

Trabalho Integrador Temático

Gabriel Alexandre Foletto¹, Jardel Batista Gonçalves¹

¹Colégio Técnico Industrial – Universidade Federal de Santa Maria (UFSM)
Caixa Postal 5071 – 97110-970 – Santa Maria – RS – Brasil

{gabriel.foletto, jardel.goncalves}@redes.ufsm.br

Abstract. *This report presents the complete implementation of a network project for the Technical Industrial College (CTISM). We used the CORE software to simulate the network and all the equipment requested by the client (teacher). The work includes the creation of VPN, DHCP, and DNS servers, among others, as well as the implementation of firewall. The entire logical network design was developed to incorporate all necessary services according to the client's specifications.*

Resumo. *Este relatório apresenta a implementação completa de um projeto de rede para o Colégio Técnico Industrial (CTISM). Utilizamos o software CORE para simular a rede e todos os equipamentos solicitados pelo cliente (professor). O trabalho inclui a criação de servidores DHCP, DNS, entre outros, além da implementação de firewall. Todo o projeto lógico da rede foi desenvolvido de forma a incorporar a maioria dos serviços necessários, conforme as especificações do cliente.*

1. Introdução

Em ambientes corporativos e institucionais, é comum a necessidade de implementar uma infraestrutura de rede completa, composta por diversos serviços e equipamentos que atendam às necessidades da organização. Esses ambientes são robustos e exigem uma configuração detalhada para garantir uma operação eficiente, assegurando boa usabilidade, segurança, desempenho e disponibilidade.

A complexidade desses ambientes exige a integração de múltiplos serviços e equipamentos, como servidores, roteadores, *switches*, *firewalls*, e dispositivos de armazenamento. Cada um desses elementos deve ser cuidadosamente configurado para garantir que o desempenho da rede seja otimizado, a segurança das informações seja mantida, e a disponibilidade dos serviços seja ininterrupta.

Este relatório detalha as etapas e considerações necessárias para a implementação de uma infraestrutura de rede completa, conforme as especificações do cliente (professor). O documento destaca e desenvolve as melhores práticas e tecnologias empregadas para garantir um ambiente seguro, eficiente e altamente disponível.

2. Objetivos

Para obter-se o conhecimento mínimo sobre como implementar um projeto de rede, o presente trabalho tem como objetivos:

- Criar e dividir logicamente as sub-redes para o projeto proposto;

- Implementar o protocolo de roteamento OSPFv2 para endereços IPv4 e os protocolos OSPFv3 e RIPng para endereços IPv6;
- Configurar um servidor DHCP para atribuição automática de endereços IPv4 e IPv6;
- Configurar um servidor DNS para receber atualizações dinâmicas de registros IPv4 a partir do servidor DHCP, enquanto os registros IPv6 são atualizados dinamicamente via *script*;
- Configurar um servidor DNS secundário (*slave*) para aumentar a disponibilidade da rede;
- Adicionar um *firewall* de filtragem de pacotes na borda da rede utilizando *iptables* e *ip6tables*;
- Implementar um método para controlar o acesso à rede das impressoras e às redes dos laboratórios por meio de *scripts*.

3. Metodologia

Com a finalidade de alcançar os objetivos estipulados na Seção 2, foi desenvolvido em ordem cronológica as tarefas que serão executadas para garantir o sucesso da atividade proposta.

- **Tempo 01:** Desenvolver um rascunho do projeto lógico e dividir as sub-redes;
- **Tempo 02:** Criar o cenário do rascunho utilizando o *software* CORE para simulação;
- **Tempo 03:** Instalar os softwares *Isc-dhcp-server* e *bind9* para atuarem como servidor DHCP e servidor DNS, respectivamente;
- **Tempo 04:** Configurar os servidores DHCP e DNS (primário e secundário);
- **Tempo 05:** Criar *scripts* para a atualização dinâmica de registros IPv6 no servidor DNS, devido à limitação do CORE em atualizar dinamicamente para IPv6;
- **Tempo 06:** Implementar a base do *firewall* de filtragem de pacotes na borda da rede;
- **Tempo 07:** Controlar inicialmente a rede das impressoras utilizando *iptables* e *ip6tables*;
- **Tempo 08:** Desenvolver *scripts* para o controle dos laboratórios, permitindo que os professores bloqueiem ou desbloqueiem a conexão com a rede global de computadores de cada laboratório;
- **Tempo 09:** Criar um *script* para liberar ou bloquear o acesso às impressoras, tanto de uma sub-rede específica quanto de um *host* específico;
- **Tempo 10:** Realizar testes para comprovar a eficiência dos recursos utilizados.

4. Desenvolvimento

Nesta seção será apresentado os principais pontos realizados em uma ordem cronológica para o desenvolvimento da atividade.

4.1. Desenvolvimento do Rascunho do Projeto e Criação do Cenário no CORE

O primeiro passo para o desenvolvimento do projeto foi elaborar um rascunho detalhado da subdivisão da rede do CTISM. Isso facilitou significativamente a criação do cenário no CORE, pois a divisão das sub-redes já estava definida. A Figura 1 apresenta o rascunho inicial utilizado.

