

Atividade 6 - Revisão 2

Aluno: Marcos Beijner Ramso - 2149435

O trabalho a seguir trata da implementação de um cenário de teste que abrange os seguintes temas:

- Servidor DHCP: O DHCP (Dynamic Host Protocol) é um tipo de servidor usado para atribuir automaticamente as configurações de rede a dispositivos em uma rede IP. Ele permite que atributos como endereço IP, máscaras de sub-rede, gateways padrão, entre outros, sejam definidos automaticamente pelo servidor;

- Servidor HTTP e HTTPS: O servidor HTTP (HyperText Transfer Protocol) é um tipo de servidor que processa pedidos e respostas em formato de texto. Esse protocolo é amplamente utilizado na internet, especialmente na World Wide Web, permitindo que os clientes façam requisições de conteúdo por meio dele;

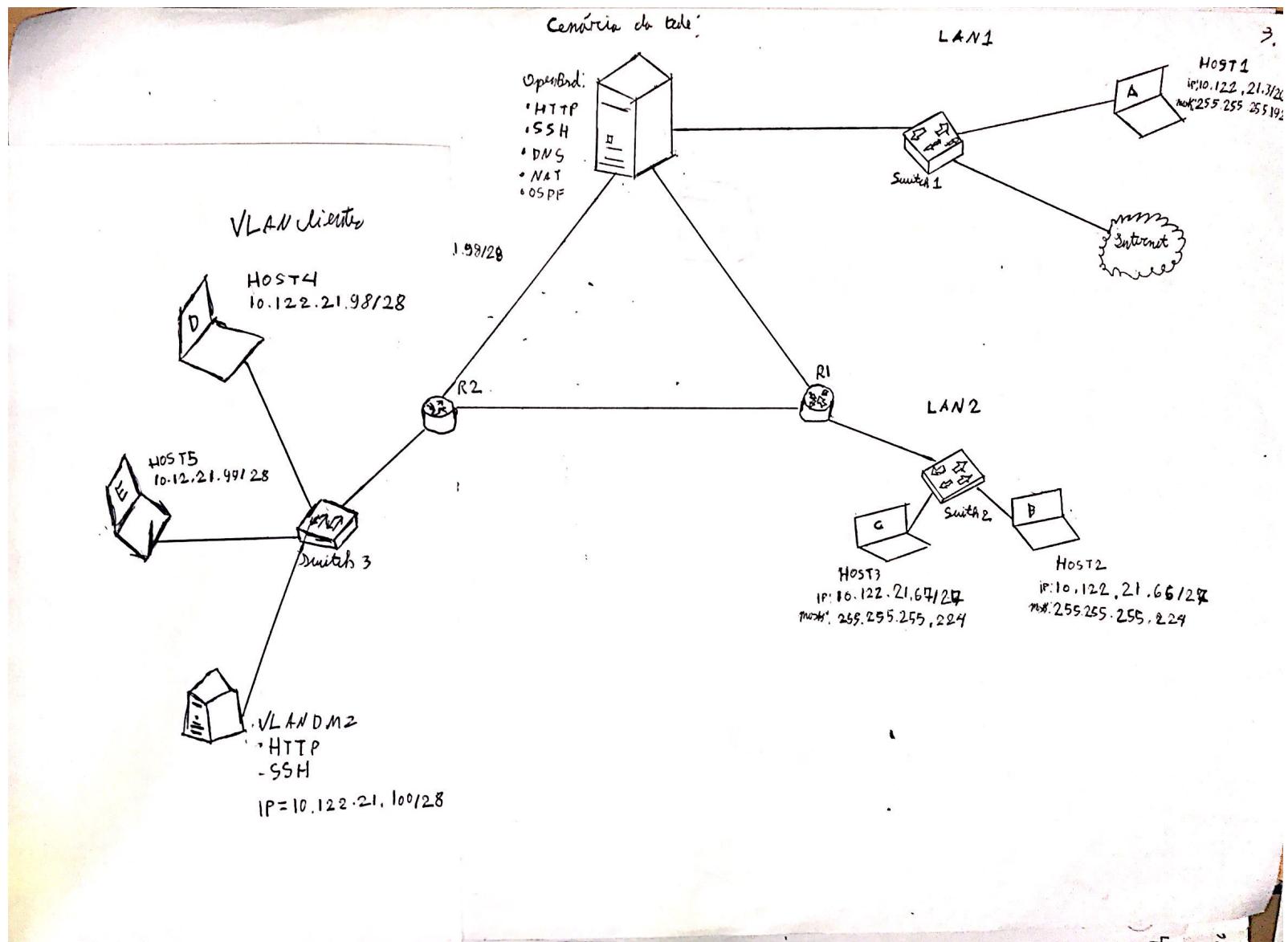
Por outro lado, o HTTPS adiciona uma camada adicional de segurança à comunicação entre o cliente e servidor. Ele utiliza a criptografia SSL/TLS, que protege a comunicação, garantindo a confidencialidade e a integridade das informações transmitidas, sendo especialmente importante para proteger dados sensíveis, como informações de login, transações financeiras e outros dados pessoais;

