

DVG001

**Introduction to Linux and Small Networks
Project: IPv6**

IPv6

av

Jonatan Rassekhnia

**Akademin för teknik och miljö
Avdelningen för industriell utveckling, IT och samhällsbyggnad**

Högskolan i Gävle
S-801 76 Gävle, Sweden

Contents

1. Introduction.....	1
1.1. Background	1
1.2. Purpose	1
2. Planning and implementation	1
2.1. Planning.....	1
2.2. Implementation.....	2
3. Description of results	9
4. Discussion.....	9
5. Conclusion	10
6. Reference	10
I. Annex(es).....	11
II. Next Attachment.....	13

1. Introduction

In today's evolving digital landscape, we are evolving rapidly. The following report is an adaption of IPv6 (Internet Protocol version 6) becoming increasingly crucial. The project is in depletion of the availed Ipv4 addresses, transitioning to IPv6 to accommodate the growing number of inter connected devices and ensure the continued expansion of the internet. Project IPv6 focuses on the implementation of an IPv6 connectivity even if our internet service provider does not natively support it. By creating an IPv6 tunnel, configuring an IPv6 router, and setting up a web server, I can explore the practical aspects of working with IPv6 and gain hands-on experience in enabling IPv6 connectivity in our local network. Throughout the project, I will follow my own planning by creating my own step-by-step instructions to successfully set up my IPv6 infrastructure, paving the way for more robust and future-proof network architecture.

1.1. Background

As technology continues to advance, the need for a robust, and scalable internet infrastructure becomes increasingly evident. IPv6, which is the latest versions of the internet protocol, offers a solution to the limitation posed by IPv4 to provide a vast pool of available addresses to accommodate the expanding of the network devices and services. However, the transition between IPv4 and IPv6 is to presents its own challenges, including the need to upgrade network equipment, configuration of routers, and to ensure the compatibilities across various systems. This project aims to address these challenges given by guiding myself through a process of setting up an IPv6 tunnel and configuring an IPv6 router. By leveraging services like tunnelbroker.net, I can establish an IPv6 connection even in the absence of the native IPv6 support from my internet provider.

1.2. Purpose

The purpose of this assignment is to understand and implement IPv6 connectivity for a web server environment. To complete this project IPv6 is to aim for a question: How can I enable and configure IPv6 connectivity for my web server, ensuring it is accessible and functional over the IPv6 protocol? Within this assignment, exploring such tasks such as setting up a tunnel broker, obtaining an IPv6 address, configuring network interfaces, enabling IPv6 support in services like Apache, and securing the environment with a firewall. By addressing these tasks, I strive to gain practical knowledge in effectively deploying and managing IPv6 connectivity in web server context. The goal and purpose are to ensure that my web server is accessible to clients both within the local network and from the wider IPv6-enabled internet, expanding it's reach and availability to a broader audience.

2. Planning and implementation

Planning and implementation are crucial aspects of any successful project, including the setup of an IPv6-enabled web server. The planning phrase, is to define the projects objectives, assess requirements, and devise a comprehensive strategy. This includes tasks such as selecting a suitable tunnel broker, obtaining an IPv6 address, and configuring network interfaces. Additionally, the planning does involve the implementation for the necessary changes that need to be made in service configurations, firewall settings, and DNS records to enable IPv6 connectivity. The implementation phase involves executing the planed steps, ensuring accurate configuration of services, verifying connectivity, and addressing any issues encountered. A systematic and well-executed approach to planning and implementation is vital for an effective deployment of IPv6 on the web server, ultimately enhancing the accessibility and reach.

2.1. Planning

Planning is a crucial part of any project, including the implementation that needs to take place. The

planning phase involved assessing the requirements for IPv6, defining the scope and determining the objectives of the project. It requires identifying the necessary resources, including hardware, software, and networking components, as well as considering any potential constraints or limitations. It involves creating a roadmap or timeline to guide the implementation process, outlining the key tasks, milestones, and dependencies. Through planning helped me ensure a systematic approach to the implementation, allowing for a smoother execution and minimizing the risk of errors or oversights. By investing the time and effort into the planning phase, I was able to lay a solid foundation for the successful implementation of the IPv6 web server and optimize our resources effectively.