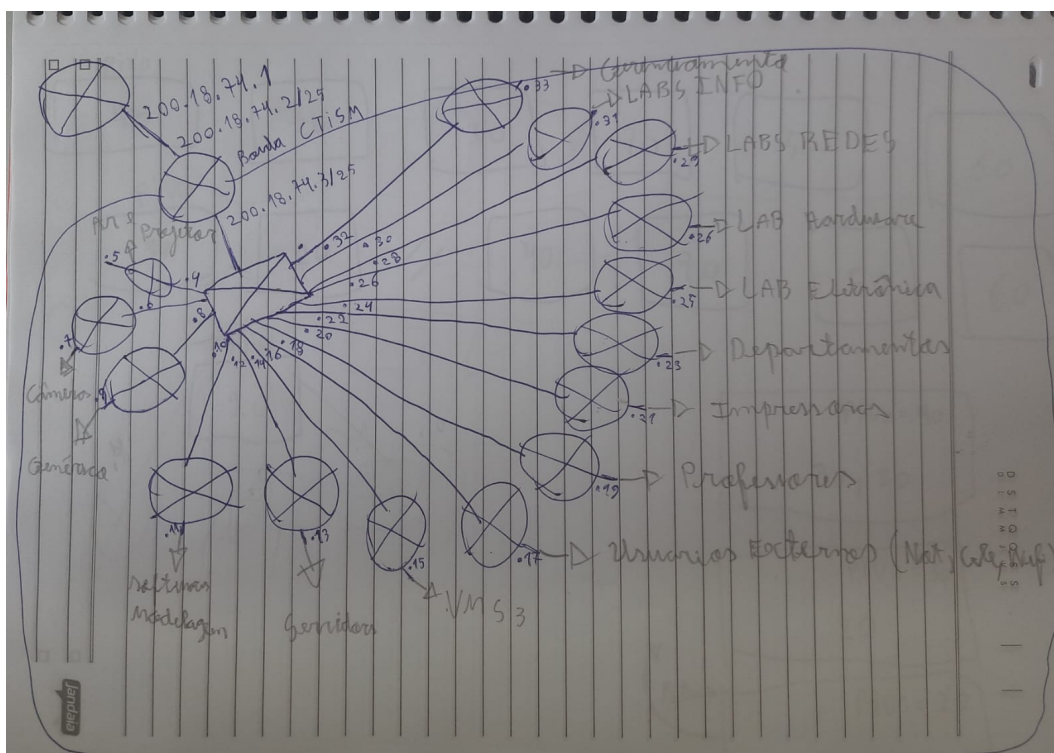


Figura 1: Rascunho do projeto lógico da rede do CTISM [Autores 2024].

Após a elaboração do rascunho, foi necessário passar os componentes para o ambiente de emulação onde foi desenvolvido o projeto. A Figura 2 mostra o cenário do CORE.

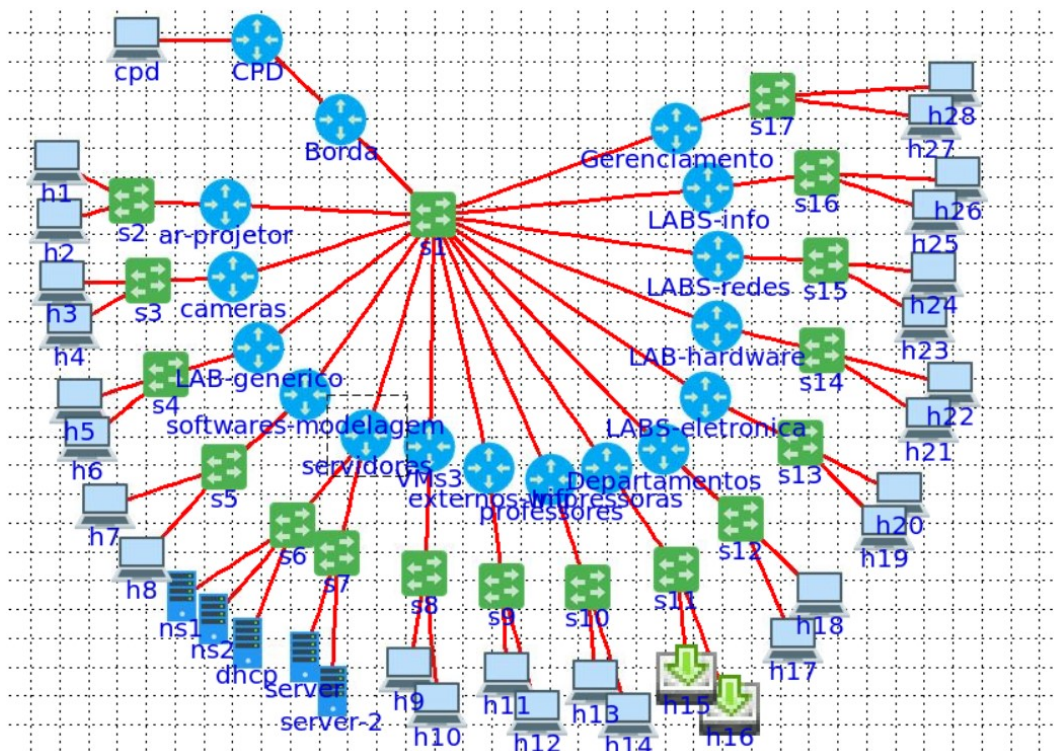


Figura 2: Cenário desenvolvido no CORE [Autores 2024].

A rede do CTISM foi dividida em 18 subredes, que são:

- **Conexão com gateway:** A primeira rede abordada é a rede ponto-a-ponto que conecta o roteador de borda do CTISM ao roteador do CPD. Seu prefixo CIDR para IPv4 é 200.18.74.0/30, enquanto o prefixo CIDR para IPv6 é 2804:1f38:2001::/64.
- **Roteadores:** Esta rede conecta os roteadores ao *switch* centralizador (s1), estabelecendo a ligação entre a borda do CTISM e os outros roteadores. O prefixo CIDR para IPv4 é 200.18.74.32/27, e o prefixo CIDR para IPv6 é 2804:1f38:2001:100::/56.
- **Ar-projetor:** Rede destinada ao gerenciamento de ar-condicionados e projetores em 40 salas, totalizando pelo menos 80 endereços IP. O prefixo CIDR para IPv4 é 172.16.1.0/24, e o prefixo CIDR para IPv6 é 2804:1f38:2001:200::/56.
- **Cameras:** Rede destinada ao gerenciamento de 100 câmeras de vigilância IP. O prefixo CIDR para IPv4 é 172.16.2.0/24 e o prefixo CIDR para IPv6 é 2804:1f38:2001:300::/56.
- **Lab-generico:** Rede destinada a computadores de propósito geral que não estão vinculados a nenhum laboratório específico. O prefixo CIDR para IPv4 é 172.16.3.0/24 e o prefixo CIDR para IPv6 é 2804:1f38:2001:400::/56.
- **Softwares-modelagem:** Rede destinada a servidores de licenças de *software*. O prefixo CIDR para IPv4 é 172.16.4.0/24 e o prefixo CIDR para IPv6 é 2804:1f38:2001:500::/56.
- **Servidores (internos):** Rede destinada aos servidores acessíveis apenas internamente, como servidores DNS e DHCP. O prefixo CIDR para IPv4 é 172.16.5.0/24, e o prefixo CIDR para IPv6 é 2804:1f38:2001:600::/56.
- **Servidores (externos):** Rede destinada aos servidores acessíveis externamente, incluindo a intranet do CTISM e o sistema web do Vms3. O prefixo CIDR para IPv4 é 200.18.74.64/27, e o prefixo CIDR para IPv6 é 2804:1f38:2001:700::/56.
- **Vms3:** Rede destinada às máquinas do Vms3, onde alunos e professores criarão suas máquinas virtuais. O prefixo CIDR para IPv4 é 172.16.16.0/20, e o prefixo CIDR para IPv6 é 2804:1f38:2001:800::/56. O uso de um prefixo /20 permite endereçar até 4094 dispositivos com endereços IPv4 privados, acomodando a criação extensiva de máquinas por usuários.
- **Externos-wifi:** Rede destinada à conexão de dispositivos móveis via Wi-Fi, como *notebooks*, *smartphones* e outros equipamentos móveis. O prefixo CIDR para IPv4 é 172.16.32.0/20, e o prefixo CIDR para IPv6 é 2804:1f38:2001:900::/56. A escolha de uma faixa ampla de endereços reflete a alta demanda atual por conectividade móvel.
- **Professores:** Rede dedicada aos professores do CTISM. O prefixo CIDR para IPv4 é 172.16.6.0/23, e o prefixo CIDR para IPv6 é 2804:1f38:2001:a00::/56. O prefixo /23 permite o endereçamento de até 510 dispositivos, acomodando a possibilidade de múltiplas máquinas virtuais por professor.
- **Impressoras:** Rede destinada às impressoras, acessível principalmente pela rede dos professores e dos Departamentos. O acesso pode ser controlado através de *scripts*. O prefixo CIDR para IPv4 é 172.16.8.0/24, e o prefixo CIDR para IPv6 é 2804:1f38:2001:b00::/56.
- **Departamentos:** Rede destinada aos servidores do CTISM. O prefixo CIDR para IPv4 é 172.16.9.0/24, e o prefixo CIDR para IPv6 é 2804:1f38:2001:C00::/56.