- Servidor FTP: O servidor FTP (File Transfer Protocol) é um protocolo que permite a transferência de arquivos entre um servidor e cliente. Esse protocolo utiliza do mesmo nome (FTP) para realizar a comunicação e permitir que os usuários façam upload e download de forma eficiente;

- Servidor SSH: O servidor SSH (Secure Shell) é um protocolo de rede seguro que possibilita a comunicação e a troca de mensagens entre um servidor e um cliente. Ele utiliza criptografia e autenticação para garantir a segurança da comunicação entre eles. O SSH é amplamente utilizado para acessar servidores remotamente de forma segura, permitindo assim o gerenciamento remoto de sistemas e a transferência segura de arquivos.

Soluções alternativas

- Servidor Samba: O Samba é um software cujo objetivo é facilitar a comunicação em ambientes heterogêneos, fornecendo serviços de compartilhamento de arquivos e impressoras para sistemas que utilizam o protocolo SMB/CIFS, como Linux, Windows e macOS;
- Servidor NFS: O NFS (Network File System) é um sistema de comunicação de arquivos entre cliente e servidor em sistemas Linux. através do NFS é possível acessar arquivos de uma máquina remota como se estivessem na máquina local;
- QoS: O protocolo Quality of Service (QoS) se refere a um conjunto de ferramentas e técnicas utilizadas para gerenciar o tráfego de rede. Essas técnicas visam garantir um uso justo e adequado dos recursos de rede, priorizando certos tipos de tráfego em detrimento de outros.



Implementando o cenário de Rede

Para configurarmos o cenário, é necessário configurar cada terminal independentemente, o GN53 nos permite modificar os IPs dos hosts, também como suas gateway, máscaras através da edição de configurações, os hosts ficaram com as seguintes configurações:

LAN1 - Slot 1:

IP → 10.122.21.3;
 Máscara → 255.255.255.192;
 Gateway → 10.122.21.2;

LAN2 - Slot 2:

IP → 10.122.21.66;
 Máscara → 255.255.255.224;
 Gateway → 10.122.21.65

LAN2 - Slot 3:

IP → 10.122.21.67
 Máscara → 255.255.255.224
 Gateway → 10.122.21.65

LAN3 - Slot 4:

IP → 10.122.21.98;
 Máscara → 255.255.255.240;
 Gateway → 10.122.21.97;

LAN3 - Slot 5:

IP → 10.122.21.99;
 Máscara → 255.255.255.240;
 Gateway → 10.122.21.97;

LAN 3 → Host Síncro

IP → 10.122.21.100;

Máscara → 255.255.255.240;