Network Setup #1	Firewall #3	Website Access #5
Network Services #2	Web-Server Installation #4	

This road that I have constructed is a guideline for my planning on my implementation. Which I will explain my steps in the implementation section of this report.

2.2. Implementation

The implementation process is based off the roadmap which I provided earlier. I first had to do my search on setting up an IPv6 tunnel. From the tutorial guide I followed the steps given from Netgate Docs [1] on how to configure my IPv6 through a tunnel broker service, first i had to sign up to recieve that tunnel broker service provider, then i had to create a tunnel using the nearest server which is located in Stockholm. To see my tunnel broker information on what i have created you can check the annex section for illustration 1. I had to enter my IPv5 address of my Linux machine for the IPv4 end point.

The next step involves configuring the IPv6 tunnel, which i had to load the modprobe create the tunnel with my ip address, public and private then i had to set the link up, I followed these steps by using Ubuntu Wiki [2] and i had to follow a YouTube video to get a clear understanding of this current process with the setup, the channel named "Packt" has helped get through this process [3].

```
(jonatan@kali)-[~]  
$ sudo modprobe ipv6
```

```
(jonatan@kali)-[~]  
$ sudo ip tunnel add he-ipv6 mode sit remote [REDACTED] local 2[REDACTED] ttl 255
```

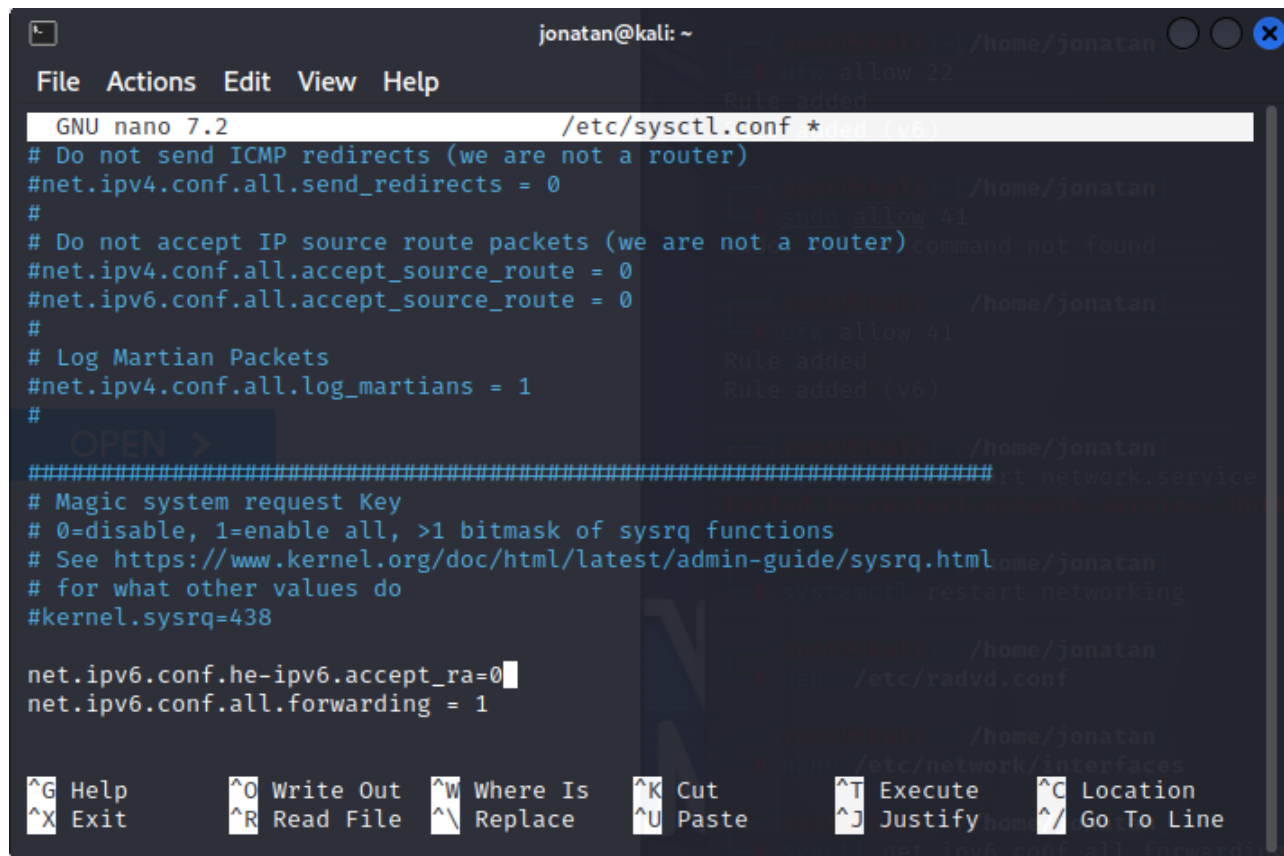
```
(root@kali)-[/home/jonatan]  
# ip link set he-ipv6 up
```

```
(jonatan@kali)-[~]  
$ sudo ip -6 addr change [REDACTED] 2/64 dev he-ipv6
```

```
(jonatan@kali)-[~]  
$ sudo ip route add ::/0 dev he-ipv6
```

