V" [4] which lists 4GB RAM minimum, and a 64-bit processor which is VMM supported, they also note that "you can run 3 or 4 basic virtual machines on a host that has 4GB of RAM, though you'll need more resources for more virtual machines. On the other end of the spectrum, you can also create large virtual machines with 32 processors and 512GB RAM, depending on your physical hardware.” Which outlies a recommendation for more virtual machines.

Alternatively, on the VMware website [5] there is recommendation for the KVM hypervisor software that gives a formula for how many CPU cores and GB of RAM is required for each virtual machine running at peak performance on the hypervisor which states 10 cores per virtual machine with 16GB RAM allocated or 32GB RAM when certificate-based authentication is enabled.

Type 2

Architecture

The type 2 hypervisor works significantly different from the type 1, despite the diagram showing only one difference, it has many effects on the hypervisors functionality. The type 2 hypervisor can be installed on another operating system like Windows 10 or Linux and runs as an application instead of dealing with the hardware resources directly like type 1, see Figure 2 [1]. A downside of this type is that the hardware has to support the hypervisor, its virtual machines and its own operating system, taking up a lot of the resources before the virtual machines are turned on.

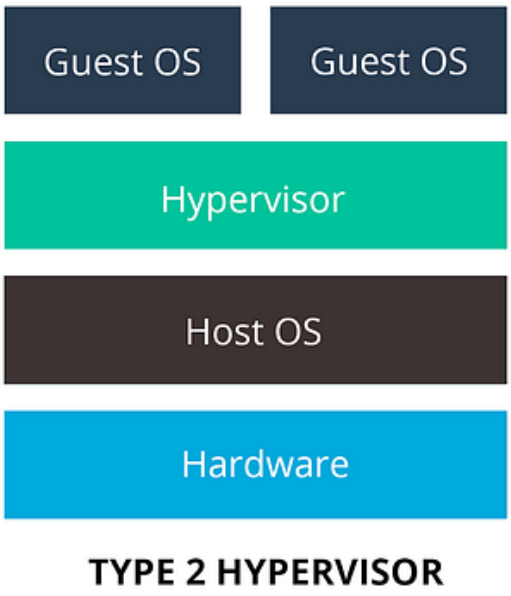

Figure 2: Type 2 Hypervisor [9]

Security and Features

Security on the type 2 hypervsior is dependent on the operating system and the environment its in, because it is running on another operating system it is only as secure as the operating system itself, therefore it is open to the same vulnerbalities. For example, if it was running on an old version of Windows then it would be vulnerable to attacks that target old versions [2].

Because of type 2’s nature, it can be installed on common operating systems making it accessible to most people and easy to test and try, although limited to the machines hardware which wouldn’t be

comparable to type 1's machines built specially for virtual machine environments, its important to note that type 2 hypervisors doesn't have peak performace as their main goal, instead it acts as an easily deployable and managable environment [6].

A useful feature of type 2 hypervisors is that you can install any operating system on the virtual machines, meaning if you were running a Windows machine you can install Linux on a virtual machine and get access to its resources and vice versa [3].

Scalability, Hardware Requirements and Use Cases

Type 2 hypervisors are inherently not designed for scalability, they are meant to be quick and easy and allow anyone to create a virtual environment on their personal computer, for example for this assignment, 2 cores each were allocated to the four virtual machines created and they were still able to run without issues if the cores and RAM were available; making type 2 not scalable and dependant on the machines hardware. Type 2 hypervisors don't have the same features as type 1 including seamlessly allocating hardware resources, meaning once you allocate an amount of cores or RAM, the virtual machine will take up all those resources and not share them until it is shut down.

In the Virtual Box official documentation which can be found at (https://www.virtualbox.org/wiki/End-user_documentation) it requires at least 512MB of RAM but recommends more than 2GB and any recent Intel or AMD processor to run a virtual environment comfortably.

Oracle server also have documentation that recommend 1GB RAM and a 64-bit processor that can run at 1.3 GHz with 2 cores. The difference in hardware requirements is monumental between type 1 and type 2 showing how much less type 2 is intended to do versus type 1 which is used by enterprises.

Hypervisor choice

Given the resources at hand and the scope of the topology, a type 2 hypervisor would be the best choice for the scenario, only 4 virtual machines are required, 2 PC's and 2 servers, which can be handled on the University PC's. A type 1 would be overkill for this network and a dedicated machine is not available to us, type 2 hypervisors are primarily used for testing and educational purposes, so it is also a perfect fit for our scenario and learning about virtual machines and how they work and interact with hardware.

IP addressing

In its most basic form, IP addresses are used to uniquely identify devices within a network, but that is only one of their functions. IP addresses as a protocol also uses TCP to make sure that packets are sent efficiently and accurately,

Setting up the network

When looking at the topology its important to decide how many network cards are required, because these make up your networks and allow the engineer to decide your IP addresses and lay out the network. For the topology, each virtual machine required 2 network cards, therefore 2 were allocated to each as can be seen under the “Network” tab in figure 3.