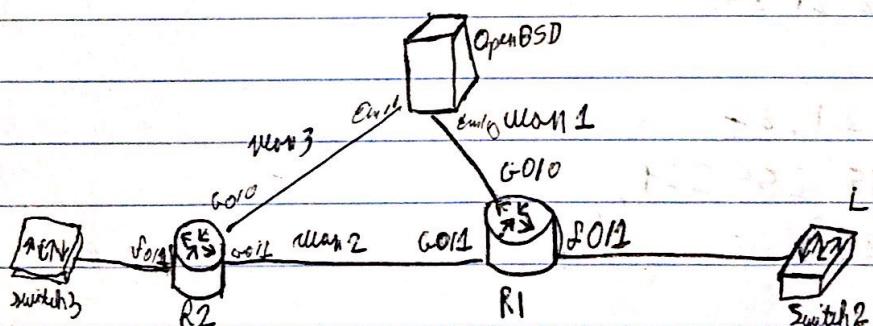
Gateway → 10.122.21.97.

Após configurar os hosts, é necessário configurar os roteadores Cisco. Para configurar os roteadores, é necessário abrir os terminais dos mesmos (R1 e R2) e imputar os seguintes comandos:

enable;

Configure terminal;

Após os comandos acima, acessamos o modo de configuração global do roteador. Após entrar no modo de configuração global, é necessário configurar as interfaces dos roteadores, para ilustrar a configuração, utilizarei o roteador R1, com suas configurações detalhadas abaixo



No cenário, os comandos ao roteador 1 (R1) serão dados por:

interface F0/1

ip address 10.122.21.65 255.255.255.224

interface G0/1

ip address 10.122.21.121 255.255.255.252

Interface G0/0

ip address 10.122.21.125 255.255.255.252

Com os comandos descritos temos a configuração das interfaces de conexão, devemos aplicar os mesmos ao roteador 2 (R2) modificando apenas os endereços de IP, no final, o cenário ficará com os ips:

R1 Configuração interface G0/1:

IP → 10.122.21.65;

Máscara → 255.255.255.252;

R1 configuração interface G0/0/1:

IP → 10.122.21.121;

Máscara → 255.255.255.252;

R1 configuração interface G0/0/0:

IP → 10.122.21.125;

Máscara → 255.255.255.252;

R2 configuração G0/1:

IP → 10.122.21.122;

Máscara → 255.255.255.252;

R2 configuração G0/1:

IP → 10.122.21.97;

Máscara → 255.255.255.240;

R2 configuração G0/0:

IP → 10.122.21.129;

Máscara → 255.255.255.252.

Após configurar as interfaces, torna-se necessário a configuração do protocolo OSPF, para isso é necessário imprimir os comandos:

Router 1:

Router ospf 100;

Router-id 1.1.1.1;

Network 10.122.21.120 0.0.0.31 area 0;

Network 10.122.21.124 0.0.0.3 area 0;

Network 10.122.21.120 0.0.0.3 area 0;

Os comandos formam a criação do OSPF com o nome sendo 100, e o identificador sendo 1.1.1.1, as linhas seguem o padrão de network < rede que soem da roteador > < máscara > < área do OSPF>, com a máscara sendo a máscara inversa da original. No roteador 2 (R2) temos:

Router 2:

Router ospf 200;

Router-id 2.2.2.2;

Network 10.122.21.120 0.0.0.31 area 0;

Network 10.122.21.128 0.0.0.3 area 0;

Network 10.122.21.96 0.0.0.15 area 0;

Com estes comandos, os roteadores estão agora configurados para trabalharem no protocolo OSPF. Note que será necessário ligar as interfaces utilizando o comando no shutdown e por último salvar as configurações com o comando copy-running config startup-config.