After taking these steps, where my device and tunnel got created i was ready to move on to the next step. You can check my interface setup on illustration 2 in the annex section of this report.

The next following implementaiton is to edit the sysctl.conf file which is enabling IPv6 forwarding before i get into the router setup.



```
jonatan@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/sysctl.conf *  
# Do not send ICMP redirects (we are not a router)  
#net.ipv4.conf.all.send_redirects = 0  
#  
# Do not accept IP source route packets (we are not a router)  
#net.ipv4.conf.all.accept_source_route = 0  
#net.ipv6.conf.all.accept_source_route = 0  
#  
# Log Martian Packets  
#net.ipv4.conf.all.log_martians = 1  
#  
#####  
# Magic system request Key  
# 0=disable, 1=enable all, >1 bitmask of sysrq functions  
# See https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html  
# for what other values do  
#kernel.sysrq=438  
  
net.ipv6.conf.he-ipv6.accept_ra=0  
net.ipv6.conf.all.forwarding = 1  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Once i saved this, i started configuring my IPv6 router which is me assigning a static IPv6 address to the interface that connects to my local network. My tunnel interface is named eth0 and i assigned my IPv6 address. I guided myself within this webpage based on this article called "Setting up an IPv6 tunnel and router on Debian Squeeze with tunnelbroker.net" [4]. This helped set up my router, which my implementations took place installing *radvd*. I had to configure the conf file and setup the router.

```
root@kali: /home/jonatan
File Actions Edit View Help
GNU nano 7.2 /etc/radvd.conf
# IPv6
interface eth0
{
  AdvSendAdvert on;
  prefix [REDACTED]
  {
    AdvOnLink on;
  };
};
```

Once this setup was configured, I used the following command lines to start the service and enable it.

```
(root@kali)-[/home/jonatan]
# systemctl start radvd
```

```
(root@kali)-[/home/jonatan]
# systemctl enable radvd
Synchronizing state of radvd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable radvd
```

Then I had to give permission by using *chmod* to the *radvd* configuration file. Once everything was configured you can check illustration 1 for status of the *radvd* router in next attachment within this following report.