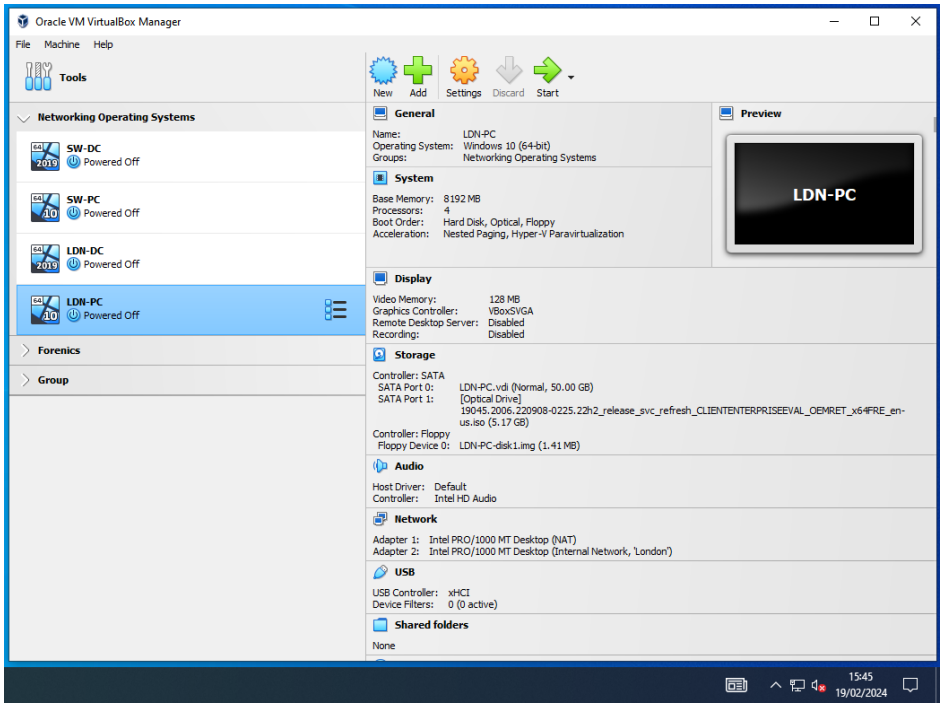

Figure 3: London PC configuration

After the network card is configured, it is important to follow the given topology and use the correct IP addresses, for the London PC the IP was “10.2.0.2”, Figure 4 shows an IPv4 properties window where IP addresses are configured.

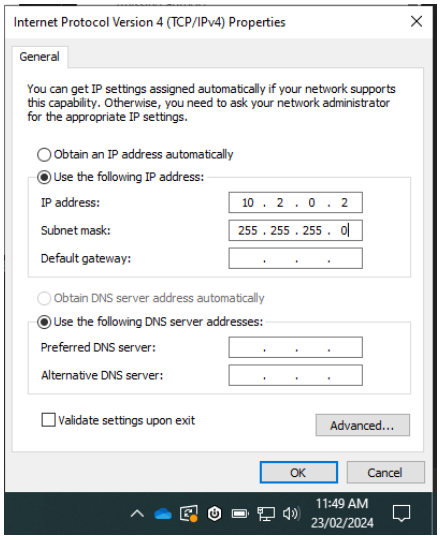

Figure 4: London PC IPv4 properties EXAMPLE

This must be repeated for each PC and Server in the topology, allocate two network cards, and give the correct IP address noted in the topology.

Because the IP addresses on the topology have been done correctly, once the PC’s and servers all have the correct IP addresses the network should work and be able to ping each other.

Its important to follow the given topology when creating a network because a lot of thought has been put into it before it reaches the engineer who creates it, therefore I chose not to subnet the network and to follow the topology “as per the requirement” although it would have been beneficial to subnet the network to make it more organised and plan ahead if the network did decide to expand in the future of NOSS company.

Testing and Troubleshooting

Troubleshooting Firewall

When attempting to ping the relevant server from SW-PC or LDN-PC, the pings were unsuccessful, after verifying that the IP addresses have been configured correctly, I did some research and found that the firewall blocks pings on a network automatically. This is likely set up to stop Denial of Service attacks as a server could be barraged with pings to fill the network traffic and slow down a network (nicknamed ping of death [7]), therefore, to get around this issue I disabled the firewalls entirely on the PC (Figure 5) and the Server (Figure 6).