Após configurar os totadores, se faz necessário a configuração do OpenBSD, para tal, será feito primeiro a configuração do servidor HTTP Apache, os comandos serão feitos na ordem de procurar o pacote, instalar, verificar o serviço, iniciar o servidor, habilitar para boot e a criação dos host virtuais, segue os comandos feitos no OpenBSD:

1. Procurando o pacote:

```
# pkg_info -q httpd
```

2. Instalando o servidor:

```
# pkg_add apache httpd <versão do apache>
```

3. Iniciando o servidor:

```
# /etc/init.d/apache2 -f start
```

No caso da poso 3, executamos o servidor

diretamente pelo script de startup, após isso, tornar-se uma boa prática verificar se o servidor encontra-se em execução, através do comando:

```
# netstat
```

Verifique se o porto 80 está no modo

listen.

4. Habilitar o comando no-boot da máquina:

Para habilitar o Apache com o boot da máquina, utiliza-se do comando:

```
# kext enable apache2
```

5. Configurando os totos de teste

Temos configurar o Apache para operar no

19.10.122.21.113, para tal, utiliza-se

do editor de texto VI. Para acessar o

arquivo de configuração obtém do VI use-se:



vi /etc/apache2/ftppdr2.conf;

listen 10.122.21.113:80

Listen 10.122.21.113:8080

apache2 testar

6. Criando o site virtual:

É necessário entrar novamente no

arquivo de configuração e

descomentou a linha: `Include /etc/apache2/extra/httpd-vhosts.conf`

6.1 Criando o arquivo de hosts virtuais:

vi /etc/apache2/extra/httpd-vhosts.conf

<VirtualHost *:80>

DocumentRoot "/var/www/html/default"

ServerName marcola.com.br

</VirtualHost>

Com os comandos, o sitio "marcola.com.br" foi criado, porém o mesmo não possui nada, para adicioná-lo tais as
informações faremos:

mkdir -p /var/www/html/default

echo "Este sitio existe em algum lugar virtual">/var/www
/html/default/index.html

Criamos o diretório com o mkdir e adicionamos o texto ao
index.html, com isso, será impressa a mensagem ao usuário
quando o mesmo acessar marcola.com.br.

Após criar o HTTP, é hora configurar o SSH, para isso, temos de iniciar o servidor SSH e configura-lo.

1. Iniciando o SSH:

Fora inicializar o servidor, utilizo-se o comando:

```
# systemctl start sshd
```

2. Alterando o porta padrão:

Uma boa prática para evitá-los ataques é alterar a porta padrão, para tal, alteraremos a porta no Port 22 para Port 2222.

3. Criando um novo usuário:

Fora criarmos um novo usuário utilizando o comando:

```
# adduser obd
```

```
# password 123mudar
```

4. Mudando as configurações do ssh:

Utilizando do vi, abrimos o arquivo de configuração com o comando:

```
# vi /etc/ssh/sshd_config
```

4.1 alterações que devem ser feitas:

Dentro do arquivo temos necessário comentar

a linha "PermitRootLogin" e adiciona um "no"

exemplo: PermitRootLogin no.

Círculo da linha, deve-se adicionar:

AllowUsers obd

Feitos os comandos descritos acima, deve-se iniciar o SSH através do comando sudo /etc/init.d/ssh restart. Agora o usuário obd está configurado para acessar o ssh.



Após configurarmos o SSH, precisaremos configurar o servidor DNS, para isso, utilizaremos do BIND, para isso basta executar os seguintes comandos:

1. Instalando o BIND:

Para instalar o bind, utilizaremos o comando `pkg_info`

- O bind, isso vai procurar o pacote no OpenBSD. Após encontrá-lo o pacote que desejamos instalar, utilizaremos do comando `pkg_add` para adicionar o pacote.