- **Labs-eletronica:** Rede destinada a quatro laboratórios de eletrônica. O prefixo CIDR para IPv4 é 172.16.10.0/24, e o prefixo CIDR para IPv6 é 2804:1f38:2001:d00::/56.
- **Lab-hardware:** Rede específica para o único laboratório de hardware do CTISM. O prefixo CIDR para IPv4 é 172.16.11.0/24, e o prefixo CIDR para IPv6 é 2804:1f38:2001:e00::/56.
- **Labs-redes:** Rede destinada a dois laboratórios de redes de computadores. O prefixo CIDR para IPv4 é 172.16.12.0/24, e o prefixo CIDR para IPv6 é 2804:1f38:2001:f00::/56.
- **Labs-info:** Rede destinada a seis laboratórios de informática. Cada laboratório deve acomodar cerca de 40 computadores, o que justifica a utilização de um prefixo que suporte um número maior de dispositivos. O prefixo CIDR para IPv4 é 172.16.14.0/23, e o prefixo CIDR para IPv6 é 2804:1f38:2001:1000::/56.
- **Gerenciamento:** Rede destinada ao gerenciamento de *switches*, roteadores e servidores. Embora o CORE não suporte a implementação de VLANs, essa rede seria configurada em uma VLAN separada para facilitar o gerenciamento isolado desses dispositivos, simplificando a manutenção e a correção de problemas. O prefixo CIDR para IPv4 é 172.16.48.0/20, e o prefixo CIDR para IPv6 é 2804:1f38:2001:1100::/56.

4.2. Utilizando os protocolos de roteamento OSPFv2, OSPFv3 e RIPng

Ao adicionar um roteador no CORE, ele já vem com os protocolos de roteamento OSPFv2 e OSPFv3 ativados. No entanto, o protocolo OSPFv3 não estava funcionando corretamente sozinho. Para garantir o pleno funcionamento da rede IPv6, foi necessário adicionar o protocolo de roteamento RIPng com redistribuição de rotas para o OSPFv3, assegurando assim a operação adequada da rede IPv6.

O CORE, via *script*, já configura automaticamente os protocolos de roteamento, portanto, foi necessário apenas adicionar o RIPng para o funcionamento correto da rede IPv6. A Figura 3 apresenta a configuração dos protocolos de roteamento no roteador de borda da rede. Vale lembrar que todos os roteadores do projeto utilizaram esses protocolos.

```

interface eth0
    ip address 200.18.74.2/30
    ipv6 address 2804:1f38:2001::2/64
    ip ospf network point-to-point
    ip ospf hello-interval 2
    ip ospf dead-interval 6
    ip ospf retransmit-interval 5
!
interface eth1
    ip address 200.18.74.48/27
    ipv6 address 2804:1f38:2001:100::10/56
    ip ospf network point-to-point
    ip ospf hello-interval 2
    ip ospf dead-interval 6
    ip ospf retransmit-interval 5
!

router ospf6
    instance-id 65
    router-id 200.18.74.2
    interface eth0 area 0.0.0.0
    interface eth1 area 0.0.0.0
!
router ripng
    redistribute static
    redistribute connected
    redistribute ospf6
    network ::/0
!
router ospf
    router-id 200.18.74.2
    network 200.18.74.2/30 area 0
    network 200.18.74.48/27 area 0
!

```

Figura 3: Configuração dos protocolos de roteamento no roteador de borda
[Autores 2024].

4.3. Configurando o servidor DHCP e o servidor DNS

Para garantir a funcionalidade da rede, foi necessário instalar e configurar um servidor DHCP e um servidor DNS. Inicialmente, foram instalados os *softwares* apropriados: o *isc-dhcp-server* para o serviço DHCP e o *Bind9* para o serviço DNS.

Para realizar a configuração do servidor DHCP para IPv4 foi necessário editar o arquivo `/etc/dhcp/dhcpd.conf` e adicionar a *subnet* e atribuição de endereços para cada rede utilizada, conforme apresenta a Figura 4.


```

#AR-PROJETOR
subnet 172.16.1.0 netmask 255.255.255.0 {
    option routers 172.16.1.1;

    host h1 {
        hardware ethernet 00:00:00:aa:00:01;
        fixed-address 172.16.1.10;
    }

    host h2 {
        hardware ethernet 00:00:00:aa:00:02;
        fixed-address 172.16.1.20;
    }

    #ADICIONAR TODOS OS HOSTS DA REDE COMO FEITO ACIMA
}

```

Figura 4: Definição da *subnet* para a rede ar-projetor [Autores 2024].

