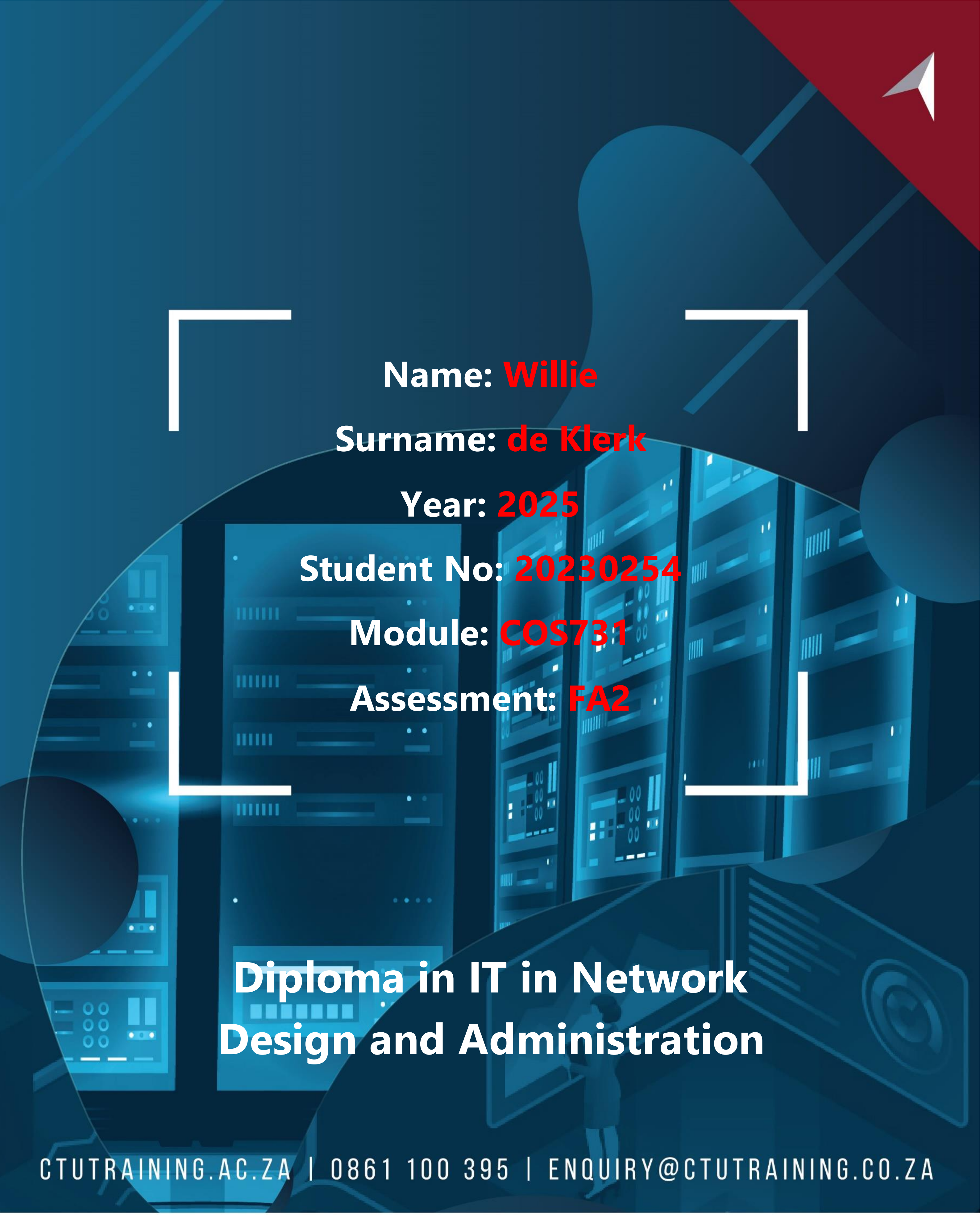


Name: Willie

Surname: de Klerk

Year: 2025

Student No: 20230254

Module: COS731

Assessment: FA2

Diploma in IT in Network Design and Administration

CTUTRAINING.AC.ZA | 0861 100 395 | ENQUIRY@CTUTRAINING.CO.ZA

CTU training solutions

Private
Tertiary
Education
Provider

1. Declaration of Authenticity

A critical aspect of any assignment is *authenticity*. Because you are completing much of the work for the assignments *unsupervised*, the examiner must be convinced that it is all your work. For this reason, you must complete the *Declaration of Authenticity* provided in the study guide and have it counter-signed by your manager, mentor, or lecturer.

NB	The declaration of authenticity is a legal document, and if found that you have made a false declaration, then not only will your results be declared null and void, but you could also have criminal charges brought against you. It is not worth taking the risk!

s

Please complete the declaration of authenticity below for all assignments:

DECLARATION OF AUTHENTICITY

I _____(Willie de Klerk)___________________________________ hereby

declare that the contents of this assignment are entirely my work except for the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	**Date**
Formative Assessment [2]	**2025/05/01**

2025/05/01

X Willie

20230254
Student
Signed by: 7e38ac48-8902-42d6-b32e-8740bcbddb0d

Signature: ___________ _____________ **Date:**
2025/05/01___________

Contents

2. Introduction

In this research report, I have demonstrated my understanding of Linux system management, showcasing my ability to effectively administer and maintain Linux environments. I illustrated my skills in securing Linux systems, applying best practices to ensure confidentiality, integrity, and availability is maintained. I demonstrated my understanding of troubleshooting Linux systems.

The questions presented to me to produce my report include:

A. Your organization has recently experienced a security breach, and you have been tasked with securing the Linux servers to prevent future incidents. Outline the security measures you would implement to protect Linux systems. Discuss how you would configure firewalls, manage user permissions, and apply security patches. Additionally, explain how you would monitor the systems for potential security threats and respond to any detected vulnerabilities. [15]

B. One of the critical Linux servers in your company has started experiencing performance issues, causing disruptions to business operations. As the Linux system administrator, describe the troubleshooting process you would follow to diagnose and resolve the performance problems. Include details on the tools and commands you would use to identify the root cause, as well as the steps you would take to implement a solution and prevent similar issues in the future. [15]

3. Methods

3.1 Research platforms used

On my journey to understanding the various concepts required for producing the report I have made use of our prescribed textbooks outlined in our study guide for this module. Additionally, I have made use of documentation made by technical writers.

3.2 Documentation methods, processes and tools used.

My report follows the format of an academic report. To produce my references, I have made use Harvard Angelia 2008 citation style.

4. Results

4.1 Recovering from a systems breach in the 20th century. (Section A Question A) [15]

TODO: write in between text

4.1.1 Implementing security measures to protect our Linux systems.

1. Disabling Unused Services

We can disable network applications that we are not making use of, as they might have vulnerabilities. This includes FTP, Telnet, Finger, and Mail Services if we do not send or receive mail.

2. Changing Default Application Network Ports

We can change the well-known and registered ports (IANA) and replace them with private ports that our applications can bind to. This will help defend against basic attacks that target well-known and registered ports, but an experienced attacker will be scanning private ports (49511) and up.

3. Using Encryption on the Network

We should make use of the Secure Sockets Layer Protocol (SSL) wherever possible and TLS (v1.3) to encrypt data as it is transferred across the network. This might be for file transfers, or web applications. (https)

4. Encrypted disks

In the modern IT environment, our data is being stored not only on premises, but in the cloud. We should ensure that we have disk encryption support with our cloud provider, and that we enable it.

(Richard Blum, 2022)

5. We should prevent access to the GRUB Bootloader

During the boot process, Linux uses the GRUB bootloader to load the appropriate operating system image from the storage. The GRUB system also provides a recovery mode, through which you can gain root access to a system. To prevent this we should make use of a password on the GRUB boot loader to prevent unauthorized access to the GRUB menu. Grub-mkpasswd-pbkdf2 can be used, along with an encrypted password value.

4.1.2 Configuring Firewalls, Managing Users, Applying Patches.

Configuring Firewalls

Firewalls allow us to control access to network resources. As this module is at nqf7, this statement may be redundant. I feel that a recap of the fundamentals is of great importance. This includes:

1. Reviewing what the packet control information is in terms of firewall configuration with access control lists and what are the typical actions that we take once we identify a packet.
2. What the difference is between stateless and stateful firewalls.
3. Possible performance impacts of choosing the "uncomplicated" solution with relation to Linux firewall configuration.

1. Reviewing packet control information and firewall packet actions.

What information is used by an access control list to identify a packet?

- Source IP address
- Destination IP address
- Network protocol (User Datagram Protocol [udp] or Transmission Control Protocol [tcp] or [any])
- Inbound port (if we have a rule that blocks or allows access to resources that we may want to share, and we are applying a rule to a web server.)
- Outbound port
- Network state

We can usually perform actions with the identified traffic such as:

- Accepting (Giving the traffic the green light)
- Reject (Giving the traffic the red light)
- Drop (Giving the traffic the red light, but not turning on the light)
- Log (For security or for regulatory compliance purposes)

6. What is the difference between stateless and stateful firewalls?

Stateless firewalls:

Stateless firewalls do not track information such as connections, network status, data flows, or traffic patterns. Instead, each packet is individually evaluated, without any tracked information.

This may not be beneficial due to the fact that network attacks can be spread among multiple packets. Another consideration is the fact that stateless firewall software may need to be restarted every time that we update the firewall rules.

Stateful firewalls:

The implementation of a stateful firewall allows us to track packet information such as connections, network status, data flows and traffic patterns. With a stateful firewall, we will be able to combat more network attacks.

To facilitate the tracking of network packets, network information is kept in computer memory, building a connection table. The creation of the table takes longer; however, decisions are made faster for established connections.

7. Possible implications of choosing the “uncomplicated” option.

Netfilter is a project that provides a framework inside the Linux kernel for packet filtering, allowing us to create both stateful and stateless firewall filters. In simple terms, the majority of other Linux “firewalling” software will call upon the Netfilter framework to create firewall rules for every packet that traverses the network stack, at the level at which the call to Netfilter has been made. (Stack as in network layer, transmission layer, and application layer)

(netfilter.org, 2024)

Solutions such as nftable, iptables, firewalld, and UFW make their calls to the netfilter framework. Firewalld and UFW adds additional processing layer due to their user interfaces that can slow down the packet filtering process. To obtain the best performance, a service that provides low-level access to netfilter should be used such as nftable and iptables.