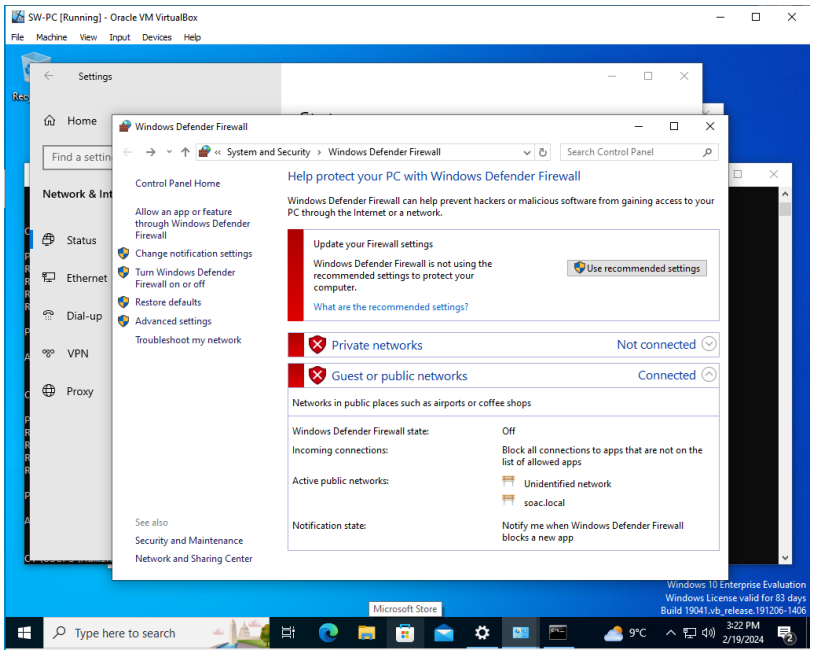

Figure 5: PC Firewall disabled

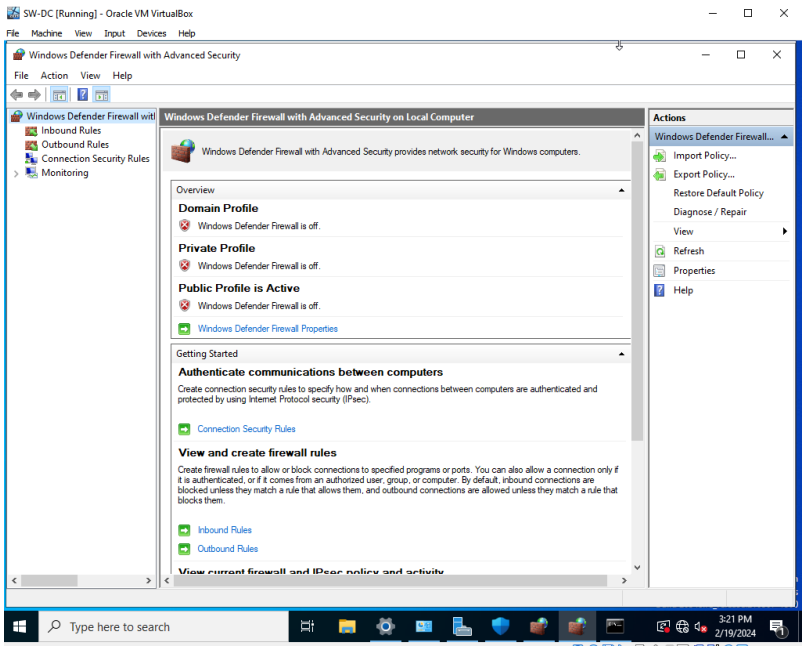

Figure 6: Server Firewall disabled

Because this network is being set up in a virtual environment it is easier to test with the firewalls down therefore keeping them down won't be an issue, in a scenario where the PC's must be connected to the internet the firewalls would have to stay up to protect the network, in this case it would be wise to change the firewall rules to allow pings. This can be done in the firewall settings by enabling the echo request rule [8], see Figure 7.