```
(jonatan@kali)-[~]
$ sudo chmod 644 /etc/radvd.conf
```

After all these steps I wanted to test my IPv6 connection if its actually working and functioning within my network and my service.

Test your IPv6 connectivity.

[Summary](#) [Tests Run](#) [Share Results / Contact](#) [Other IPv6 Sites](#) [For the Help Desk](#)

- Your IPv4 address on the public Internet appears to be [REDACTED] (BAHNHOF)
- Your IPv6 address on the public Internet appears to be [REDACTED] (HURRICANE)
- Since you have IPv6, we are including a tab that shows how well you can reach other IPv6 sites. [\[more info\]](#)
- Your DNS server (possibly run by your ISP) appears to have IPv6 Internet access.

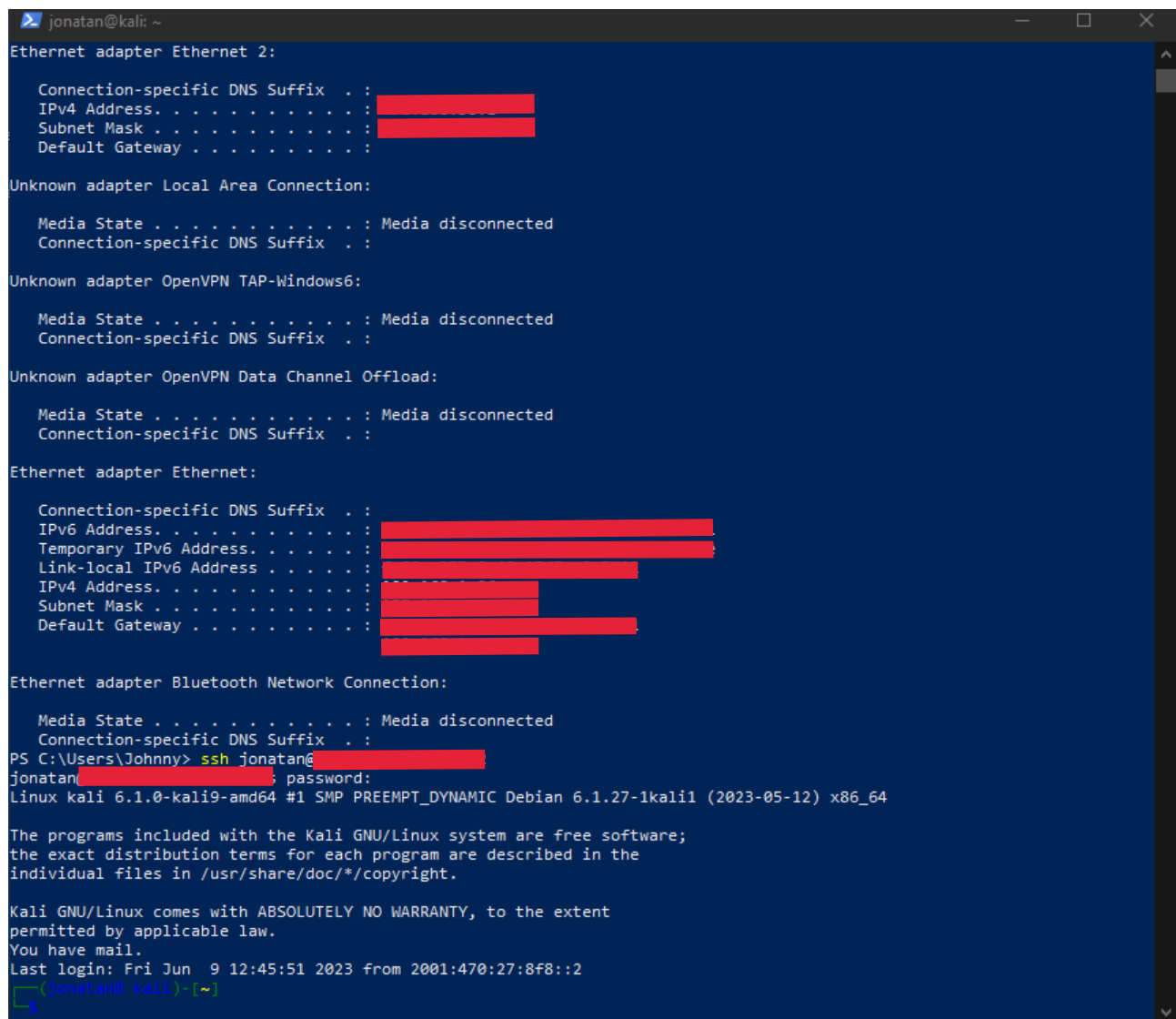
Your readiness score
10/10 for your IPv6 stability and readiness, when publishers are forced to go IPv6 only

[Click to see Test Data](#)
(Updated server side IPv6 readiness stats)

This mirror is provided by RoEduNet

Once I saw my IPv6 result, I wanted a more detailed version, so I ended up using another webpage called IPv6 test. You can check my illustration for a more detailed view in the annex section and its under illustration 3.

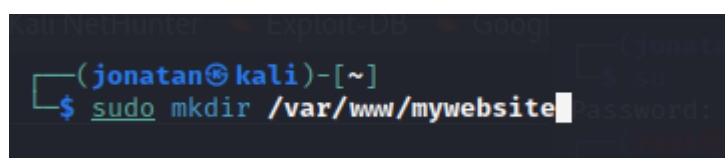
Then I wanted to test if it my IPv6 works remote on SSH connecting to my machine from another machine. This was a pure success connection as you can see my image on from Windows PowerShell.