Naturalmente, essas configurações precisaram ser aplicadas a cada sub-rede do projeto. Em seguida, foi necessário definir uma chave HMAC-MD5 para permitir a atualização dos registros A e PTR no servidor DNS, além de adicionar as zonas que o servidor DHCP tem permissão para modificar. A Figura 5 apresenta o cabeçalho do arquivo de configuração do DHCP, incluindo a chave HMAC-MD5 e as zonas que podem ser alteradas por ele. Cada zona que o DHCP atualiza deve ser adicionada a este arquivo.

```

log-facility local6;
default-lease-time 3600;
max-lease-time 7200;
ddns-update-style interim;
ddns-domainname "jbgoncalvess.predes.ufsm.br";
option domain-name "jbgoncalvess.predes.ufsm.br";
ignore client-updates;
authoritative;
ddns-updates on;
update-static-leases on;

key "dhcp_updater" {
    algorithm hmac-md5;
    secret "Sbf4X+AWB16R3jBqQgdJvA==";
};

zone jbgoncalvess.predes.ufsm.br. {
    primary 172.16.5.2; # IP do servidor DNS
    key dhcp_updater;
}

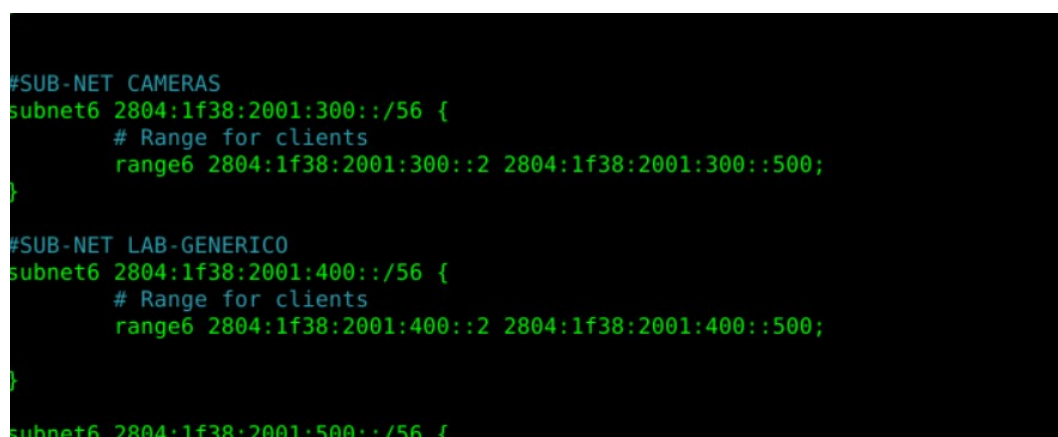
zone 1.16.172.in-addr.arpa. {
    primary 172.16.5.2; # IP do servidor DNS
    key dhcp_updater;
}

```

Figura 5: Definição do arquivo de configuração do servidor DHCP [Autores 2024].

Vale lembrar que o `ddns-updates` deve estar como `on` e o `ddns-update-style` deve estar como `interim` ou `standard`.

No servidor DHCPv6, como a atualização dinâmica dos registros não funcionou, foi necessário apenas definir as sub-redes. A adição dos registros no DNS será realizada por meio de um *script* que será apresentado a seguir. A Figura 6 mostra a definição das sub-redes “cameras” e “lab-generico” para IPv6.



```
#SUB-NET CAMERAS
subnet6 2804:1f38:2001:300::/56 {
    # Range for clients
    range6 2804:1f38:2001:300::2 2804:1f38:2001:300::500;
}

#SUB-NET LAB-GENERICO
subnet6 2804:1f38:2001:400::/56 {
    # Range for clients
    range6 2804:1f38:2001:400::2 2804:1f38:2001:400::500;
}

subnet6 2804:1f38:2001:500::/56 {
```

Figura 6: Definição da *subnet* para a rede “cameras” e “lab-generico” [Autores 2024].

Foi necessário fazer o processo apresentado na Figura 5 para as demais *subnets* também.

Tanto para IPv4 quanto para IPv6, foi necessário utilizar a ferramenta *Dhcrelay*. Esta ferramenta permite que as solicitações dos clientes sejam encaminhadas para a rede do servidor DHCP e, conseqüentemente, cheguem ao próprio servidor DHCP.

Por exemplo, no roteador da sub-rede das câmeras, foram adicionados os seguintes comandos: `dhcrelay -i eth1 -i eth0 172.16.5.4` para IPv4 e `dhcrelay -6 -l eth1 -u 2804:1f38:2001:600::4%eth0` para IPv6. Em ambos os comandos, é necessário especificar o endereço do servidor DHCP para que, ao sair da rede (solicitação `dhclient`), os próximos roteadores saibam qual é o próximo salto.

Foi necessário adicionar esses comandos em todos os roteadores de sub-redes diferentes da sub-rede do servidor DHCP, para permitir que os hosts dessas sub-redes adquirissem um endereço IP, tanto IPv4 quanto IPv6.

A configuração do servidor DNS foi um pouco mais complexa, exigindo a criação de vários arquivos e a alteração de diversos parâmetros. Primeiramente, foi necessário modificar o arquivo “`named.conf.options`”, adicionando a linha `listen-on { any; };`. Além disso, para adicionar os logs do servidor DNS, foram incluídas configurações adicionais no mesmo arquivo, conforme mostrado na Figura 7.


```

        directory "/var/log";
    };

    logging {
        channel abacate.log {
            file "/var/log/dns.log";
            // Set the severity to dynamic to see all the debug messages.
            severity info;
        };
        category default {
            abacate.log;
        };
    };
};

```

Figura 7: Adição de log no DNS [Autores 2024].

Em seguida, foi necessário modificar o arquivo “named.conf.local”, adicionando as zonas de tradução. Isso inclui a zona direta `jbgoncalvess.predes.ufsm.br` e as zonas reversas, que correspondem às sub-redes invertidas. Por exemplo, a zona reversa da rede “ar-projetor” é `1.16.172.in-addr.arpa`.