(Blum, 2022)

4.1.2.1 A practical example of configuring firewalls

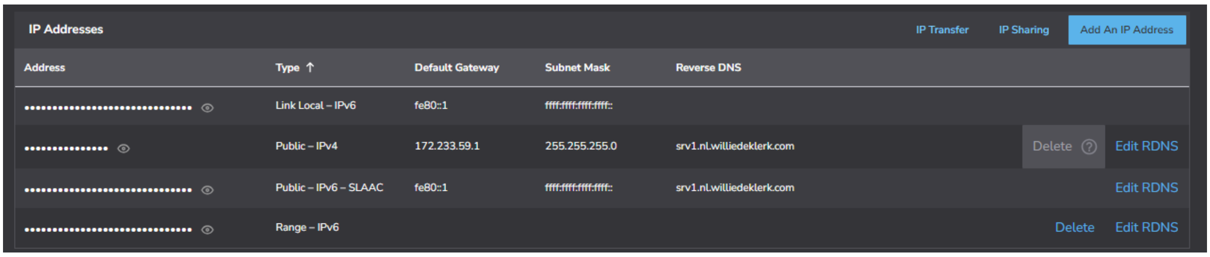
I am deploying a recursive resolver (dns-server) srv1.nl.williedeklerk.com with my dns transport protocol being udp/tcp port 53, using Linode as my cloud service provider and using their services in the Netherlands. I will be using both IPv4 and IPv6 addressing.

Creating a recursive resolver can have security risks, and people can misuse the company's resolver for malicious purposes. (Can use my server to generate large volumes of dns traffic and flood someone else's infrastructure)

Due to the above mentioned, configuring firewall rules to allow dns traffic from specific company branches, or cloud gateways where we have clients (stub resolvers) is of vital importance. In my example, I will have firewall rules that allow dns requests from srv2.nl.williedeklerk.com.

Whilst "mom-and-pop" shops mostly use google dns servers or any other well-known provider, in a large enterprise environment an organization may choose to deploy their own recursive resolvers in line with company policies and procedures.

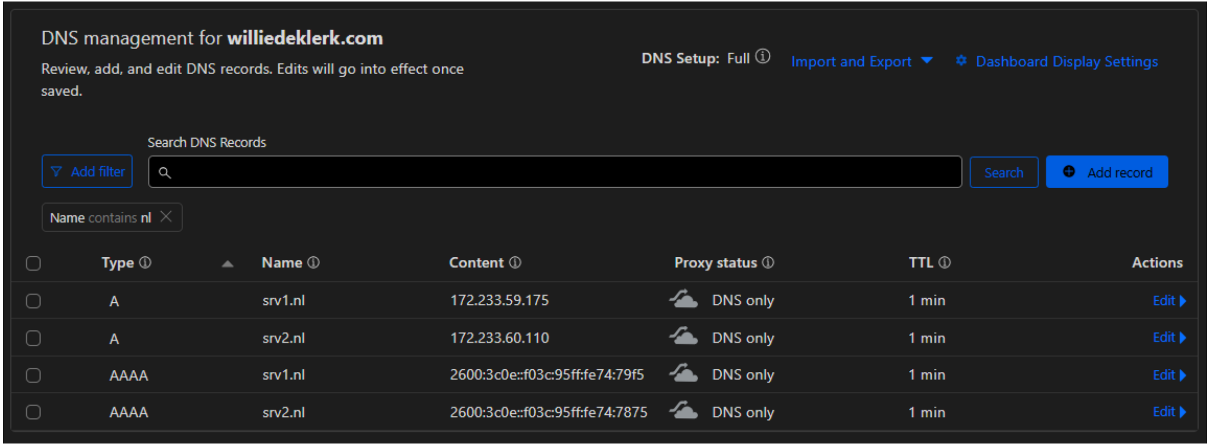
Reference to unbound that I installed: (nlnetlabs.nl, 2025)



IP Addresses

Address	Type ↑	Default Gateway	Subnet Mask	Reverse DNS	
••••••••••••••••••••••••••••••	Link Local – IPv6	fe80::1	ffff:ffff:ffff:ffff::		
•••••••••••••••	Public – IPv4	172.233.59.1	255.255.255.0	srv1.nl.williedeklerk.com	Delete ⓘ Edit RDNS
••••••••••••••••••••••••••••••	Public – IPv6 – SLAAC	fe80::1	ffff:ffff:ffff:ffff::	srv1.nl.williedeklerk.com	Edit RDNS
••••••••••••••••••••••••••••••	Range – IPv6				Delete Edit RDNS

Figure 1 srv1.nl.williedeklerk.com Updated Reverse DNS records, This is used when "inverse queries" are made, however we should always remember that in reality we are making standard queries to *.in-addr.arpa or *.ip6.arpa



DNS management for **williedeklerk.com**

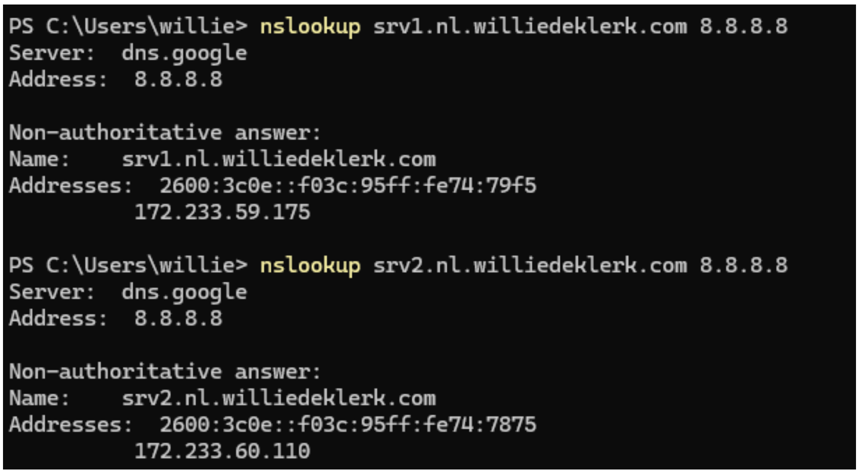
Review, add, and edit DNS records. Edits will go into effect once saved.

DNS Setup: Full ⓘ

Name contains nl

□	Type ⓘ	Name ⓘ	Content ⓘ	Proxy status ⓘ	TTL ⓘ	Actions
□	A	srv1.nl	172.233.59.175	DNS only	1 min	Edit ▸
□	A	srv2.nl	172.233.60.110	DNS only	1 min	Edit ▸
□	AAAA	srv1.nl	2600:3c0e::f03c:95ff:fe74:79f5	DNS only	1 min	Edit ▸
□	AAAA	srv2.nl	2600:3c0e::f03c:95ff:fe74:7875	DNS only	1 min	Edit ▸

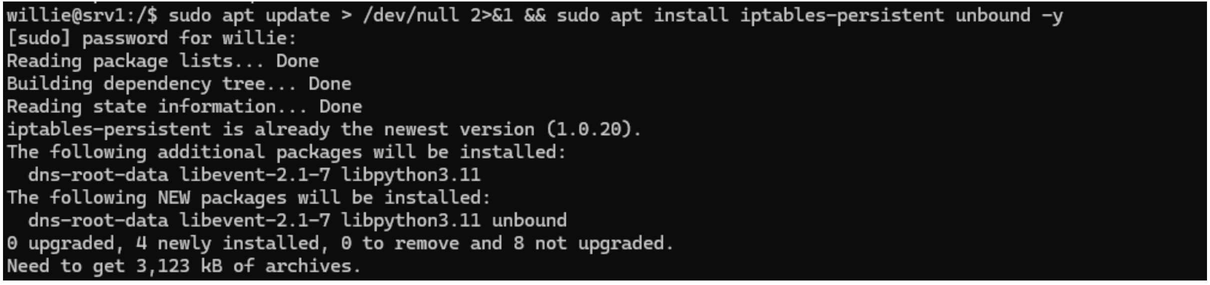
Figure 2 DNS records for my example servers



```
PS C:\Users\willie> nslookup srv1.nl.williedeklerk.com 8.8.8.8
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    srv1.nl.williedeklerk.com
Addresses:  2600:3c0e::f03c:95ff:fe74:79f5
          172.233.59.175

PS C:\Users\willie> nslookup srv2.nl.williedeklerk.com 8.8.8.8
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    srv2.nl.williedeklerk.com
Addresses:  2600:3c0e::f03c:95ff:fe74:7875
          172.233.60.110
```

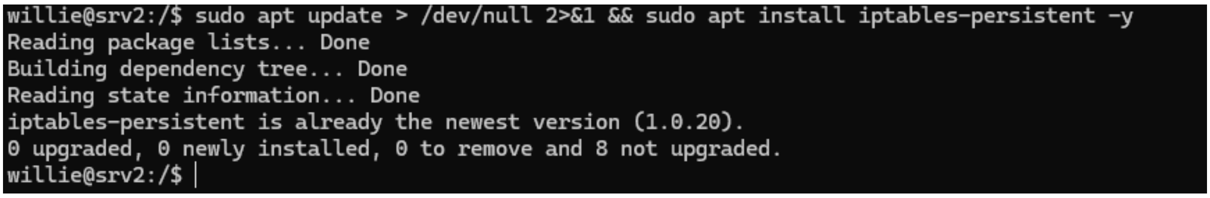
Figure 3 Performing a dns Standard Query using nslookup and specifying 8.8.8.8 as my recursive resolver. Point of this is to check my records that I just added.