```
jonatan@kali: ~  
Ethernet adapter Ethernet 2:  
    Connection-specific DNS Suffix  . :   
    IPv4 Address. . . . . :   
    Subnet Mask . . . . . :   
    Default Gateway . . . . . :   
  
Unknown adapter Local Area Connection:  
    Media State . . . . . : Media disconnected  
    Connection-specific DNS Suffix  . :   
  
Unknown adapter OpenVPN TAP-Windows6:  
    Media State . . . . . : Media disconnected  
    Connection-specific DNS Suffix  . :   
  
Unknown adapter OpenVPN Data Channel Offload:  
    Media State . . . . . : Media disconnected  
    Connection-specific DNS Suffix  . :   
  
Ethernet adapter Ethernet:  
    Connection-specific DNS Suffix  . :   
    IPv6 Address. . . . . :   
    Temporary IPv6 Address. . . . . :   
    Link-local IPv6 Address . . . . . :   
    IPv4 Address. . . . . :   
    Subnet Mask . . . . . :   
    Default Gateway . . . . . :   
  
Ethernet adapter Bluetooth Network Connection:  
    Media State . . . . . : Media disconnected  
    Connection-specific DNS Suffix  . :   
PS C:\Users\Johnny> ssh jonatan@  
jonatan@kali:~$  
Linux kali 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali1 (2023-05-12) x86_64  
  
The programs included with the Kali GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
You have mail.  
Last login: Fri Jun  9 12:45:51 2023 from 2001:470:27:8f8::2  
(jonatan@kali)-[~]  
$
```

This also provides my window IPv6 address, which I have not seen before, until I implemented this setup based on this project. It was a successful connection.

The final step of the implementation of this IPv6 project is to create a webserver by using the IPv6 address I have created so it can be accessed world-wide. With my implementation it was a successful route. The implementation is very similar to my pervious report in our pervious assignments so the setup was a lot quicker than usually. As you can see these are the following, I have taken place in creating the webserver than can be accessed worldwide.



```
(jonatan@kali)-[~]  
$ sudo mkdir /var/www/mywebsite
```

```
(jonatan@kali)-[~]
$ sudo chown -R www-data:www-data /var/www/ipv6/index.html
```

By learning this new access from the article "How do I give myself access to /var/www to create and edit files and folders in it without 'sudo'?", [5] which helped understand that www-data is giving you webserver access, Apache2 runs the www-data by default which makes the setup more simple.

```
(jonatan@kali)-[~]
$ sudo chmod -R 755 /var/www/he-ipv6
```

I have assigned the appropriate permissions to the directory and I got indicated that it was time to move on to the next step by creating an index.html file.

```
jonatan@kali: ~
File Actions Edit View Help
GNU nano 7.2 /var/www/mywebsite/index.html
!DOCTYPE html>
<html>
<head>
<title>Welcome to IPv6 Test Webpage</title>
</head>
<body>
<h1>IPv6</h1>
<p>This is a Test page for IPv6 Constructed by: Jonatan Rassekhnia</p>
</body>
</html>
```

Now the HTML file was created my next step is to configure the webserver to give access towards the IPv6 address and to have it tested from different people if it actually works worldwide.