A Figura 8 mostra o arquivo de configuração com a declaração da zona direta e de algumas zonas reversas. As zonas reversas devem ser adicionadas conforme o número de redes presentes. Nota-se que a chave para a comunicação e atualização dos registros do DNS pelo DHCP também está presente nesse arquivo.

```

//CHAVES DHCP'S PARA ATUALIZAÇÕES DINÂMICAS
key "dhcp_updater" {
    algorithm hmac-md5;
    secret "Sbf4X+AWB16R3jBqQgdJvA==";
};

//TRADUÇÃO DIRETA
zone jbgoncalvess.predes.ufsm.br {
    type master;
    file "/etc/bind/db.jbgoncalvess.predes.ufsm.br";
    allow-query{"any";}
    allow-update{ key "dhcp_updater"; key "dhcp_updater_v6"; };
};

//TRADUÇÃO REVERSA (ar-projetor)
zone 1.16.172.in-addr.arpa IN {
    type master;
    file "/etc/bind/db.172.16.1.0";
    notify no;
    allow-query{"any";}
    allow-update{ key "dhcp_updater"; };
};

//TRADUÇÃO REVERSA (cameras)
zone 2.16.172.in-addr.arpa IN {
    type master;
    file "/etc/bind/db.172.16.2.0";
    notify no;
    allow-query{"any";}
    allow-update{ key "dhcp_updater"; };
};

```

Figura 8: Configuração do arquivo “named.conf.local” [Autores 2024].

Após definir as zonas, é necessário criar um arquivo de configuração para cada uma delas. Por exemplo, o arquivo de configuração para a zona `jbgoncalvess.predes.ufsm.br` é “`db.jbgoncalvess.predes.ufsm.br`”. As configurações desse arquivo são apresentadas na Figura 9.

```
$TTL 86400      ; 1 day
jbgoncalvess.predes.ufsm.br IN SOA ns1.jbgoncalvess.predes.ufsm.br. email.jbgoncalvess.predes.ufsm.br. (
    2024061246 ; serial
    3600       ; refresh (1 hour)
    1800       ; retry (30 minutes)
    1209600    ; expire (2 weeks)
    86400      ; minimum (1 day)
)
NS ns1.jbgoncalvess.predes.ufsm.br.
$ORIGIN jbgoncalvess.predes.ufsm.br.
ar-projetor      A      200.18.74.33
                  AAAA   2804:1f38:2001:100::1
borda            A      200.18.74.48
                  AAAA   2804:1f38:2001:100::10
cameras          A      200.18.74.34
                  AAAA   2804:1f38:2001:100::2
departamentos    A      200.18.74.42
                  AAAA   2804:1f38:2001:100::a
dhcp             A      172.16.5.2
                  AAAA   2804:1f38:2001:600::4
$TTL 1800       ; 30 minutes
dns              A      172.16.5.2
                  TXT    "00faed2ddd6c75bd86f60866737ca7dd6a"
dns-2            A      172.16.5.3
                  TXT    "00c083114e7323f6726475ce06d85125f9"
```

Figura 9: Arquivo de configuração da zona de tradução direta [Autores 2024].

Neste arquivo, são adicionados os registros A e AAAA para cada dispositivo. A sintaxe do arquivo é frequentemente modificada devido às atualizações dinâmicas realizadas pelo servidor DHCP.

Em seguida, foi configurado um arquivo de configuração para cada zona reversa. A Figura 10 mostra o arquivo chamado “`db.172.16.10.0`”, que corresponde à configuração da zona reversa `10.16.172.in-addr.arpa`, referente à sub-rede dos laboratórios de eletrônica.

```
$TTL 86400      ; 1 day
10.16.172.in-addr.arpa IN SOA ns1.jbgoncalvess.predes.ufsm.br. email.jbgoncalvess.predes.ufsm.br. (
    2024061007 ; serial
    3600       ; refresh (1 hour)
    1800       ; retry (30 minutes)
    1209600    ; expire (2 weeks)
    86400      ; minimum (1 day)
)
NS ns1.jbgoncalvess.predes.ufsm.br.
$ORIGIN 10.16.172.in-addr.arpa.
$TTL 1800       ; 30 minutes
2 PTR h19.jbgoncalvess.predes.ufsm.br.
3 PTR h20.jbgoncalvess.predes.ufsm.br.
```

Figura 10: Arquivo de configuração da zona reversa `10.16.172.in-addr.arpa` [Autores 2024].

Para zonas reversas IPv6, a declaração é um pouco diferente, pois é necessário escrever os dígitos hexadecimais do endereço IPv6 em ordem inversa, seguido de `ip6.arpa`. A Figura 11 apresenta a declaração de uma zona reversa IPv6 no arquivo “`na-med.conf.local`”.

```
//TRADUÇÃO REVERSA GERAL IPV6 (todas sub-redes)
zone "1.0.0.2.8.3.f.1.4.0.8.2.ip6.arpa" {
    type master;
    file "/etc/bind/db.2804:1f38:2001";
    notify no;
    allow-query{"any";}
    allow-update{ key "dhcp_updater_v6"; };
};
```

Figura 11: Declaração de zona reversa IPv6 [Autores 2024].

Dentro do arquivo de configuração da zona, denominado “db.2804:1f38:2001”, são adicionados os registros reversos de cada *host*. Se o recurso \$ORIGIN não for utilizado, é necessário escrever todos os 32 dígitos hexadecimais (128 bits) para representar os endereços e seus ponteiros reversos. A Figura 12 apresenta o arquivo de configuração da zona reversa 1.0.0.2.8.3.f.1.4.0.8.2.ip6.arpa.

[illegible]

Figura 12: Arquivo de configuração da zona reversa IPv6 [Autores 2024].

As configurações do servidor secundário foram praticamente idênticas às do primário, com algumas exceções. O tipo do servidor secundário é definido como `slave`, e no arquivo “`named.conf.local`”, foi adicionada a diretiva `masters { 172.16.5.2; };`, que especifica o servidor mestre responsável por suas atualizações.