Figure 4 My Public IP Addressing Information



```
willie@srv1:/$ sudo apt update > /dev/null 2>&1 && sudo apt install iptables-persistent unbound -y
[sudo] password for willie:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables-persistent is already the newest version (1.0.20).
The following additional packages will be installed:
  dns-root-data libevent-2.1-7 libpython3.11
The following NEW packages will be installed:
  dns-root-data libevent-2.1-7 libpython3.11 unbound
0 upgraded, 4 newly installed, 0 to remove and 8 not upgraded.
Need to get 3,123 kB of archives.
```

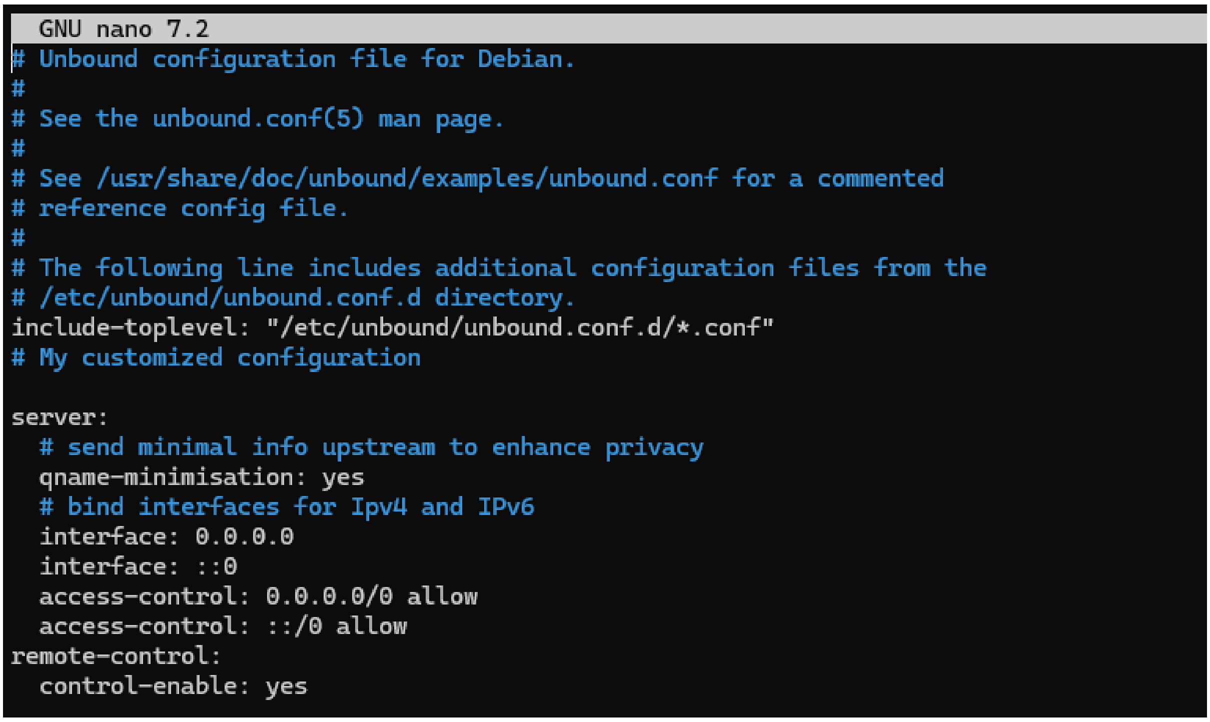
Figure 5 Running sudo apt update on srv1.nl.williedeklerk.com, suppressing the output of apt update to hide my sources from screenshot. Additionally, I am installing iptables-persistent and unbound to turn by server into a recursive resolver



```
willie@srv2:/$ sudo apt update > /dev/null 2>&1 && sudo apt install iptables-persistent -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables-persistent is already the newest version (1.0.20).
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
willie@srv2:/$ |
```

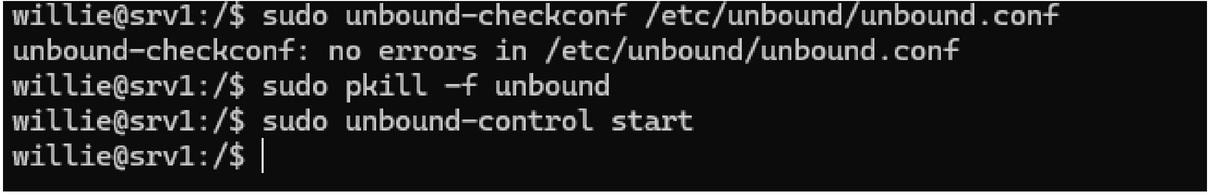
Figure 6 Running sudo apt update on srv2.nl.williedeklerk.com, suppressing the output of apt update to hide my sources. Additionally, I am installing iptables-persistent

Configure unbound



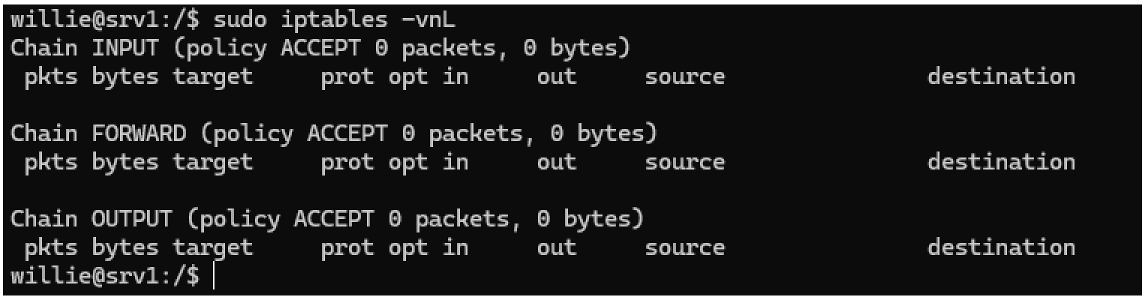
```
 GNU nano 7.2
# Unbound configuration file for Debian.
#
# See the unbound.conf(5) man page.
#
# See /usr/share/doc/unbound/examples/unbound.conf for a commented
# reference config file.
#
# The following line includes additional configuration files from the
# /etc/unbound/unbound.conf.d directory.
include-toplevel: "/etc/unbound/unbound.conf.d/*.conf"
# My customized configuration

server:
  # send minimal info upstream to enhance privacy
  qname-minimisation: yes
  # bind interfaces for Ipv4 and IPv6
  interface: 0.0.0.0
  interface: ::0
  access-control: 0.0.0.0/0 allow
  access-control: ::/0 allow
remote-control:
  control-enable: yes
```

Figure 7 My config file for unbound, it states that all ip addresses are allowed but unbound does not make a firewall rule thus I am going to add a firewall rule instead of changing the config file



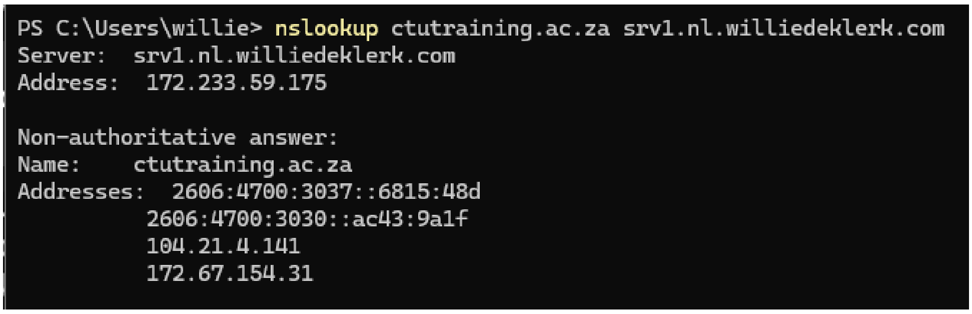
```
willie@srv1:/$ sudo unbound-checkconf /etc/unbound/unbound.conf
unbound-checkconf: no errors in /etc/unbound/unbound.conf
willie@srv1:/$ sudo pkill -f unbound
willie@srv1:/$ sudo unbound-control start
willie@srv1:/$
```

Figure 8 Checking unbound configuration for errors, stopping unbound, and starting unbound again



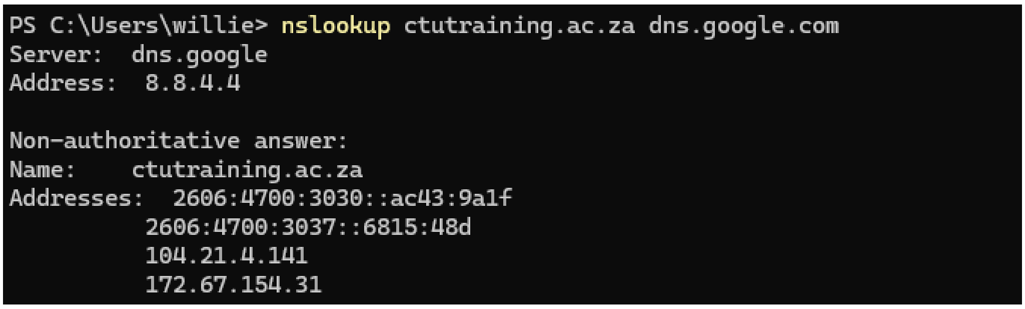
```
willie@srv1:/$ sudo iptables -vnL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
willie@srv1:/$
```

Figure 9 My Current Firewall rules (This is the default which as accept any any. Not a very good idea especially in a production environment)



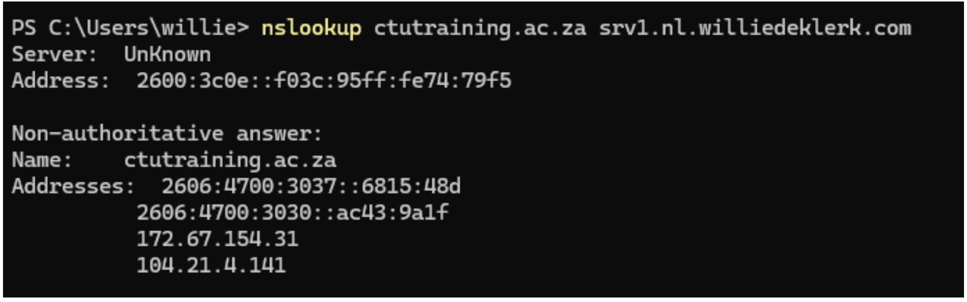
```
PS C:\Users\willie> nslookup ctutraining.ac.za srv1.nl.williedeklerk.com
Server:  srv1.nl.williedeklerk.com
Address:  172.233.59.175

Non-authoritative answer:
Name:    ctutraining.ac.za
Addresses:  2606:4700:3037::6815:48d
          2606:4700:3030::ac43:9a1f
          104.21.4.141
          172.67.154.31
```

Figure 10 nslookup on my laptop looking at the CTU Training Solutions main site, using my recursive resolver. (IPv4)



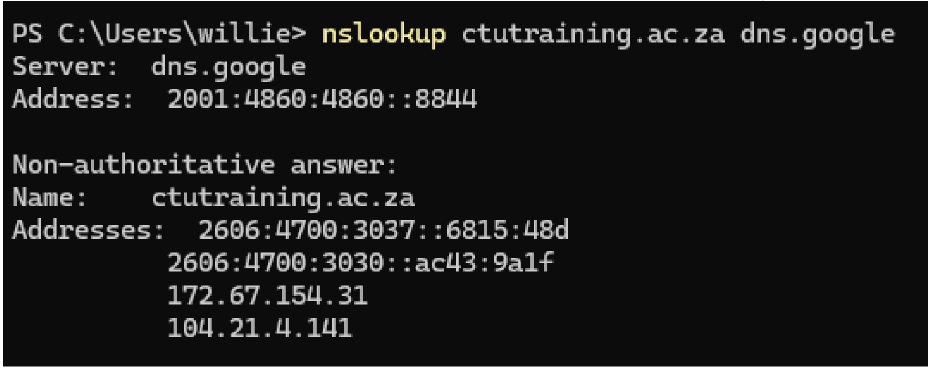
```
PS C:\Users\willie> nslookup ctutraining.ac.za dns.google.com
Server:  dns.google
Address:  8.8.4.4

Non-authoritative answer:
Name:    ctutraining.ac.za
Addresses:  2606:4700:3030::ac43:9a1f
          2606:4700:3037::6815:48d
          104.21.4.141
          172.67.154.31
```

Figure 11 nslookup on my laptop looking at the CTU Training Solutions main site, using one of google's free public recursive dns resolvers. (IPv4)



```
PS C:\Users\willie> nslookup ctutraining.ac.za srv1.nl.williedeklerk.com
Server:  UnKnown
Address:  2600:3c0e::f03c:95ff:fe74:79f5

Non-authoritative answer:
Name:    ctutraining.ac.za
Addresses:  2606:4700:3037::6815:48d
          2606:4700:3030::ac43:9a1f
          172.67.154.31
          104.21.4.141
```

Figure 12 performing an nslookup on my laptop for the CTU Training Solutions main website using my recursive resolver srv1.nl.williedeklerk.com (IPv6) Server name should load in 48 hours. I am not waiting around.