```
jonatan@kali: ~
File Actions Edit View Help
GNU nano 7.2 /etc/apache2/sites-available/mywebsite.conf
<VirtualHost [2]
    ServerName j-ipv6
    ServerAlias www.ipv6project.com
    DocumentRoot /var/www/mywebsite

    <Directory /var/www/mywebsite>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



```
(jonatan@kali)-[~]  
$ sudo a2enmod rewrite
```

```
(jonatan@kali)-[~]  
$ sudo systemctl restart apache2
```

Once I enabled the rewrite module, I then needed to restart the Apache2 service then test my IPv6 on web browsers and not using my local network. Tested it on my phones network and also my own network. You can also check my functioning IPv6 address with the webserver I have constructed with this following link: [\[2001:172:27:23::23\]](http://[2001:172:27:23::23]), If you take a look at my screenshot using the 5G network from my phone provider you can see that my IPv6 web server does function, I also had different people test and everyone informed that it works.



The final results from requirements given from the project, are testing my IPv6 with various commands to retrieve the information to provide for my professor. I started out with checking ip -6 addr

```
(jonatan@kali)-[~]
$ ip -6 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1000
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
   inet6 [REDACTED]/64 scope global temporary dynamic
       valid_lft 86088sec preferred_lft 7498sec
   inet6 [REDACTED]/64 scope global temporary deprecated dynamic
       valid_lft 86088sec preferred_lft 0sec
   inet6 [REDACTED]/64 scope global temporary deprecated dynamic
       valid_lft 86088sec preferred_lft 0sec
   inet6 [REDACTED]/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86088sec preferred_lft 14088sec
   inet6 2[REDACTED]/64 scope global
       valid_lft forever preferred_lft forever
   inet6 [REDACTED]/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
17: he-ipv6@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1480 state UNKNOWN qlen 1000
   inet6 [REDACTED]/64 scope global
       valid_lft forever preferred_lft forever
   inet6 f[REDACTED]/64 scope link
       valid_lft forever preferred_lft forever
```

Then i had to check my ip -6 route

```
(jonatan@kali)-[~]
$ ip -6 route
::1 dev lo proto kernel metric 256 pref medium
[REDACTED] dev he-ipv6 metric 1024 pref medium
[REDACTED] dev eth0 proto kernel metric 256 pref medium
[REDACTED] dev he-ipv6 proto kernel metric 256 pref medium
[REDACTED] dev eth0 proto ra metric 100 pref medium
[REDACTED] he-ipv6 proto kernel metric 256 pref medium
[REDACTED] v eth0 proto kernel metric 1024 pref medium
default via [REDACTED] dev he-ipv6 metric 1024 onlink pref medium
```

Then i watned to ping my IPv6

```
(jonatan@kali)-[~]
$ ping6 -c4 [REDACTED]
PING 2[REDACTED] 56 data bytes
64 bytes from 2[REDACTED]: icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from 1[REDACTED]: icmp_seq=2 ttl=64 time=0.083 ms
64 bytes from [REDACTED]: icmp_seq=3 ttl=64 time=0.063 ms
^X^C
— 2001:470:27:8f8::2 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2046ms
rtt min/avg/max/mdev = 0.031/0.059/0.083/0.021 ms
```

Then i wanted to check nmap -6

```

(jonatan@kali)-[~]
$ nmap -6 [redacted] -A
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-12 10:33 CEST
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 10:33 (0:00:06 remaining)
Nmap scan report for tunnel855964-pt.tunnel.tserv24.sto1.ipv6.he.net ([redacted])
Host is up (0.00015s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2 (protocol 2.0)
| ssh-hostkey:
| [redacted] ecdsa
| [redacted] (ED25519)
80/tcp    open  http      Apache httpd 2.4.57 ((Debian))
|_ http-title: Welcome to IPv6 Test Webpage
|_ http-server-header: Apache/2.4.57 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.75 seconds