Os comandos ficam, em ordem:

```
# pkg_info -a
```

```
# pkg_add tsx-bind <versão>
```

2. Verificando o status:

Para verificar o status do serviço, utilizando

da cláusula netstat com os parâmetros -a para

listar todos os conexões, -n para mostrar números de

índice e -p UDP, já que o BIND é executado

No protocolo UDP.

```
# netstat -a -n -p udp
```

3. Iniciando o BIND:

O comando para iniciar o serviço chama-se pelo

Prefixo do script, com os parâmetros start, stop,

restart, iniciando tudo como:

```
# /etc/rc.d/named start
```

4. Criando a zona no servidor

Basta abrir o arquivo de nome (`named.conf`) para

Mudar o setor ~~master~~ master.com.br, para isso

precumos:

```
# vi /var/named/etx/named.conf
```

```
zone "marcola.com.br" {
```

```
    type master;
```

```
    file "master/db.marcola.com.br"; 3;
```



4.1. Verificando se o arquivo está correto

Após modificar o arquivo, é necessário verificar se ele está de acordo com os parâmetros. Para isso, usar-se o comando:

```
# named -checkconf /var/named/etc/named.conf
```

5. Criando o arquivo de zona

Para configurar a zona, é uma boa (é) prática utilizar do arquivo modelo, assim não será necessário escrever do zero. Para isso utilizar-se de comando:

```
# cp /var/named/stdard/soahost /var/named/master/db.morolo.com.br
```

O comando copia o arquivo soahost do db.morolo.com.br

5.1 Editando o arquivo

Iremos agora editar o arquivo com o vi, para isso imputo - re:

```
# vi /var/named/master/morolo.com.br
```

Após isso, para editar a zona, iremos configurar o arquivo, para que fique:

\$ TTL 6h

@ IN SOA vm.morolo.com.br. nobody.ain. morolo.com.br.

1 ; serial

1h ; refresh

30m ; retry

7d ; expiration

1h) ; minimum

@ NS ns.morolo.com.br,

vm.morolo.com.br. IN A 10.122.21.113

Note que o DNS foi impedido de IP de slot Apache.



6. Criando o arquivo de zona reversa

- Depois de verificar o arquivo é necessário criar uma zona reversa, onde se o arquivo criado (zona de domínio) faz com que dada um nome de host decolhe o IP, a zona reversa faz o contrário, ou seja, dado o IP, decolhe o nome do host. Para isso, copiaremos o arquivo criado no passo 5 para /var/named/motu/~~db.113~~
db.21.122.10

Após copiarmos, mantremos a mesma configuração, alterando apenas a linha que contém:

vm.marcelo.com.br. IN A 10.122.21.113

111 IN PTR vm.marcelo.com.br.

Após isso basta ~~reiniciar~~ reiniciar com o comando:

/etc/init.d/named restart

Realizados os comandos acima, foram criados as zonas domínio e reversa do DNS marcelo.com.br, onde agora o servidor openBSD pode prover suporte.

Depois de configurado o Domain Name System é agora necessário configurar o servidor OpenBSD para operar também como um roteador, para atingir tal objetivo seguiremos a sequência de passos a seguir:

1. Adicionar o ponto ospfd

É necessário o ponto ospfd para executarmos o ospf no servidor, para tal, utilizo-se do comando:

```
# pkg_add ospfd
```

2. Alterar o arquivo de configuração

Após adicionar o ponto é necessário configurar o arquivo, para isso, utilizo-se do comando:

```
# vi /etc/ospfd.conf
```

Após alterar o arquivo é necessário editar o arquivo para ser compatível com o serviço de Iface, o arquivo dentro ficará:

area 0 {

 interface em0 {

 hello-interval 10;

 dead-interval 40;

 cost 1;

 authentication-type none;

 interface em1 {

 hello-interval 10;

 dead-interval 40;