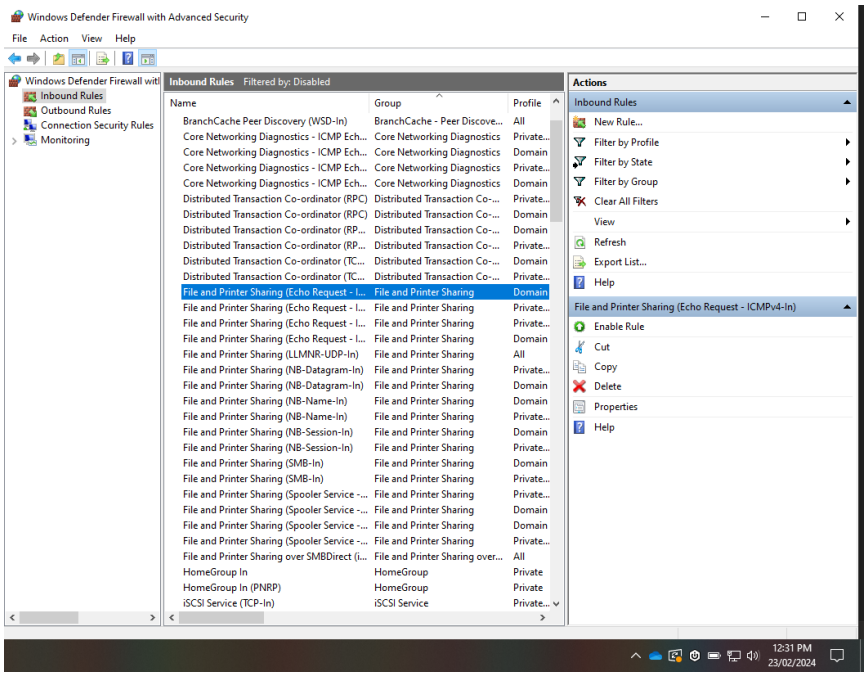

Figure 7: Echo request enabling

Pinging on the Network

After disabling the firewall, the network was able to ping successfully across the board, meaning the London Server could ping the London PC and vice versa, the Swansea Server could ping the Swansea PC and vice versa, and the London Server and the Swansea Server could ping each other (all figures below), creating a working network. In the figures there are 2 successful pings shown off to ensure reliability, and 0% loss is shown in each showing a 100% success rate in packet transfer.