```

The last test is using the commands `traceroute6` and `tracepath6` to show my network path by the packets.

```

(jonatan@kali)-[~]
$ traceroute6 [redacted]
traceroute to [redacted] ([redacted]::2), 30 hops max, 80 byte packets
1 tunnel855964-pt.tunnel.tserv24.sto1.ipv6.he.net ([redacted]::2) 0.029 ms 0.006 ms 0.005 ms

```

I tested the IPv6 connectivity by pinging IPv6 addresses and verifying successful communication. Monitoring the tunnel status and troubleshooting was a big part of the task. The potential issues, which made me refer to the documentations to seek support from other users, Regularly I review and update the configuration is required to ensure the continued functionality and security. By implementing these steps, users can successfully connect to my webserver with the HTML file that has been constructed.

3. Description of results

The project focuses on the setup and configuration of an IPv6-enabled web server, with the objective of making it accessible both locally and from external networks. By following the step-by-step instructions, the concrete result of the assignment includes the successful establishment of an IPv6 tunnel connection, IPv6 address, configuration of Apache2 web server, and proper firewall settings. These results in the web server being reachable via it's IPv6 address, allowing users to access the website and its contents using IPv6-compatible devices and networks, mostly networks. The importance of planning and implementation, highlights the significance of proper network configuration, service settings, and security measures to ensure smooth operations and optimal accessibility of the web server.

4. Discussion

The results of the assignment demonstrated my successful implementation of an IPv6-enabled web server, enabling access to the website using IPv6 addresses. Through this process, I have gained valuable knowledge and experience in various aspects of IPv6 networking, including tunneling, addressing, configuring services, and managing firewalls. I have learned about the importance of proper planning and documentation in setting up a functional and secure web server. This assignment has deepened my understanding of IPv6 protocols and their role in modern internet importance. Highlighting the increasing importance of IPv6 adoption as IPv4 address became a challenge to achieve, but successfully configuring and accessing my web server over the IPv6, I have witnessed firsthand the potential for expanded connectivity and improved network performance offered by IPv6, I also have witnessed firsthand potential for expanded connectivity

and improved network performance offered by IPv6. This experience has broadened my skill set to prepare me for the real world evolving the networking needs in an IPv6-enabled world.

5. Conclusion

In conclusion, the purpose of this assignment was to implement an IPv6-enabled web server and assess its functionality and accessibility. By addressing the questions of whether, we can effectively deploy an IPv6 web server, I have achieved positive results. I have successfully configured and launched an Apache web server with IPv6 support, allowing external access to the website using IPv6 addresses. Through the process, I have gained hands-on experience in IPv6 networking, including addressing, tunneling, service configuration, and firewall management. I have also learned about the importance of proper planning, documentation, and troubleshooting in deploying an IPv6 infrastructure. The project has underscored the significance of IPv6 adoption in ensuring a continued growth and connectivity in the evolving internet landscape. By successfully setting up an IPv6 web server, I had demonstrated the ability to adapt the changing network requirements and contribute to the wider adoption of IPv6 technology.

6. Reference

Bibliography

- [1]: Configuring IPv6 Through A Tunnel Broker Service," Netgate Docs. Date of Article: Jul. 01, 2022 [Online]. Available: <https://docs.netgate.com/pfsense/en/latest/recipes/ipv6-tunnel-broker.html>. Accessed: Jul. 01, 2022.
- [2]: Ubuntu Wiki. "IPv6." Jan 26, 2016 [Online]. Available: https://wiki.ubuntu.com/IPv6#Get_connected_with_Hurricane_Electric. [Accessed: June, 11. 2023].
- [3]: Packt, "Linux Networking Solutions - Part 1: Setting Up an IPv6 Tunnel Via Hurricane Elect," YouTube, April 10, 2017. [Online]. Available: https://www.youtube.com/watch?v=bCp3F5Gw7xA&list=PLTgRMOcmRb3OAeT_qlg5BU-MNMsRcot59&index=3. [Accessed: June 08, 2023].
- [4]: J. S., "Setting up an IPv6 tunnel and router on Debian Squeeze with tunnelbroker.net," sevalecan, 2015. [Online]. Available: <https://www.sevalecan.com/debianipv6>. [Accessed: June 12, 2023].
- [5]: SoldierFox, "How do I give myself access to /var/www to create and edit files and folders in it without 'sudo'?", Apr. 19, 2017. [Online]. Available: <https://medium.com/@haquangvu/how-do-i-give-myself-access-to-var-www-to-create-and-edit-files-and-folders-in-it-without-sudo-ac93ca943a26>. [Accessed: June 10, 2023].