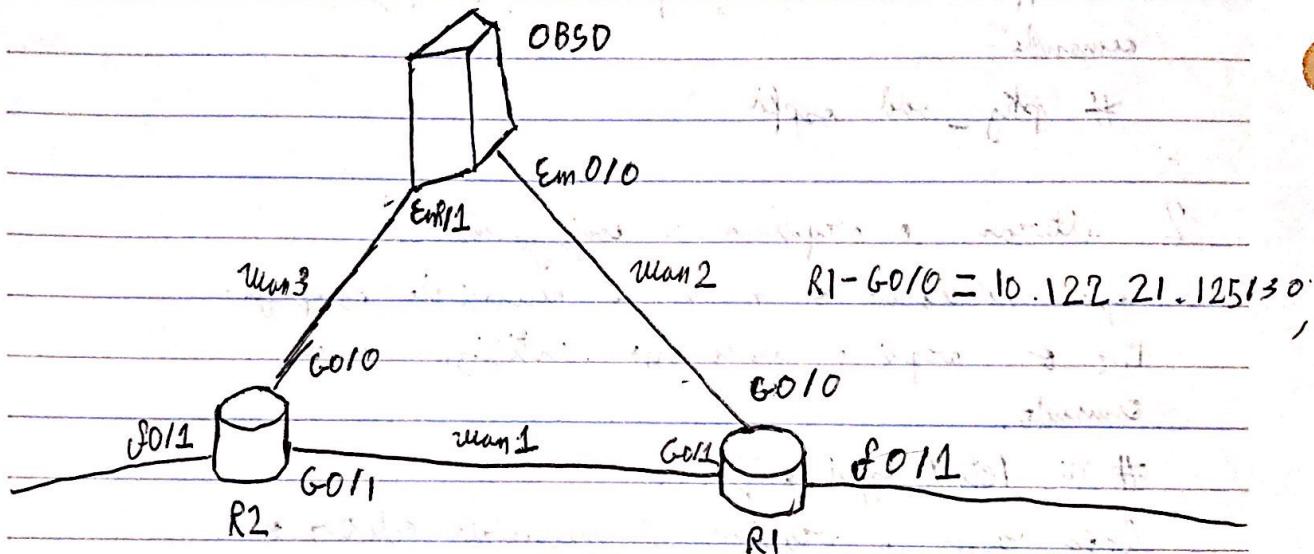
Cost 1;

authentication-type none;

3

3

Agora o OpenBSD foi configurado para operar na área 0, juntamente com os outros roteadores Cisco. Segue um esboço dos links:



$$R2 - \text{G}0/1 = 10.122.21.97/28;$$

$$R2 - \text{G}0/11 = 10.122.21.122/30;$$

$$R2 - \text{G}0/10 = 10.122.21.125/30;$$

$$R1 - \text{G}0/1 = 10.122.21.121/30;$$

$$R1 - \text{G}0/11 = 10.122.21.65/27;$$

Note que é necessário escolher os IPs de Em0/10 e Em0/11 para isso faremos o seguinte:

1. Criar o arquivo dos hostnames

Vou ilustrar o comando, os alterarão serão

fazidos no Em0/10. Cremos o arquivo através do

vi: # vi /etc/hostname.em0



(6) Dentro do arquivo, agora editar-se o init para:
inet 10.122.21.126 255.255.255.252 NONE