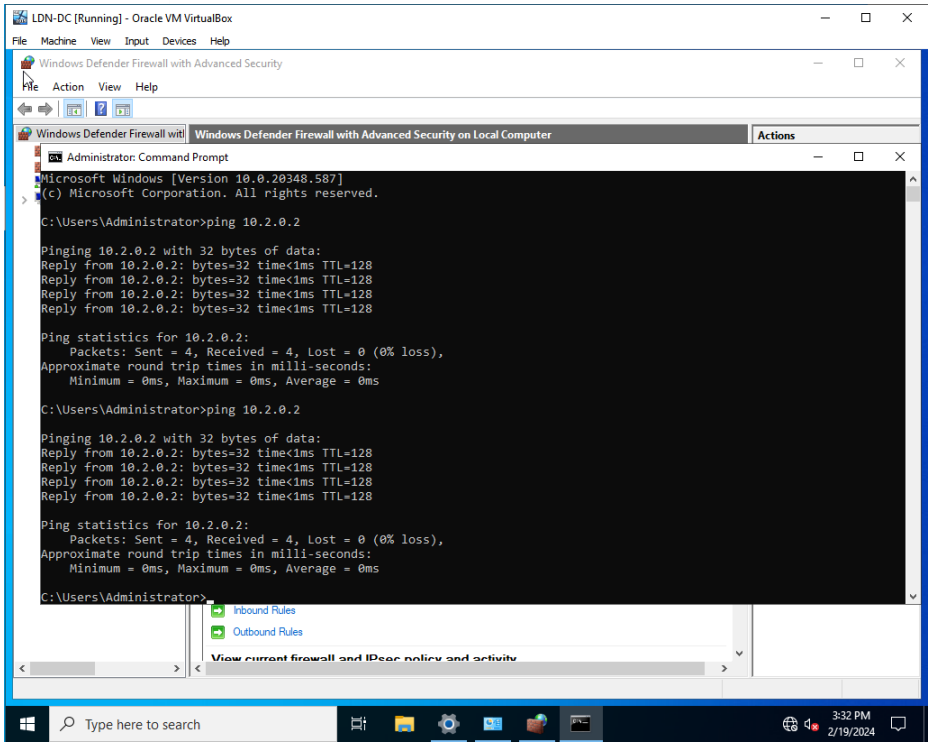

Figure 8: London Server pinging London PC

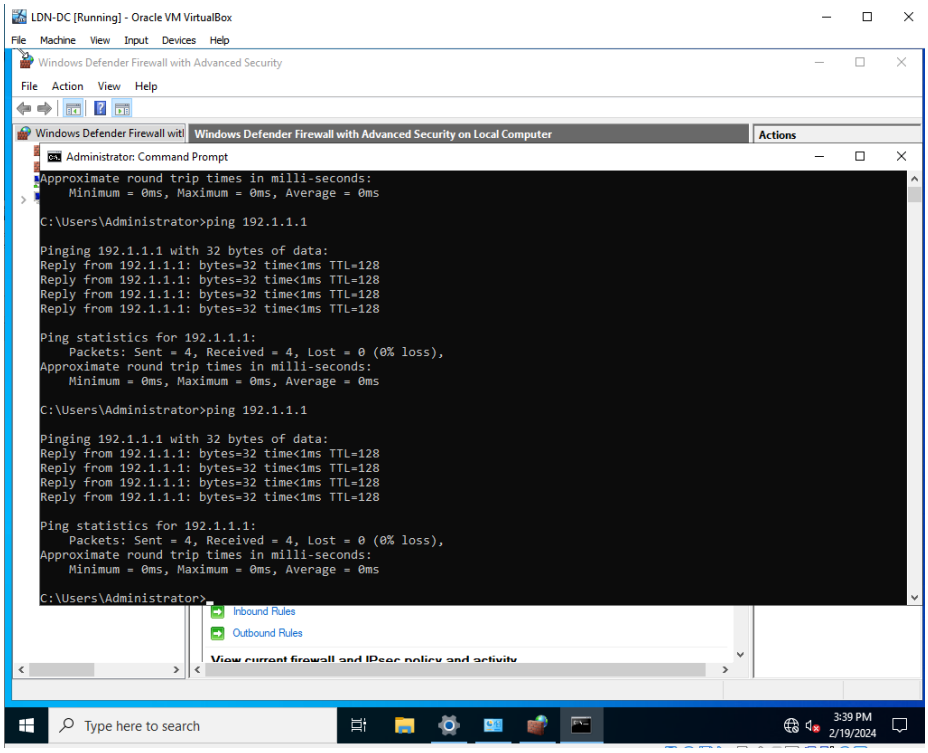

Figure 9: London Server pinging Swansea Server

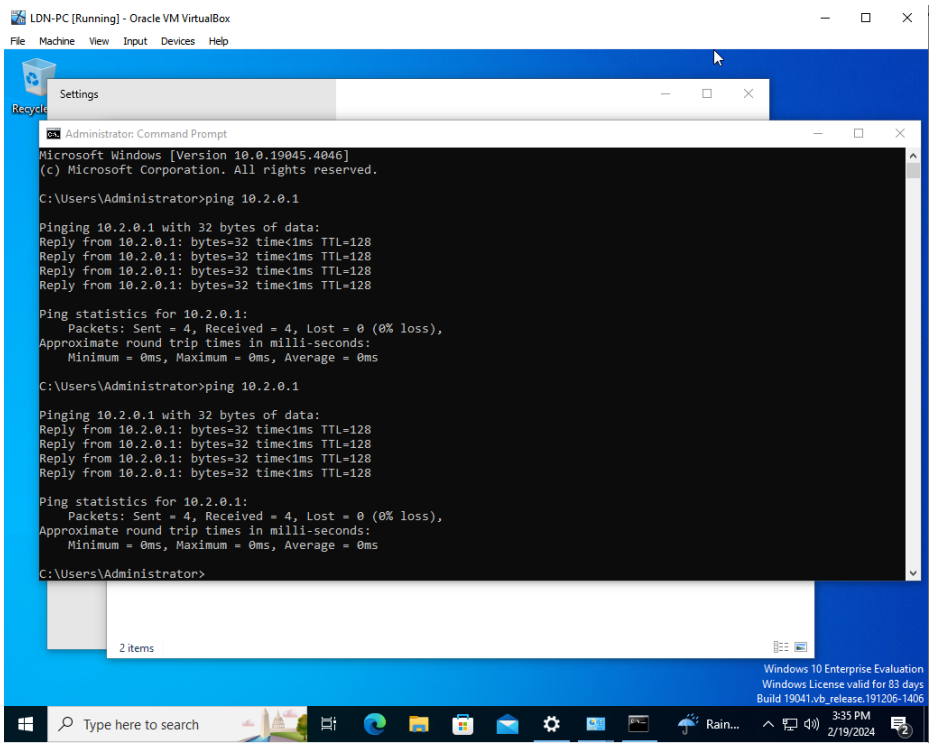

Figure 10: London PC pinging London Server

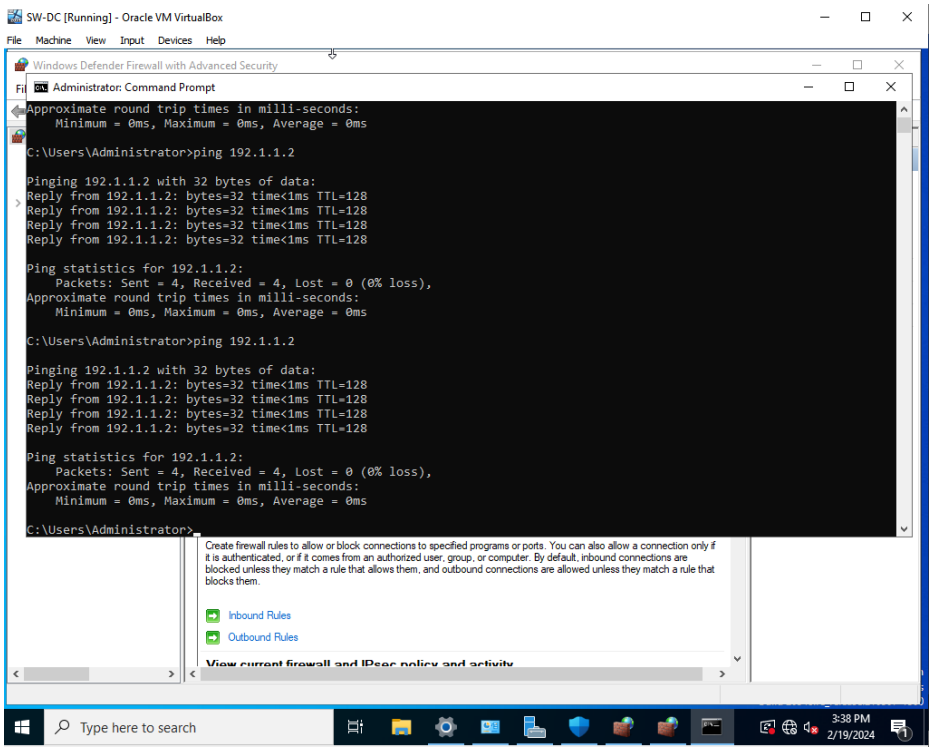

Figure 11: Swansea Server pinging London Server

0

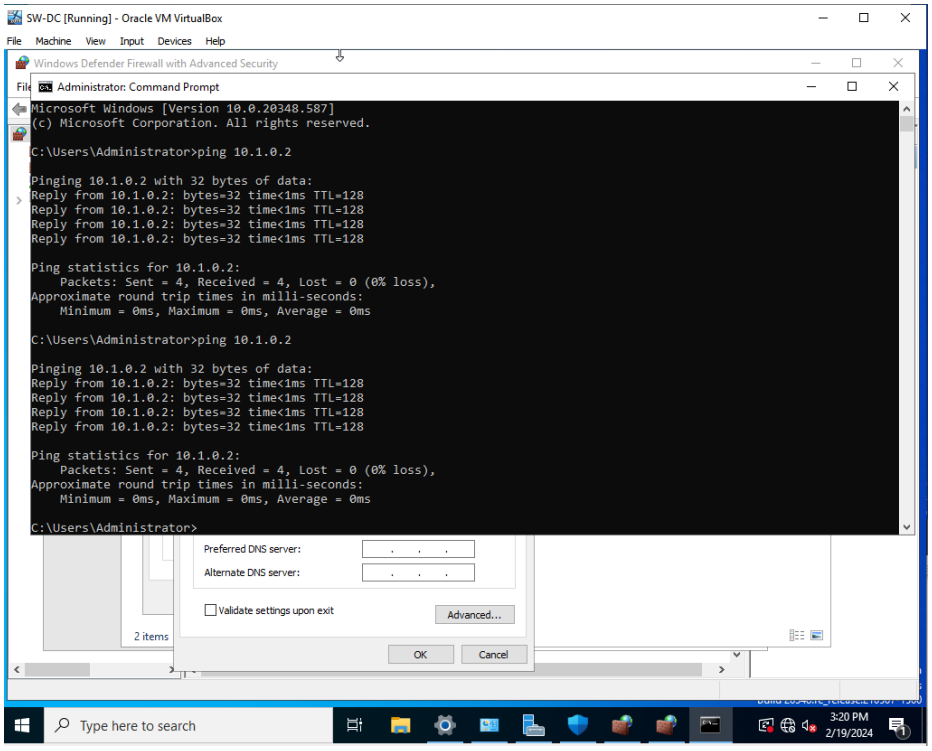

Figure 12: Swansea Server pinging Swansea PC

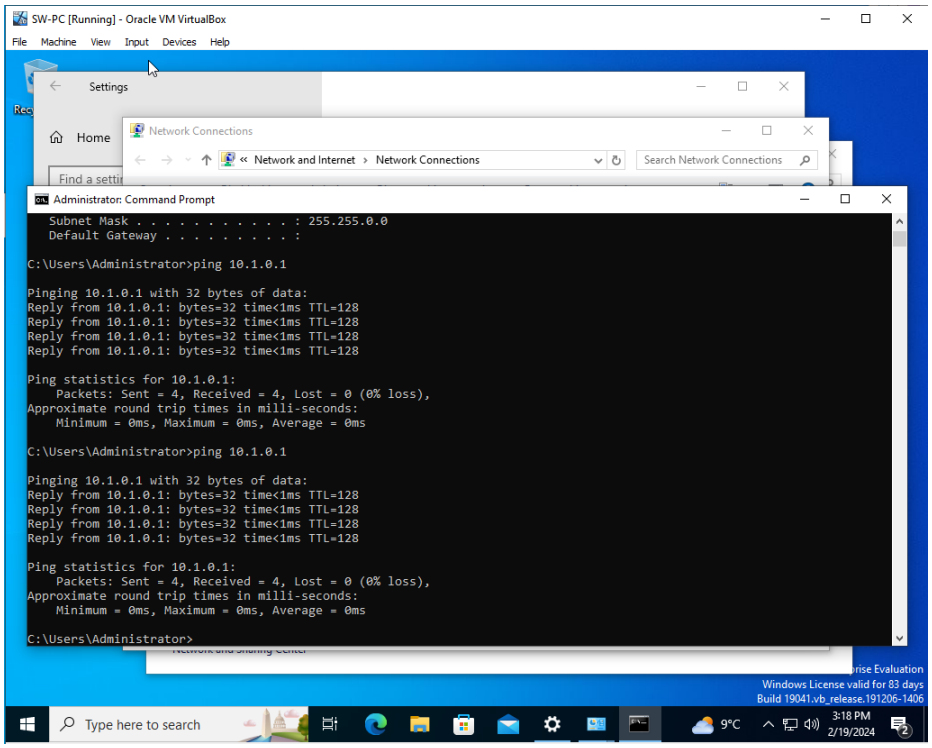

Figure 13: Swansea PC pinging Swansea Server

Bibliography

References

[1] A. Blenk, A. Basta, M. Reisslein and W. Kellerer. A. Blenk, A. Basta, M. Reisslein and W. Kellerer. Survey on network virtualization hypervisors for software defined networking. Survey on network virtualization hypervisors for software defined networking. *Comst Comst 1818(1)*, pp. 655-685. pp. 655-685. 