I. Annex(es)

Tunnel Details

IPv6 Tunnel

Example Configurations

Advanced

Tunnel ID: 855964

Creation Date:

May 31, 2023

Description:

IPv6

Delete Tunnel

IPv6 Tunnel Endpoints

Server IPv4 Address:

Server IPv6 Address:

Client IPv4 Address:

Client IPv6 Address:

Routed IPv6 Prefixes

Routed /64:

Routed /48:

Assign /48

DNS Resolvers

Anycast IPv6 Caching Nameserver:

Anycast IPv4 Caching Nameserver:

DNS over HTTPS / DNS over TLS:

ordns.he.net

rDNS Delegations

rDNS Delegated NS1:

rDNS Delegated NS2:

rDNS Delegated NS3:

rDNS Delegated NS4:

rDNS Delegated NS5:

Edit

Illustration 1: Tunnel broker Information

Sida 11

```
File Actions Edit View Help root@kali: /home/jonatan
File Actions Edit View Help /var/www/html
GNU nano 7.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

#The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

inet6 for eth0
iface eth0 inet6 static
address 2001:470:27:8f8::2
netmask 64

auto he-ipv6
iface he-ipv6 inet6 v4tunnel
address [REDACTED]
netmask 64
endpoint [REDACTED]
local [REDACTED]
ttl 255
mtu 1480
gateway 2 [REDACTED] 1
pre-up curl https://jkrass:4kx4V80i6d0H3eY-@ip4v.tunnelbroker.net/nic/update?hostname=855964

[ Read 33 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location ^M-U Undo
^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line ^M-E Redo
```

Illustration 2: Creation of Interface he-IPv6

The screenshot displays the IPv6 test results on the left and the network configuration on the right. The IPv6 test results show that both IPv4 and IPv6 connectivity are supported. The IPv4 connectivity section shows a score of 19/20, with IPv4 supported, address [REDACTED], hostname [REDACTED], and ISP Bahnhof AB. The IPv6 connectivity section shows a score of 19/20, with IPv6 supported, address 2001:470:27:8f8::2, type Native IPv6, and SLAAC No. The network configuration on the right shows the creation of the he-ipv6 interface using the v4tunnel method, with a static address, netmask, endpoint, local, ttl, mtu, gateway, and pre-up curl command.

Illustration 3: IPv6 Detailed

II. Next Attachment

```
(jonatan@kali)~  
$ sudo systemctl status radvd.service  
[sudo] password for jonatan:  
● radvd.service - Router advertisement daemon for IPv6  
   Loaded: loaded (/lib/systemd/system/radvd.service; enabled; preset: disabled)  
   Active: active (running) since Fri 2023-06-09 12:51:52 CEST; 2 days ago  
     Docs: man:radvd(8)  
  Main PID: 540543 (radvd)  
    Tasks: 2 (limit: 11350)  
  Memory: 492.0K  
     CPU: 761ms  
   CGroup: /system.slice/radvd.service  
           └─540543 /usr/sbin/radvd --logmethod stderr_clean  
             └─540544 /usr/sbin/radvd --logmethod stderr_clean  
  
Jun 09 12:51:52 kali systemd[1]: Starting radvd.service - Router advertisement daemon for IPv6...  
Jun 09 12:51:52 kali radvd[540540]: config file, /etc/radvd.conf, syntax ok  
Jun 09 12:51:52 kali radvd[540542]: version 2.19 started  
Jun 09 12:51:52 kali systemd[1]: Started radvd.service - Router advertisement daemon for IPv6.
```

Illustration 1: radvd status