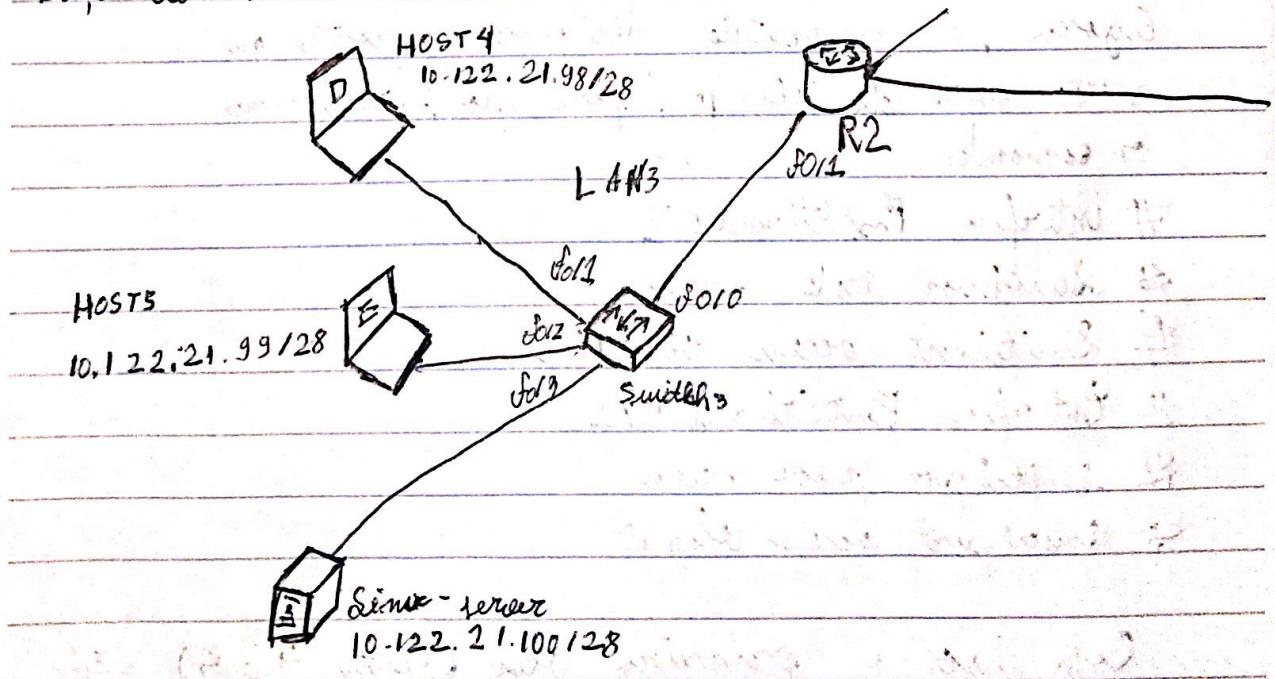
Repetindo o passo para o arquivo
let / testname, Em1:

inet 10-122.21.130 255.255.255.252 NONE

Com as alterações, agora o ospf está configurado no openBSD.

Configurando os VLANs

O cenário a ser configurado é o da LAN3, onde será feita a sublaciação de LANs; segue o esboço do sub-cenário:



Tudo configurar a VLAN, faremos os seguintes passos:

1. Entrar em modo configuração global

O primeiro passo é ser realizado é entrar e sair do Switch em modo de configuração global, para isso, utilizaremos:

#enable;

#configure terminal

2. Criando as VLAN

Após entrar o modo de configuração global, iremos criar a VLAN cliente através dos comandos:

vlan 10

nome Cliente

2.1. Adicionando as portas:

Agora, é necessário adicionar as portas que serão parte da VLAN 10, para isso, executaremos os comandos:

interface FastEthernet 0/1

switchport mode access

switchport access vlan 10

interface FastEthernet 0/2

switchport mode access

switchport access vlan 10

Com isso a primeira VLAN (Slots 4 e 5) está criada. Toda a VLAN DMZ, faremos o seguinte:

3. Configurando a VLAN DMZ

Us comandos para criar a segunda VLAN segùo:

```
# vlan 20
```

```
# nome DMZ
```

```
# interface FastEthernet 0/3
```

```
# switchport mode access
```

```
# switchport access vlan 20
```

4. Configurando o porto trunk

Agora é necessário configurar o porto que irá conectar todos as VLANs ao roteador

R2, o nome disto é trunk, logo configurar o porto trunk realizando os comandos:

```
# interface FastEthernet 0/0
```

```
# switchport mode trunk
```

```
# switchport trunk allowed vlan 10,20
```

Com isso a VLAN foi concluída em seu todo. Note que ao final deve-se salvar com o comando:

```
# copy running-config startup-config
```

Configurando o servidor

Linux

A última tarefa do cenário é configurar o servidor Linux, ele deve possuir um servidor HTTP e um SSH, para isso bemos utilizar novamente o Apache.