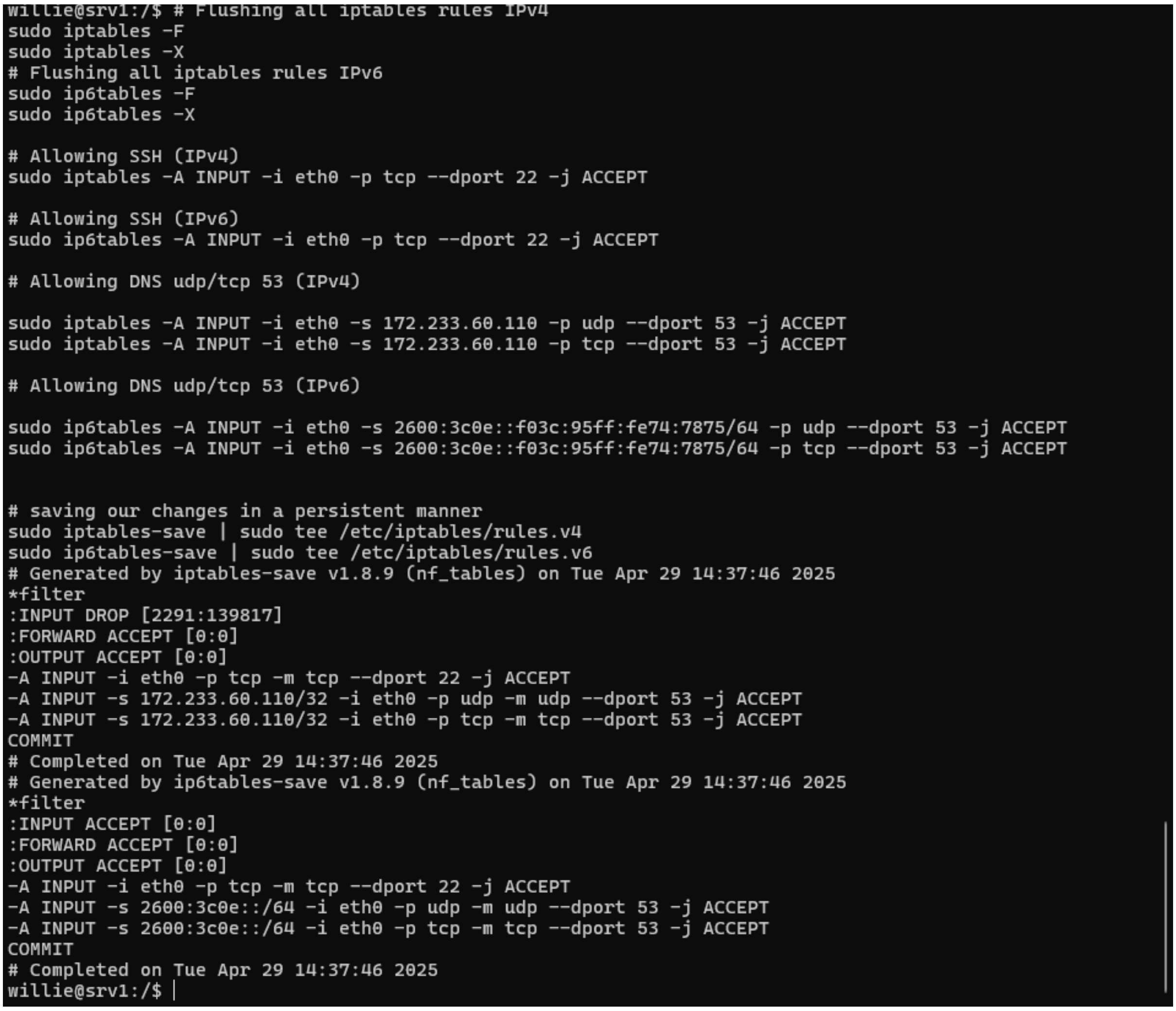


```
PS C:\Users\willie> nslookup ctutraining.ac.za dns.google
Server:  dns.google
Address:  2001:4860:4860::8844

Non-authoritative answer:
Name:    ctutraining.ac.za
Addresses:  2606:4700:3037::6815:48d
          2606:4700:3030::ac43:9a1f
          172.67.154.31
          104.21.4.141
```

Figure 13 performing an nslookup on my laptop for the CTU Training Solutions main website using one of google's free recursive resolvers.(IPv6)

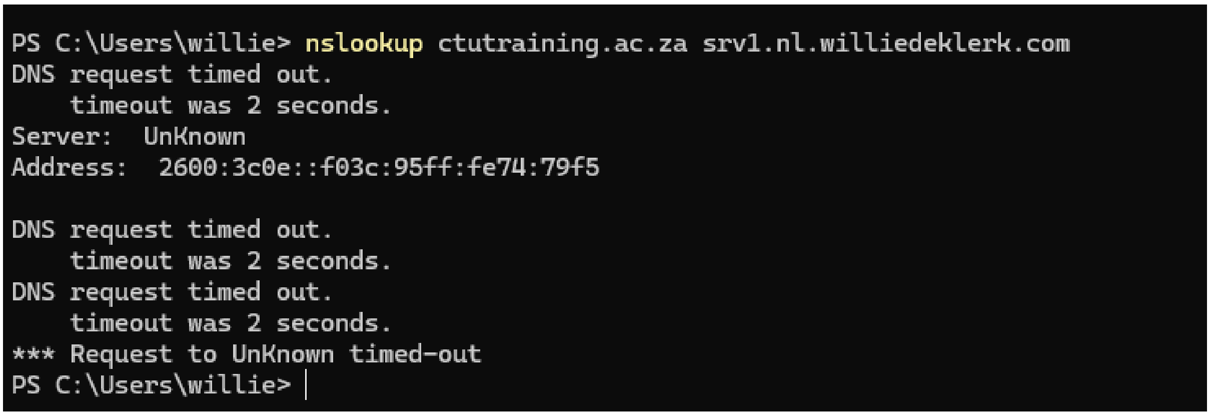
Closing the resolver with firewall rules and our goal is :

- that only srv2.nl.williedeklerk.com can access and use it for dns purposes.
- I should still be able to ssh into it from home.
- My dns server srv1.nl.williedeklerk.com should not be accepting any other input traffic.
- IPv4 and IPv6 firewall configuration.



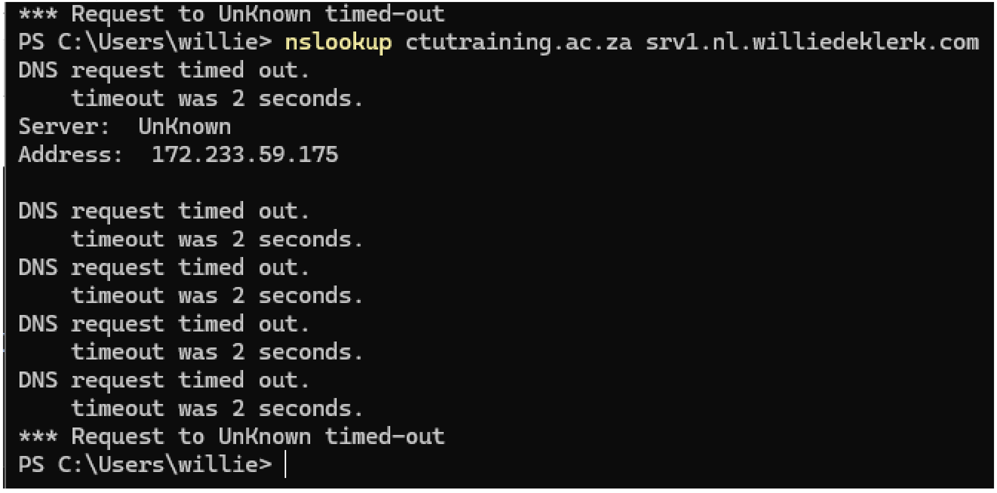
```
willie@srv1:/$ # Flushing all iptables rules IPv4
sudo iptables -F
sudo iptables -X
# Flushing all iptables rules IPv6
sudo ip6tables -F
sudo ip6tables -X

# Allowing SSH (IPv4)
sudo iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT

# Allowing SSH (IPv6)
sudo ip6tables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT

# Allowing DNS udp/tcp 53 (IPv4)

sudo iptables -A INPUT -i eth0 -s 172.233.60.110 -p udp --dport 53 -j ACCEPT
sudo iptables -A INPUT -i eth0 -s 172.233.60.110 -p tcp --dport 53 -j ACCEPT

# Allowing DNS udp/tcp 53 (IPv6)

sudo ip6tables -A INPUT -i eth0 -s 2600:3c0e::f03c:95ff:fe74:7875/64 -p udp --dport 53 -j ACCEPT
sudo ip6tables -A INPUT -i eth0 -s 2600:3c0e::f03c:95ff:fe74:7875/64 -p tcp --dport 53 -j ACCEPT


# saving our changes in a persistent manner
sudo iptables-save | sudo tee /etc/iptables/rules.v4
sudo ip6tables-save | sudo tee /etc/iptables/rules.v6
# Generated by iptables-save v1.8.9 (nf_tables) on Tue Apr 29 14:37:46 2025
*filter
:INPUT DROP [2291:139817]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i eth0 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -s 172.233.60.110/32 -i eth0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -s 172.233.60.110/32 -i eth0 -p tcp -m tcp --dport 53 -j ACCEPT
COMMIT
# Completed on Tue Apr 29 14:37:46 2025
# Generated by ip6tables-save v1.8.9 (nf_tables) on Tue Apr 29 14:37:46 2025
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i eth0 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -s 2600:3c0e::/64 -i eth0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -s 2600:3c0e::/64 -i eth0 -p tcp -m tcp --dport 53 -j ACCEPT
COMMIT
# Completed on Tue Apr 29 14:37:46 2025
willie@srv1:/$
```