4.3.1. Criação do *Script* para Atualização dos Registros IPv6

Como os registros IPv6 não são atualizados dinamicamente no CORE, mesmo com as configurações corretas, foi necessário criar um *script* para adicionar os registros IPv6 para tradução direta. A tradução reversa não foi implementada para os *hosts*, pois é mais relevante para roteadores. As Figuras 13 e 14 apresentam o *script* utilizado.

```
#!/bin/bash

#Arquivo com os nomes e os da leases
leases_file="/var/lib/dhcp/dhcpd6.leases"

#leases_file="/tmp/pycore.1/dhcp.conf/var.lib.dhcp/dhcpd6.leases"
saida="hostnames_ipv6.txt"
dns_file="/etc/bind/db.jbgoncalvess.predes.ufsm.br"

#Pego os endereços IPv6 da leases do dhcpv6
ipv6_addresses=$(grep -oP 'iaaddr \K[^\s]+' "$leases_file")

#Verifico se os endereços foram encontrados
if [ -z "$ipv6_addresses" ]; then
    echo "Nenhum endereço IPv6 encontrado no arquivo de leases."
    exit 1
fi

#Teste
echo "Endereços IPv6 encontrados:"
echo "$ipv6_addresses"

#Credenciais de cada host
ssh_user="core"
ssh_password="87654321"

#Ler cada endereço IPv6 e obter o hostname via SSH
for ipv6 in $ipv6_addresses; do
    echo "Conectando-se a $ipv6..."

    #Obter o hostname via SSH
    hostname=$(sshpass -p "$ssh_password" ssh -o ConnectTimeout=10 -o StrictHostKeyChecking=no $ssh_user@$ipv6 hostname 2>/dev/null)
```

Figura 13: *Script* utilizado para atualizar os registros IPv6 (parte 1) [Autores 2024].

```
#Obter o hostname via SSH
hostname=$(sshpass -p "$ssh_password" ssh -o ConnectTimeout=10 -o StrictHostKeyChecking=no $ssh_user@$ipv6 hostname 2>/dev/null)

if [ -z "$hostname" ]; then
    hostname="Não foi possível obter o hostname"
fi

if [ "$hostname" = "ns1" ]; then
    ipv6_ns1=$ipv6
fi

# Salvar o resultado no formato desejado
echo "$hostname AAAA $ipv6" >> "$saida"

echo "Hostname para $ipv6: $hostname"
done

echo "Resultados salvos em: $saida"
cat $saida

hostnames=$(awk '{print $1}' "$saida")
echo "$hostnames"

for host in $hostnames; do
    sed -i '/^'"$host"' \s+AAAA\s/d' "$dns_file"
done

cat "$saida" >> "$dns_file"

#REINICIAR O SERVIDOR DNS PARA ATUALIZAR NA HORA
sshpass -p "$ssh_password" ssh -o ConnectTimeout=10 -o StrictHostKeyChecking=no $ssh_user@$ipv6 ns1 sudo pkill named 2>/dev/null
sshpass -p "$ssh_password" ssh -o ConnectTimeout=10 -o StrictHostKeyChecking=no $ssh_user@$ipv6 ns1 sudo named -c /etc/bind/named.conf 2>/dev/null

rm -f /home/core/Desktop/controle/$saida
echo "FINALIZOU"
```

Figura 14: *Script* utilizado para atualizar os registros IPv6 (parte 2) [Autores 2024].

4.4. Implementação do iptables e ip6tables na Borda da Rede

Na rede do CTISM, a maioria dos recursos deve ser acessível apenas para usuários internos. No entanto, a rede dos servidores externos, que inclui a página da intranet, o site do VMs3, entre outros recursos, precisa estar acessível pela Internet. Para atender a essas necessidades, foram criadas regras para gerenciar o acesso, conforme apresenta a Figura 15.


```
#!/bin/bash

iptables -F FORWARD
ip6tables -F FORWARD

iptables -P FORWARD DROP
ip6tables -P FORWARD DROP

iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
ip6tables -A FORWARD -i eth1 -o eth0 -j ACCEPT

iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
ip6tables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

#Liberar servidores acessíveis externamente
iptables -A FORWARD -d 200.18.74.64/27 -i eth0 -o eth1 -j ACCEPT
ip6tables -A FORWARD -d 2804:1f38:2001:700::/56 -i eth0 -o eth1 -j ACCEPT
```

Figura 15: *Script* de inicialização para o *firewall* da borda da rede [Autores 2024].

4.5. Implementação do iptables e ip6tables na rede das impressoras

As impressoras devem ser prioritariamente utilizadas pelos professores e pelos funcionários concursados da universidade, que utilizam a rede “departamentos”. A rede dos “servidores” também deve ter acesso a essa rede, pois há *scripts* que se conectam aos dispositivos dessa rede para coletar informações. A Figura 16 mostra a implementação inicial desse controle de acesso às impressoras, utilizando iptables e ip6tables.

```
#!/bin/bash

#SOMENTE PROFESSORES,DEPARTAMENTOS E SERVIDORES TEM ACESSO AS IMPRESSORAS
iptables -F FORWARD
ip6tables -F FORWARD

iptables -P FORWARD DROP
ip6tables -P FORWARD DROP

iptables -A FORWARD -i eth0 -o eth1 -s 172.16.6.0/23 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -d 172.16.6.0/23 -j ACCEPT

ip6tables -A FORWARD -i eth0 -o eth1 -s 2804:1f38:2001:a00::/56 -j ACCEPT
ip6tables -A FORWARD -i eth1 -o eth0 -d 2804:1f38:2001:a00::/56 -j ACCEPT

iptables -A FORWARD -i eth0 -o eth1 -s 172.16.9.0/24 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -d 172.16.9.0/24 -j ACCEPT

ip6tables -A FORWARD -i eth0 -o eth1 -s 2804:1f38:2001:c00::/56 -j ACCEPT
ip6tables -A FORWARD -i eth1 -o eth0 -d 2804:1f38:2001:c00::/56 -j ACCEPT

iptables -A FORWARD -i eth0 -o eth1 -s 172.16.5.0/24 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -d 172.16.5.0/24 -j ACCEPT

ip6tables -A FORWARD -i eth0 -o eth1 -s 2804:1f38:2001:600::/56 -j ACCEPT
ip6tables -A FORWARD -i eth1 -o eth0 -d 2804:1f38:2001:600::/56 -j ACCEPT
```

Figura 16: *Script* de inicialização para o controle da rede das impressoras [Autores 2024].