1. Instalando o Apache

O primeiro passo para configurar o HTTP é instalar o Apache, em um Linux é feito através do comando:

```
# sudo apt update
```

```
# sudo apt install apache2
```

2. Iniciando o Apache

Torna iniciar o Apache basta executar o script com o parâmetro start

```
# apache2ctl start
```

2. 1 Ativando o Apache no boot

Torna o servidor ligar sempre com a máquina, é necessário ativar no boot com o comando:

```
# systemctl enable apache2
```

3. Configurando o host de teste

Agora é necessário configurar o Apache para operar no IP do Host - Linux, para isso acessa-se o arquivo de configurações através do vi e editar as linhas Sisten, veja exemplo:

```
# vi /etc/apache2/httpd2.conf
```

```
Listen 10.122.21.100:80
```

```
Listen 10.122.21.100:8080
```

Após configurados as linhas, tente o serviço:

```
# apache2ctl test
```

4. Criando o host virtual.

Dentro do arquivo de configurações é necessário descomentar a linha: Include /etc/apache2/extra/httpd-vhosts.conf

5. Configurando a sua versão

```
# vi /etc/apache2/sites-available/000-default.conf
```

Alteraremos o arquivo para criar o site

Linux.com, ficando assim:

```
<VirtualHost *:80>
```

```
DocumentRoot "/var/www/html/docs"
```

```
ServerName linux.com.br
```

```
</VirtualHost>
```

Agora o site foi criado, adicionaremos seu nome e frase "Olá seu servidor Linux", através dos comandos:

```
# mkdir -p /var/www/html/default
```

```
# echo "Olá seu servidor Linux" > /var/www/html/default/index.html
```

Com estes comandos o servidor linux.com agora existe no seu computador HTTP. Agora vamos configurar o SSH; para isso realizaremos os passos:

1. Instalando o SSH

No Linux, nem sempre o SSH está incluso, por conta disto execute o comando de instalação:

```
# sudo apt update
```

```
# sudo apt install openssh-server
```

Após instalar, verifique o status

```
# systemctl status ssh
```

2. Criando o usuário

Como requisitado, deve-se criar um usuário "Linux"

com a senha "123mudar", para isso execute:



sudo useradd -m -s /bin/bash lime

sudo passwd lime

Após inserir e confirmar o senha "123 mudar", o usuário lime foi criado, para acessar o ssh através do mesmo, utiliza-se o comando:

ssh lime 10.122.21.100.

Conclusão

No final do trabalho, é evidente que o conhecimento para criar e manter cenários de todos é muito desafiador, principalmente quando se trata da esta quantidade de comandos e protocolos que devem ser seguidos.

Quando tratamos de OpenBSD é evidente a quantidade de bibliotecas e o enorme suporte que este sistema operacional possui os programadores. A dificuldade do mesmo encontra-se em entender o funcionamento de todos os bibliotecas disponíveis e como utilizá-las.

O Linux foi o mais fácil de se encontrar tutoriais na internet, visto que o mesmo possui uma grande comunidade e documentações online. No entanto o Linux, embora com muitas bibliotecas, não foi criado com tanta ênfase em segurança e recursos quanto a OpenBSD, logo, caso eu fosse criar um servidor em um sistema UNIX, utilizaria aídeia do OpenBSD.

Foi último, tivemos os ciso, estes foram os mais fáceis de trabalhar. É notável a padronização de seus terminais, facilitando o uso, além da vasta documentação que possuem.

Esta atividade foi extremamente completa e difícil, pois em elas trouxe consigo todos os conhecimentos adquiridos na disciplina, desde conceitos teóricos à manutenção de servidores.

III

córes OpenBSD. através da atividade pude concluir que o conhecimento técnico de um profissional de rede é vasto e desafiador.