2016. . DOI: 10.1109/COMST.2015.2489183.

[2] M. Portnoy. M. Portnoy. *Virtualization Essentials Virtualization Essentials* (1st ed.) 2012.

[3] S. Simic. (-09-29T01:56:00+00:00). *What is a Hypervisor? Types of Hypervisors Explained (1 & 2)*. Available: https://phoenixnap.com/kb/what-is-hypervisor-type-1-2.

[4] Anonymous (). *Windows 10 Hyper-V System Requirements*. Available: https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/reference/hyper-v-requirements.

[5] Anonymous (). *Minimum Hypervisor Hardware Requirements*. Available: https://docs.vmware.com/en/VMware-SD-WAN/5.4/vmware-sd-wan-partner-guide/GUID-17432A7A-693D-477E-8D7F-1D74673072AA.html.

[6] A. S. Tanenbaum and H. Bos. A. S. Tanenbaum and H. Bos. *Modern Operating Systems Modern Operating Systems* 2014.

[7] Anonymous (). *What is a Denial-of-Service Attack?*. Available: https://www.techtarget.com/searchsecurity/definition/denial-of-service.

[8] Anonymous Anonymous Configure the windows firewall to allow pings. Configure the windows firewall to allow pings. Available: https://kb.iu.edu/d/aopy.

[9] Anonymous Anonymous Type 1 and type 2 hypervisors: What makes them different | by ResellerClub | ResellerClub | medium. Type 1 and type 2 hypervisors: What makes them different | by ResellerClub | ResellerClub | medium. Available: https://medium.com/teamresellerclub/type-1-and-type-2-hypervisors-what-makes-them-different-6a1755d6ae2c.