Em seguida, foi desenvolvido um *script* para liberar ou bloquear o acesso à rede das impressoras a partir de uma sub-rede, conforme as necessidades do administrador da rede. O *script* permite ou bloqueia tanto sub-redes e *hosts* IPv4 quanto sub-redes e *hosts* IPv6, de acordo com o endereço passado como parâmetro. As Figuras 17 e 18 apresentam o *script*.

```
#!/bin/bash

echo "Digite um endereço IPv4 para liberar o acesso à impressora:"
echo "ATENÇÃO: O endereço pode ser um host ou uma sub-rede no formato CIDR."

read ip

#Verificar se e sub-rede ipv6
n=$(echo "$ip" | rev | cut -c 3)
echo "$n"

# ip de host
if echo "$ip" | grep -P '^d{1,3}\.d{1,3}\.d{1,3}\.d{1,3}$' > /dev/null; then
    REGRA_UM="-A FORWARD -s $ip/32 -i eth0 -o eth1 -j ACCEPT"
    REGRA_DOIS="-A FORWARD -d $ip/32 -i eth1 -o eth0 -j ACCEPT"
    REGRA_UM_DEL="-D FORWARD -s $ip/32 -i eth0 -o eth1 -j ACCEPT"
    REGRA_DOIS_DEL="-D FORWARD -d $ip/32 -i eth1 -o eth0 -j ACCEPT"

    iptables-save | grep -- "$REGRA_UM" > /dev/null
    if [ $? -ne 0 ]; then
        iptables $REGRA_UM
        iptables $REGRA_DOIS
        echo "USUÁRIO $ip COM ACESSO LIBERADO AS IMPRESSORAS."
    else
        iptables $REGRA_UM_DEL
        iptables $REGRA_DOIS_DEL
        echo "USUÁRIO $ip COM ACESSO RETIRADO AS IMPRESSORAS."
    fi
fi

# ip de sub-rede
elif echo "$ip" | grep -P '^d{1,3}\.d{1,3}\.d{1,3}\.d{1,3}/d{1,2}$' > /dev/null; then
    REGRA_UM="-A FORWARD -s $ip -i eth0 -o eth1 -j ACCEPT"
    REGRA_DOIS="-A FORWARD -d $ip -i eth1 -o eth0 -j ACCEPT"
    REGRA_UM_DEL="-D FORWARD -s $ip -i eth0 -o eth1 -j ACCEPT"
    REGRA_DOIS_DEL="-D FORWARD -d $ip -i eth1 -o eth0 -j ACCEPT"

    iptables-save | grep -- "$REGRA_UM" > /dev/null
    if [ $? -ne 0 ]; then
        iptables $REGRA_UM
        iptables $REGRA_DOIS
        echo "SUB-REDE $ip COM ACESSO LIBERADO AS IMPRESSORAS."
    else
        iptables $REGRA_UM_DEL
        iptables $REGRA_DOIS_DEL
        echo "SUB-REDE $ip COM ACESSO RETIRADO AS IMPRESSORAS."
    fi
fi
```

Figura 17: *Script* para a liberação das impressoras (parte 1) [Autores 2024].

```
# sub-rede ipv6
elif [ "$n" = "/" ]; then
    REGRA_UM="-A FORWARD -s $ip -i eth0 -o eth1 -j ACCEPT"
    REGRA_DOIS="-A FORWARD -d $ip -i eth1 -o eth0 -j ACCEPT"
    REGRA_UM_DEL="-D FORWARD -s $ip -i eth0 -o eth1 -j ACCEPT"
    REGRA_DOIS_DEL="-D FORWARD -d $ip -i eth1 -o eth0 -j ACCEPT"

    ip6tables-save | grep -- "$REGRA_UM" > /dev/null
    if [ $? -ne 0 ]; then
        ip6tables $REGRA_UM
        ip6tables $REGRA_DOIS
        echo "SUB-REDE $ip COM ACESSO LIBERADO AS IMPRESSORAS."
    else
        ip6tables $REGRA_UM_DEL
        ip6tables $REGRA_DOIS_DEL
        echo "SUB-REDE $ip COM ACESSO RETIRADO AS IMPRESSORAS."
    fi
fi

#host ipv6
else
    REGRA_UM="-A FORWARD -s $ip/128 -i eth0 -o eth1 -j ACCEPT"
    REGRA_DOIS="-A FORWARD -d $ip/128 -i eth1 -o eth0 -j ACCEPT"
    REGRA_UM_DEL="-D FORWARD -s $ip/128 -i eth0 -o eth1 -j ACCEPT"
    REGRA_DOIS_DEL="-D FORWARD -d $ip/128 -i eth1 -o eth0 -j ACCEPT"

    ip6tables-save | grep -- "$REGRA_UM" > /dev/null
    if [ $? -ne 0 ]; then
        ip6tables $REGRA_UM
        ip6tables $REGRA_DOIS
        echo "USUÁRIO $ip COM ACESSO LIBERADO AS IMPRESSORAS."
    else
        ip6tables $REGRA_UM_DEL
        ip6tables $REGRA_DOIS_DEL
        echo "USUÁRIO $ip COM ACESSO RETIRADO AS IMPRESSORAS."
    fi
fi
```

Figura 18: *Script* para a liberação das impressoras (parte 2) [Autores 2024].