Figure 14 Configuring the Firewall for IPv4 and IPv6 operations, using iptables and ip6tables. Saving for rule presistance.



```
PS C:\Users\willie> nslookup ctutraining.ac.za srv1.nl.williedeklerk.com
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  2600:3c0e::f03c:95ff:fe74:79f5

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to UnKnown timed-out
PS C:\Users\willie> |
```

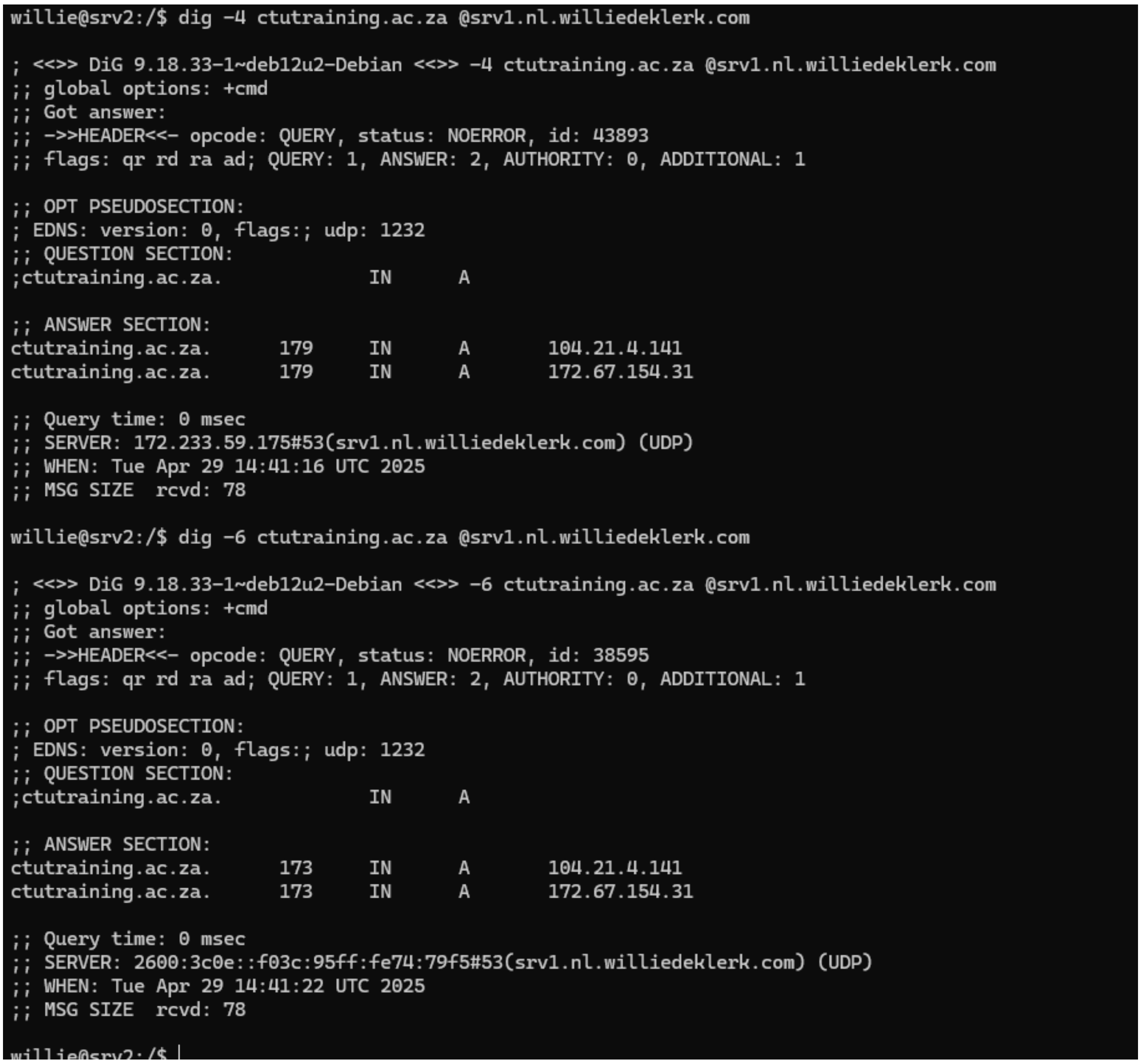
Figure 15 Performing a DNS query on my laptop after the firewall rules have been applied. (IPv6)



```
*** Request to UnKnown timed-out
PS C:\Users\willie> nslookup ctutraining.ac.za srv1.nl.williedeklerk.com
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  172.233.59.175

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to UnKnown timed-out
PS C:\Users\willie> |
```

Figure 16 Performing a DNS query on my laptop after the firewall rules have been applied (IPv4)

Performing dns query using dig with IPv4 and IPv6 on srv2.nl.williedeklerk.com



```
willie@srv2:/$ dig -4 ctutraining.ac.za @srv1.nl.williedeklerk.com

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> -4 ctutraining.ac.za @srv1.nl.williedeklerk.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43893
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;ctutraining.ac.za.             IN      A

;; ANSWER SECTION:
ctutraining.ac.za.      179     IN      A       104.21.4.141
ctutraining.ac.za.      179     IN      A       172.67.154.31

;; Query time: 0 msec
;; SERVER: 172.233.59.175#53(srv1.nl.williedeklerk.com) (UDP)
;; WHEN: Tue Apr 29 14:41:16 UTC 2025
;; MSG SIZE  rcvd: 78

willie@srv2:/$ dig -6 ctutraining.ac.za @srv1.nl.williedeklerk.com

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> -6 ctutraining.ac.za @srv1.nl.williedeklerk.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38595
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;ctutraining.ac.za.             IN      A

;; ANSWER SECTION:
ctutraining.ac.za.      173     IN      A       104.21.4.141
ctutraining.ac.za.      173     IN      A       172.67.154.31

;; Query time: 0 msec
;; SERVER: 2600:3c0e::f03c:95ff:fe74:79f5#53(srv1.nl.williedeklerk.com) (UDP)
;; WHEN: Tue Apr 29 14:41:22 UTC 2025
;; MSG SIZE  rcvd: 78

willie@srv2:/$
```

Figure 17 Performing dns query using dig ipv4 -4 and ipv6 -6 on srv2.nl.williedeklerk.com

4.1.2.2 Managing users

1. Enforce the use of strong passwords, using Pluggable authentication Modules on Linux.

2. We should also enforce the use of a Pluggable authentication module for password history, to prevent a user from using an old password when they are making a new password.

3. Allow remote access to servers through ssh, not telnet, and require the use of properly managed certificates for ssh.

4. We should make use of Access Control lists to manage user permissions to files. It offers more advanced capabilities in terms of discretionary access control (DAC) methods than the conventional File and Directory Permissions.

5. We should make use of context based, mandatory access controls, such as Role Based Access Control to assign security permissions on roles and processes in our Linux Environments.

(Richard Blum, 2022) (Richard Blum, 2022)

4.1.2.3 Applying Patches

What is a Patch?

A patch refers to program changes or configuration file updates for a system service or application. They may be for code problems or for security problems. It does not include updating all system software.

How do we apply them?

We can manually apply kernel patches using the **patch** command, however package managers such as apt typically handles this for us when we upgrade the packages on the system.

(Richard Blum, 2022)

4.1.3 Monitoring Systems for potential security threats and respond to any detected vulnerabilities.

Nmap

Nmap is a free and open-source utility for network discovery and auditing. It is also used for inventorying, managing service upgrade schedules and monitoring host or service uptime. (nmap.org, 2025)

Pialert

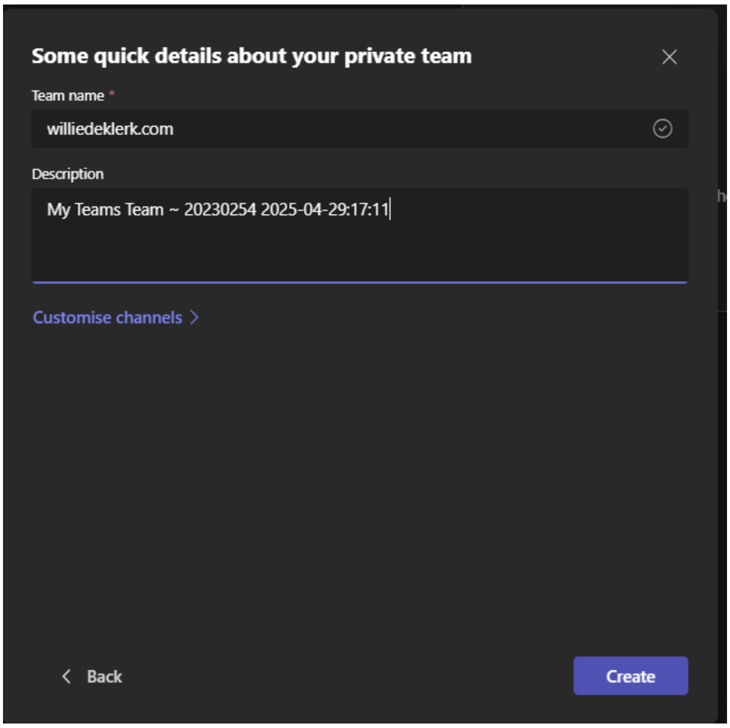
Pialert can be used to keep inventory, and to find roque dhcp servers, and perform scheduled, automated Nmap scans with reports stored in a database of sorts. I have personally made use of Pi.Alert during my Work Integrated Learning (WIL620) module last year, on a Fusion Broadband SD-WAN edge node, in the form of a nspawn container. (Pi.Alert, 2025)

Practical Example using srv2.nl.williedeklerk.com

I am stating my references here; I have to three because I am doing my own thing. One shows how connectors work with webhooks, and the other one showed me how to add a web hook to fail2ban when it bans an ip address. It served as inspiration.

(Postman Downloads, 2025) (Cole Turner, 2020) (vxpse.blog, 2022)

When a server is rebooted or powered off, without the administrators instruction there might be malicious activity where someone is trying to gain access before the operating system loads. This is a rather simple example but it works.