Appendix

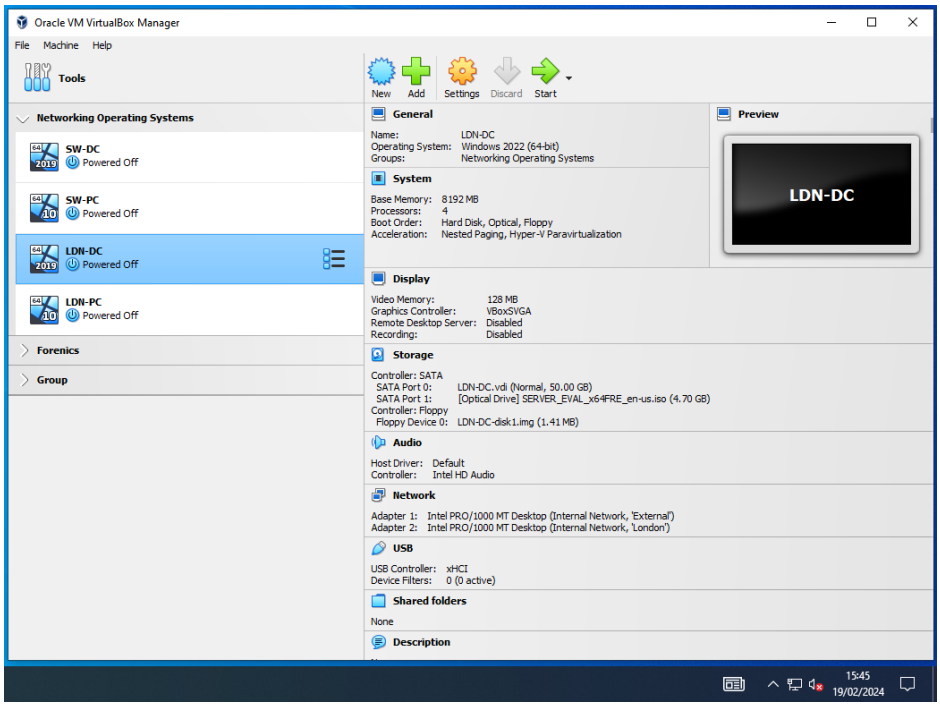

Figure 14: London Server configuration

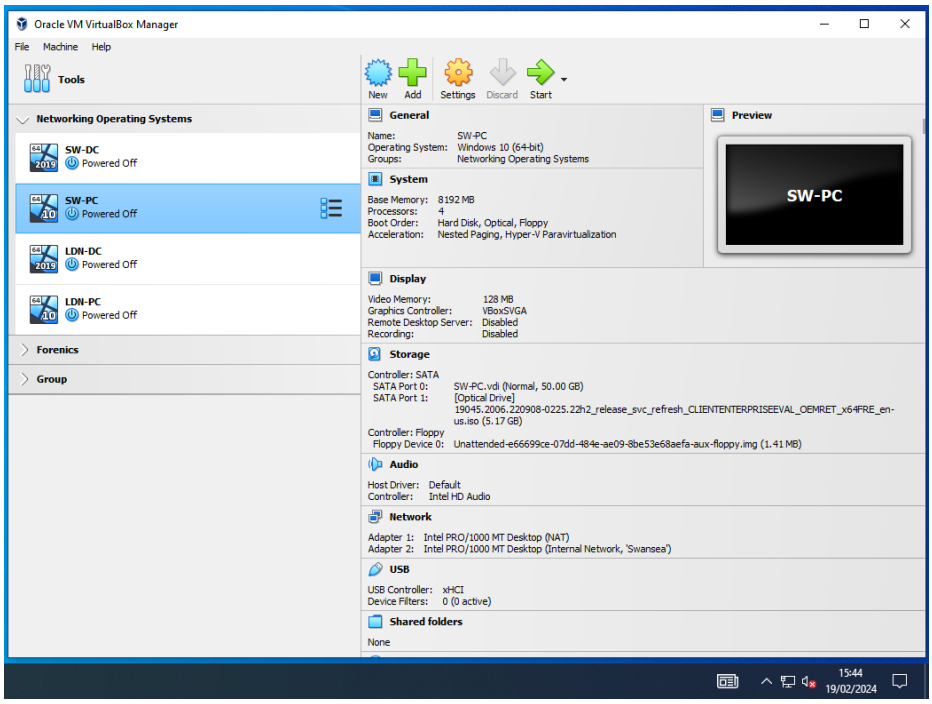

Figure 15: Swansea PC configuration

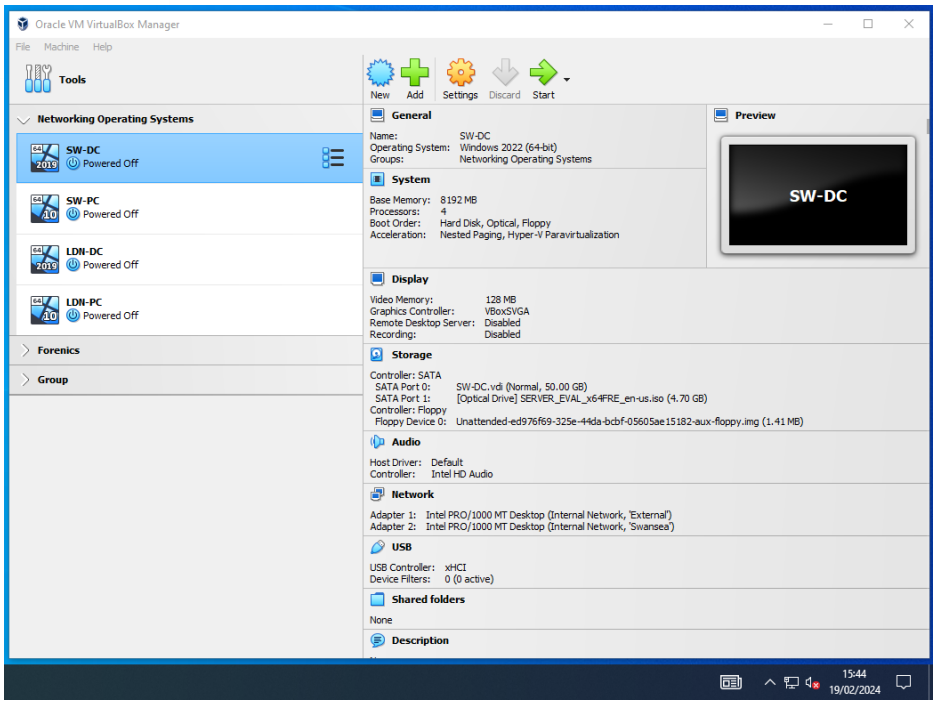

Figure 16: Swansea Server configuration