Some quick details about your private team

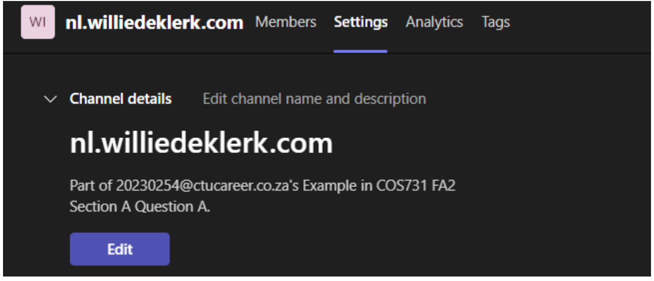
Team name *

williedeklerk.com

Description

My Teams Team ~ 20230254 2025-04-29:17:11

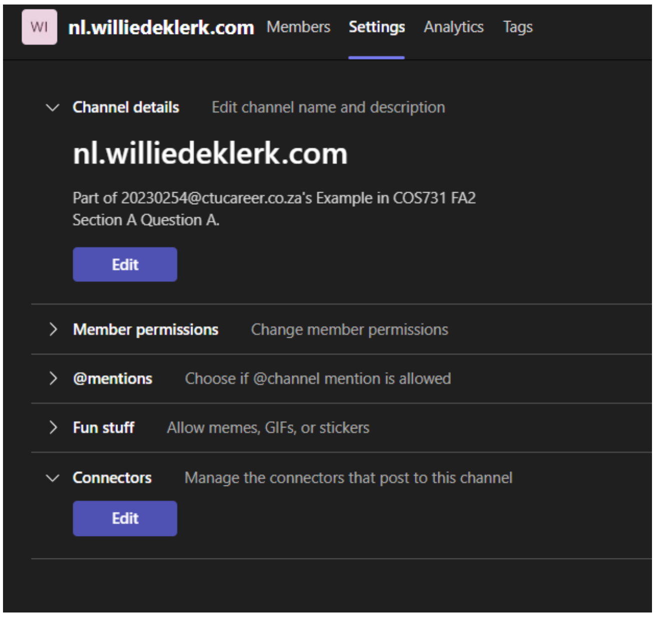
Figure 18 Creating my team



nl.williedeklerk.com

Part of 20230254@ctucareer.co.za's Example in COS731 FA2 Section A Question A.

Figure 19 Creating a Teams channel



Channel details — Edit channel name and description

nl.williedeklerk.com

Part of 20230254@ctucareer.co.za's Example in COS731 FA2 Section A Question A.

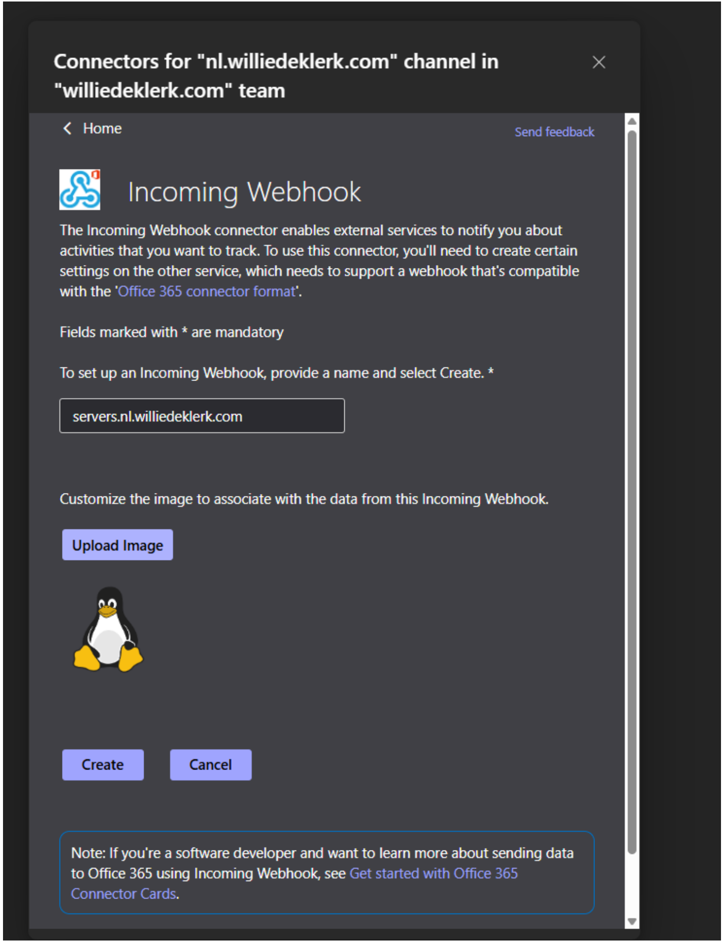
Member permissions — Change member permissions

@mentions — Choose if @channel mention is allowed

Fun stuff — Allow memes, GIFs, or stickers

Connectors — Manage the connectors that post to this channel

Figure 20 Settings Connectors



Connectors for "nl.williedeklerk.com" channel in "williedeklerk.com" team

Incoming Webhook

The Incoming Webhook connector enables external services to notify you about activities that you want to track. To use this connector, you'll need to create certain settings on the other service, which needs to support a webhook that's compatible with the 'Office 365 connector format'.

Fields marked with * are mandatory

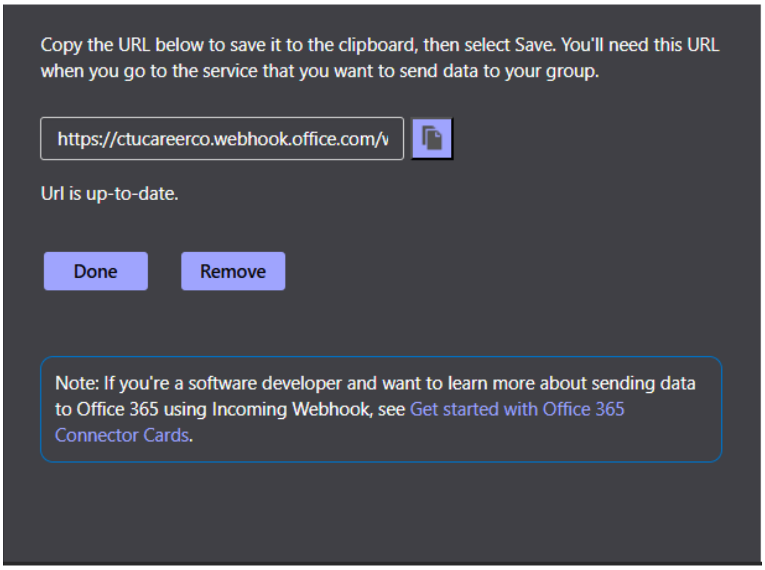
To set up an Incoming Webhook, provide a name and select Create. *

servers.nl.williedeklerk.com

Customize the image to associate with the data from this Incoming Webhook.

Note: If you're a software developer and want to learn more about sending data to Office 365 using Incoming Webhook, see Get started with Office 365 Connector Cards.

Figure 21 Adding an Incoming Webhook



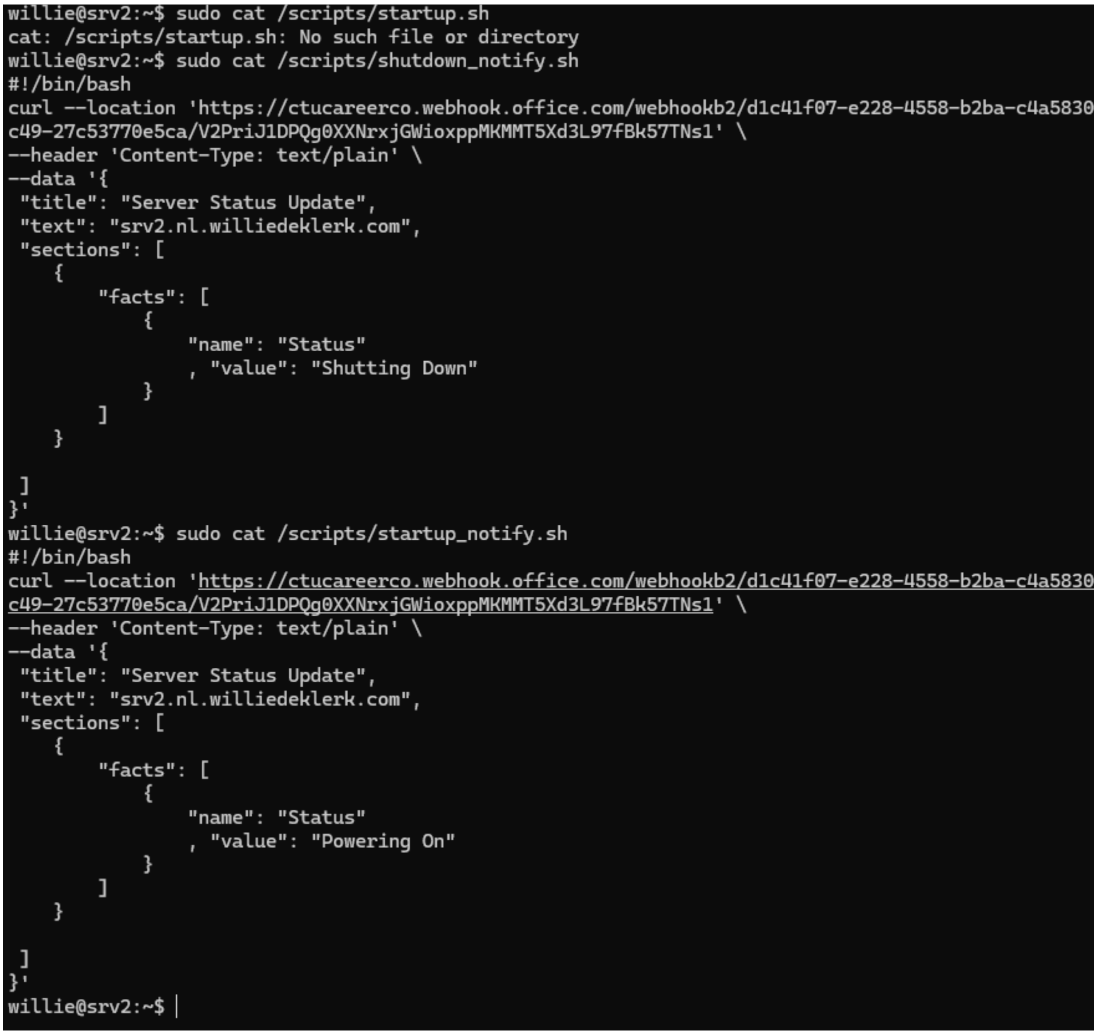
Copy the URL below to save it to the clipboard, then select Save. You'll need this URL when you go to the service that you want to send data to your group.

https://ctucareerco.webhook.office.com/v

Url is up-to-date.

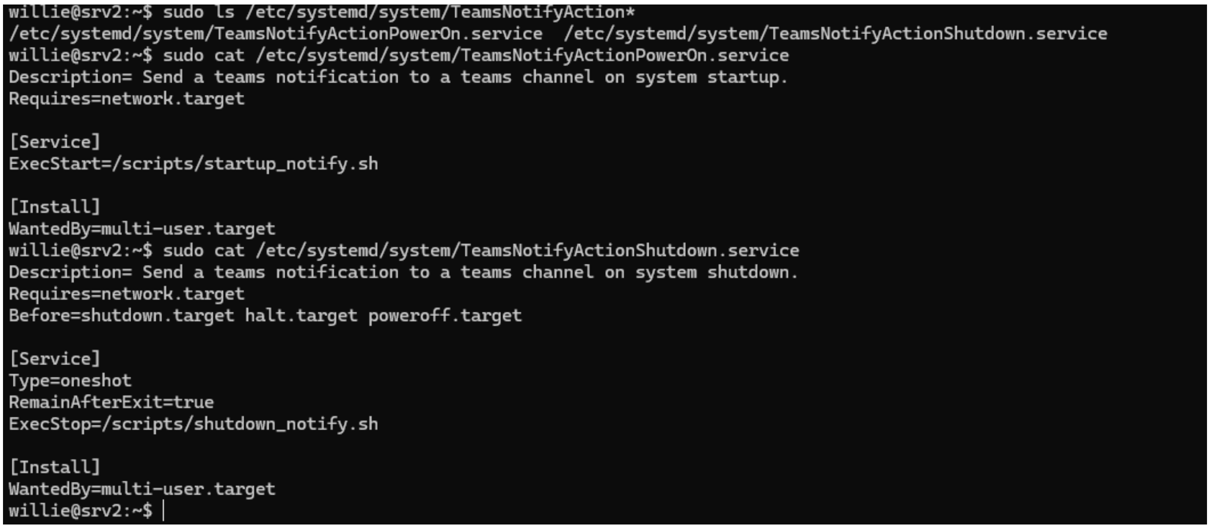
Note: If you're a software developer and want to learn more about sending data to Office 365 using Incoming Webhook, see Get started with Office 365 Connector Cards.

Figure 22 The url that I will be using for my web hook



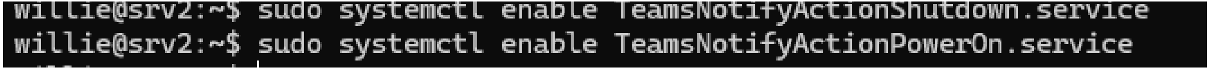
```
willie@srv2:~$ sudo cat /scripts/startup.sh
cat: /scripts/startup.sh: No such file or directory
willie@srv2:~$ sudo cat /scripts/shutdown_notify.sh
#!/bin/bash
curl --location 'https://ctucareerco.webhook.office.com/webhookb2/d1c41f07-e228-4558-b2ba-c4a5830
c49-27c53770e5ca/V2PriJ1DPQg0XXNrxjGWioxppMKMMT5Xd3L97fBk57TNs1' \
--header 'Content-Type: text/plain' \
--data '{
 "title": "Server Status Update",
 "text": "srv2.nl.williedeklerk.com",
 "sections": [
    {
        "facts": [
            {
                "name": "Status"
                , "value": "Shutting Down"
            }
        ]
    }

 ]
}'
willie@srv2:~$ sudo cat /scripts/startup_notify.sh
#!/bin/bash
curl --location 'https://ctucareerco.webhook.office.com/webhookb2/d1c41f07-e228-4558-b2ba-c4a5830
c49-27c53770e5ca/V2PriJ1DPQg0XXNrxjGWioxppMKMMT5Xd3L97fBk57TNs1' \
--header 'Content-Type: text/plain' \
--data '{
 "title": "Server Status Update",
 "text": "srv2.nl.williedeklerk.com",
 "sections": [
    {
        "facts": [
            {
                "name": "Status"
                , "value": "Powering On"
            }
        ]
    }

 ]
}'
willie@srv2:~$
```

Figure 23 My Bash Scripts that post the web request using curl



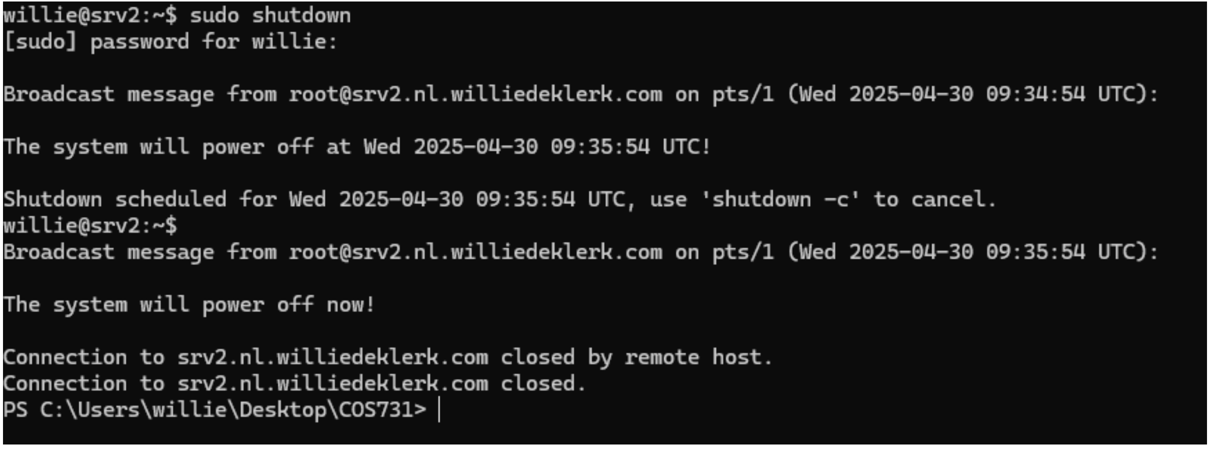
```
willie@srv2:~$ sudo ls /etc/systemd/system/TeamsNotifyAction*
/etc/systemd/system/TeamsNotifyActionPowerOn.service  /etc/systemd/system/TeamsNotifyActionShutdown.service
willie@srv2:~$ sudo cat /etc/systemd/system/TeamsNotifyActionPowerOn.service
Description= Send a teams notification to a teams channel on system startup.
Requires=network.target

[Service]
ExecStart=/scripts/startup_notify.sh

[Install]
WantedBy=multi-user.target
willie@srv2:~$ sudo cat /etc/systemd/system/TeamsNotifyActionShutdown.service
Description= Send a teams notification to a teams channel on system shutdown.
Requires=network.target
Before=shutdown.target halt.target poweroff.target

[Service]
Type=oneshot
RemainAfterExit=true
ExecStop=/scripts/shutdown_notify.sh

[Install]
WantedBy=multi-user.target
willie@srv2:~$ |
```

Figure 24 My services that I created



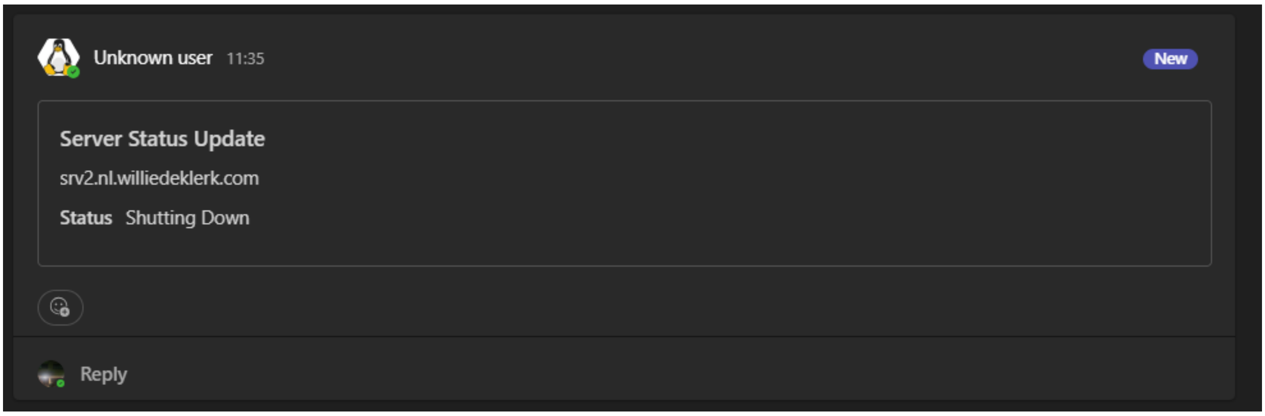
```
willie@srv2:~$ sudo systemctl enable TeamsNotifyActionShutdown.service
willie@srv2:~$ sudo systemctl enable TeamsNotifyActionPowerOn.service
```

Figure 25 I need to enable the services to use them



```
willie@srv2:~$ sudo shutdown
[sudo] password for willie:

Broadcast message from root@srv2.nl.williedeklerk.com on pts/1 (Wed 2025-04-30 09:34:54 UTC):

The system will power off at Wed 2025-04-30 09:35:54 UTC!

Shutdown scheduled for Wed 2025-04-30 09:35:54 UTC, use 'shutdown -c' to cancel.
willie@srv2:~$
Broadcast message from root@srv2.nl.williedeklerk.com on pts/1 (Wed 2025-04-30 09:35:54 UTC):

The system will power off now!

Connection to srv2.nl.williedeklerk.com closed by remote host.
Connection to srv2.nl.williedeklerk.com closed.
PS C:\Users\willie\Desktop\COS731> |
```

Figure 26 Shutting down the machine for a test



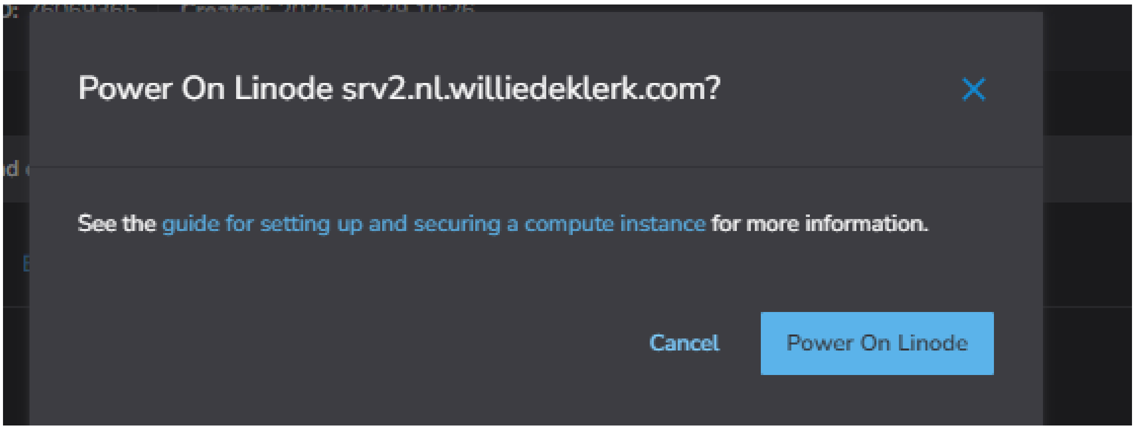
Unknown user 11:35

Server Status Update

srv2.nl.williedeklerk.com

Status Shutting Down

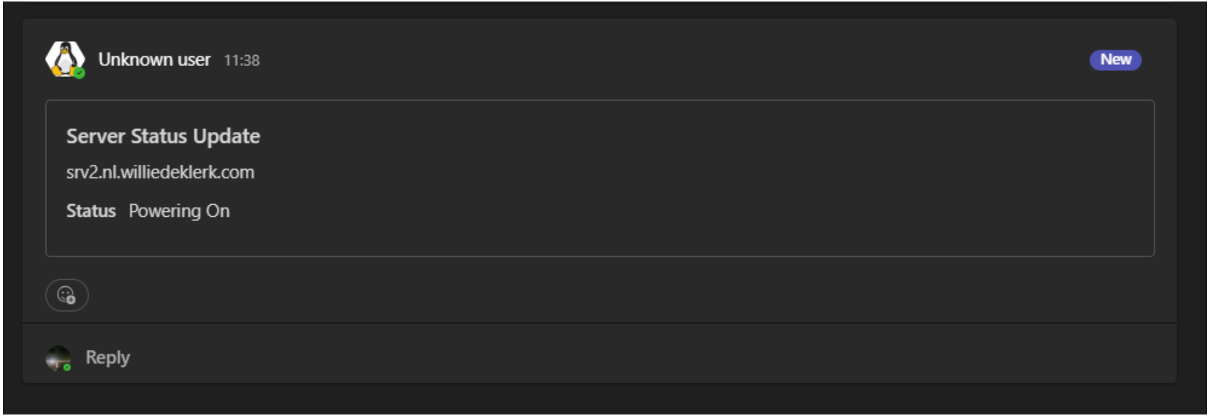
Figure 27 Result in teams channel



Power On Linode srv2.nl.williedeklerk.com?

See the guide for setting up and securing a compute instance for more information.

Figure 28 Powering the server back on



Server Status Update

srv2.nl.williedeklerk.com

Status Powering On

Figure 29 Result of powering on the server

4.2 Troubleshooting Linux systems in a production environment. [15]

4.2.1 What is a root cause analysis?

As stated by Benjamin Cane in the book: Red Hat Enterprise Linux Troubleshooting Guide Chapter 1 Section 3, A root cause analysis is an analytical process performed after incidents arise. The goal of the root cause analysis is to identify the root cause of the incident and to identify any possible corrective actions to prevent the same incident from occurring again.

A good root cause analysis consists of the following elements:

- The problem, as it was first described.
- The actual root cause of the problem, after analysis.
- An accurate timeline of events and actions taken.
- Any key data points.
- A plan of amendatory action, to prevent future occurrences.

(Cane, 2015)

4.2.2 What commands can be used to troubleshoot systems in a production environment.

1. ping

2. trippy (combines ping and traceroute functionality similar to my trace route) (fujiapple852, 2025)

3. telnet

4. netstat

5. tcpdump

6. systemctl status *.service

7. lscpu

8. dmesg

4.2.3 Implementation steps for amendatory action

There are five implementation steps for amendatory action:

1. Understanding the problem statement as it was described.
2. Establishing a hypothesis.
3. Trial and error.
4. Getting help.
5. Documentation.

(Cane, 2015)

Implementation of these steps may vary slightly depending on the environment and the scenario.

An example of troubleshooting in a production environment.

A user, utilizing a dns server srv1.nl.williedeklerk.com submitted a support ticket to support.nl.williedeklerk.com:

I am facing problems reaching a resource located at the domain ctutraining.ac.za, however I can ping the IP address of the dns server. I am is using the dns server for home usage, thus I do not have any enterprise networking setups such as having a forward proxy, or a firewall in more than the sense of source network address translation. I am not making use of any overlay technologies such as IPsec or WireGuard.

Please help.

Sincerely,

User ID: 256512102420484096

Email: JohnDoe2003@gmail.com

Step 1: Understanding the problem statement as it was described.

In a production environment we should establish what the problem is, and the problem is that the user can't reach their website. The root cause might not be the obvious answer. We can ask questions such as:

- Is the recursive resolver refusing connections?
- Or is the recursive resolver denying access to a specific domain name or resource?
- Is the stub resolver (client) using the wrong transport method (dns over https instead of udp tcp 53) ?
- Is the record that the client is trying to access carrying a large, such as a TXT service validation record of 2048 bytes? (a large google site verification record, or something internal)
    - Depending on the recursive resolver configuration, it might not serve requests larger than 512 bytes when the udp /tcp 53 transport method is used. It might then go and switch over to a TCP request.
    - Due to the extra overhead of TCP, some administrators will blatantly drop tcp requests in the firewall configuration.
- Is the client's public IP address being blocked due to rate limiting being triggered?
- What happens when they use another recursive resolver such as the one provided by cloudflare (1.1.1.1) ?
- Has the resource record that the user is trying to reach properly propagated? (It can take up to 48 hours for dns records to propagate in some cases)
- Has the user just made a typo? (quite an embarrassing, common problem)
- Is there a stub resolver cache issue? (user can clear with ipconfig /flushdns) on their windows client pc.

To answer these questions, we would have run investigatory commands, and we would most likely try and duplicate the issue.

Step 2: Establishing a hypothesis

To establish our hypothesis, we should be making use of the data points that we collected in the previous steps, and we should perform a pattern analysis.

Step3: Trial and Error

In this step, we should first of all ensure that we have configuration backups in place.

We could try to remediate the user's issue by allowing all requests to a specific domain name, whitelisting the domain name if it is not already whitelisted. If our hypothesis is that the record is too large, we might allow TCP requests to our dns server.

Step 4: Getting help

If we cannot resolve the user’s ticket, we should try and get help from documentation, man pages, and books.

Step 5: Documentation

At every step during the process, we should document what we have done. It will help when we could not remediate the problem, and we have to escalate. If the issue is recurring, additional documentation should be updated for root cause analysis.

What should our documentation contain?

- The problem statement as we understand it.
- The hypothesis of what is causing the issue.
- The data that we have collected, including the relevant system metrics, and specific errors that we have found.
- Commands that we have executed during the information gathering steps.
- Steps taken during attempts to resolve the issue, including specific commands that we have executed.

(Cane, 2015)

5. Discussion

My findings were made to the best of my knowledge and interpretation of the proposed questions. I answered the questions not with a relative estimation of the mark allocation.

Through the research that I have conducted, I have gained valuable insight into the inner workings of firewalls, applying patches, and the use Pluggable Authentication Modules for strengthening user passwords.

In this research report, I have demonstrated my understanding of Linux system management, showcasing my ability to effectively administer and maintain Linux environments. I illustrated my skills in securing Linux systems, applying best practices to ensure confidentiality, integrity, and availability is maintained. I demonstrated my understanding of troubleshooting Linux systems.

6. Conclusion

In a business environment where Linux is deployed and maintained in a production environment, understanding troubleshooting and setup commands are of vital importance. Having technical knowledge of setting up systems, and the inner workings of the system being deployed will help with troubleshooting.

7. Table of Figures

8. References

Blum, R., 2022. *CompTIA Linux+ Study Guide, 5th Edition Chapter 18 Overseeing Linux Firewalls.* [Online]
Available at: https://learning.oreilly.com/library/view/comptia-linux-study/9781119878940/c18.xhtml#head-2-387
[Accessed 28 04 2025].

Cane, B., 2015. *Red Hat Enterprise Linux Troubleshooting Guide | Chapter 1 Root Cause Analysis | Section 3.* [Online]
Available at: https://learning.oreilly.com/library/view/red-hat-enterprise/9781785283550/ch01s03.html
[Accessed 01 05 2025].

Cane, B., 2015. *Red Hat Enterprise Linux Troubleshooting Guide | Chapter 1 Troubleshooting Best Practices | Section 2 Troubleshooting Steps.* [Online]
Available at: https://learning.oreilly.com/library/view/red-hat-enterprise/9781785283550/ch01s02.html
[Accessed 01 05 2025].

Cole Turner, 2020. *fail2ban-slack-action.* [Online]
Available at: https://github.com/coleturner/fail2ban-slack-action
[Accessed 29 03 2025].

fujiapple852, 2025. *Trippy.* [Online]
Available at: https://github.com/fujiapple852/trippy
[Accessed 01 05 2025].

netfilter.org, 2024. *The netfilter.org project.* [Online]
Available at: https://netfilter.org/
[Accessed 28 04 2025].

nlnetlabs.nl, 2025. *Unbound Installation Guide.* [Online]
Available at: https://unbound.docs.nlnetlabs.nl/en/latest/getting-started/installation.html
[Accessed 29 04 2025].

nmap.org, 2025. *nmap home page.* [Online]
Available at: https://nmap.org/
[Accessed 30 04 2025].

Pi.Alert, 2025. *github.com.* [Online]
Available at: https://github.com/pucherot/Pi.Alert
[Accessed 30 04 2025].

Postman Downloads, 2025. *Postman Downloads.* [Online]
Available at: https://www.postman.com/downloads/
[Accessed 29 04 2025].

Richard Blum, 2022. *CompTIA Linux+ Study Guide, 5th Edition | Chapter 19: Embracing Security Best Practices.* [Online]
Available at: https://learning.oreilly.com/library/view/comptia-linux-study/9781119878940/c19.xhtml#head-2-418
[Accessed 30 04 2025].

Richard Blum, 2022. *CompTIA Linux+ Study Guide, 5th Edition | Chapter 24 Troubleshooting Application and Hardware Issues.* [Online]
Available at: https://learning.oreilly.com/library/view/comptia-linux-study/9781119878940/c24.xhtml
[Accessed 30 04 2025].

Richard Blum, 2022. *CompTIA Linux+ Study Guide, 5th Edition | Chapter 15.* [Online]
Available at: https://learning.oreilly.com/library/view/comptia-linux-study/9781119878940/c15.xhtml#head-2-327
[Accessed 30 04 2025].

Richard Blum, 2022. *ComptTIA Linux+ Study Guide, 5th Edition | Chapter 16: Looking at Access and Authentication Methods..* [Online]
Available at: https://learning.oreilly.com/library/view/comptia-linux-study/9781119878940/c16.xhtml
[Accessed 30 04 2025].

vxpse.blog, 2022. *Send message to teams channel with rest api or ansible.* [Online]
Available at: https://vexpose.blog/2022/11/29/send-message-to-teams-channel-with-rest-api-or-ansible/
[Accessed 29 04 2025].