4.6. Uso do *Script* para Controle de Acesso à Internet nos Laboratórios

Atendendo a uma das solicitações do cliente, que desejava controlar o acesso à Internet de cada laboratório, foi desenvolvido um *script* para essa função. A Figura 19 mostra o *script*. Note-se que a Figura apresenta os comandos apenas para os dois primeiros laboratórios, mas o *script* foi implementado para todos os laboratórios.

```
#!/bin/bash

echo "DIGITE O VALOR PARA O LABORATÓRIO QUE DESEJA BLOQUEAR/DESBLOQUEAR A CONEXÃO COM A INTERNET:"
echo "1 - INFORMÁTICA 01\n2 - INFORMÁTICA 02\n3 - INFORMÁTICA 03\n4 - INFORMÁTICA 04\n5 - INFORMÁTICA 05\n6 - INFORMÁTICA 06\n7 - REDES 01\n8 - REDES 02\n9 - HARDWARE 01\n10 - MODELAGEM 01\n11 - MODELAGEM 02\n12 - ELETRÔNICA 01\n13 - ELETRÔNICA 02"
read opcao

case $opcao in
    1)
        REGRA="-A FORWARD -i eth1 -o eth0 -m iprange --src-range 172.16.14.2-172.16.14.41 -j DROP"
        REGRA6="-A FORWARD -i eth1 -o eth0 -s 2804:1f38:2001:1000::/56 -j DROP"
        REGRA_DEL="-D FORWARD -i eth1 -o eth0 -m iprange --src-range 172.16.14.2-172.16.14.41 -j DROP"
        REGRA6_DEL="-D FORWARD -i eth1 -o eth0 -s 2804:1f38:2001:1000::/56 -j DROP"

        iptables-save | grep -- "$REGRA" > /dev/null

        if [ $? -ne 0 ]; then
            iptables $REGRA
            ip6tables $REGRA6
            echo "LABORATÓRIO DE INFORMÁTICA 01 BLOQUEADO."
        else
            iptables $REGRA_DEL
            ip6tables $REGRA6_DEL
            echo "LABORATÓRIO DE INFORMÁTICA 01 DESBLOQUEADO PARA INTERNET."
        fi
    ;;
    2)
        REGRA="-A FORWARD -i eth1 -o eth0 -m iprange --src-range 172.16.14.42-172.16.14.81 -j DROP"
        REGRA6="-A FORWARD -i eth1 -o eth0 -s 2804:1f38:2001:1000::/56 -j DROP"
        REGRA_DEL="-D FORWARD -i eth1 -o eth0 -m iprange --src-range 172.16.14.42-172.16.14.81 -j DROP"
        REGRA6_DEL="-D FORWARD -i eth1 -o eth0 -s 2804:1f38:2001:1000::/56 -j DROP"

        iptables-save | grep -- "$REGRA" > /dev/null

        if [ $? -ne 0 ]; then
            iptables $REGRA
            ip6tables $REGRA6
            echo "LABORATÓRIO DE INFORMÁTICA 02 BLOQUEADO."
        else
            iptables $REGRA_DEL
            ip6tables $REGRA6_DEL
            echo "LABORATÓRIO DE INFORMÁTICA 02 DESBLOQUEADO PARA INTERNET."
        fi
    ;;
    *)
        ;;
esac
```

Figura 19: *Script* para o controle da conexão externa dos laboratórios [Autores 2024].

5. Testes e Resultados

Nesta seção, serão realizados testes e apresentados os resultados em relação ao funcionamento da rede.

5.1. DHCP, DHCPv6 e DNS

Para demonstrar que os serviços DHCP e DHCPv6 estão funcionando corretamente, são exibidos os arquivos de *leases* de ambos os serviços, conforme mostrado nas Figuras 20 e 21.

```

authoring-byte-order little-endian;

lease 172.16.32.2 {
  starts 1 2024/07/22 18:52:40;
  ends 1 2024/07/22 19:52:40;
  cltt 1 2024/07/22 18:52:40;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 00:00:00:aa:00:17;
  set ddns-rev-name = "2.32.16.172.in-addr.arpa.";
  set ddns-txt = "004ed82533c534a476a7048f283dbf123e";
  set ddns-fwd-name = "h12.jbgoncalvess.predes.ufsm.br";
  client-hostname "h12";
}
lease 172.16.32.3 {
  starts 1 2024/07/22 18:52:56;
  ends 1 2024/07/22 19:52:56;
  cltt 1 2024/07/22 18:52:56;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 00:00:00:aa:00:16;
  set ddns-rev-name = "3.32.16.172.in-addr.arpa.";
  set ddns-txt = "00dfd84aa95addalcf031da5f5782aab07";
  set ddns-fwd-name = "h11.jbgoncalvess.predes.ufsm.br";
  client-hostname "h11";
}

```

Figura 20: Arquivo de *leases* do DHCP [Autores 2024].

```

ia-na "\000\252\000\000\001\000\001.1G\370\000\000\000\252\000" {
  cltt 1 2024/07/22 16:40:31;
  iaaddr 2804:1f38:2001:e00::4b1 {
    binding state active;
    preferred-life 2250;
    max-life 3600;
    ends 1 2024/07/22 17:40:31;
  }
}

ia-na "\003\000\252\000\000\001\000\001.1G\370\000\000\000\252\000\003" {
  cltt 1 2024/07/22 16:40:31;
  iaaddr 2804:1f38:2001:300::4c5 {
    binding state active;
    preferred-life 2250;
    max-life 3600;
    ends 1 2024/07/22 17:40:31;
  }
}

ia-na "\002\000\252\000\000\001\000\001.1G\367\000\000\000\252\000\002" {
  cltt 1 2024/07/22 16:40:31;
  iaaddr 2804:1f38:2001:200::45a {
    binding state active;
    preferred-life 2250;
    max-life 3600;
    ends 1 2024/07/22 17:40:31;
  }
}

```

Figura 21: Arquivo de *leases* do DHCPv6 [Autores 2024].

Se o DHCP está escrevendo nas *leases* as atribuições de endereços IP, significa que ele está realizando elas.

acessado externamente) e para o servidor SERVER (que deve ser acessado externamente). A Figura 24 apresenta os resultados.

```
root@cpd:/tmp/pycore.1/cpd.conf# ping 172.16.5.2
PING 172.16.5.2 (172.16.5.2) 56(84) bytes of data.
^C

root@cpd:/tmp/pycore.1/cpd.conf# ping 200.18.74.66
PING 200.18.74.66 (200.18.74.66) 56(84) bytes of data.
64 bytes from 200.18.74.66: icmp_seq=1 ttl=61 time=0.234 ms
64 bytes from 200.18.74.66: icmp_seq=2 ttl=61 time=0.146 ms
64 bytes from 200.18.74.66: icmp_seq=3 ttl=61 time=0.097 ms
^C
--- 200.18.74.66 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.097/0.159/0.234/0.056 ms
root@cpd:/tmp/pycore.1/cpd.conf#
```

Figura 24: Testando conexão da Internet com os servidores internos e externos [Autores 2024].

Em seguida, foi realizado o mesmo teste utilizando IPv6, conforme apresenta a Figura 25.

```
root@cpd:/tmp/pycore.1/cpd.conf# ping 2804:1f38:2001:600::4
PING 2804:1f38:2001:600::4(2804:1f38:2001:600::4) 56 data bytes
^C

root@cpd:/tmp/pycore.1/cpd.conf# ping 2804:1f38:2001:700::4ca
PING 2804:1f38:2001:700::4ca(2804:1f38:2001:700::4ca) 56 data bytes
64 bytes from 2804:1f38:2001:700::4ca: icmp_seq=1 ttl=61 time=0.154 ms
64 bytes from 2804:1f38:2001:700::4ca: icmp_seq=2 ttl=61 time=0.168 ms
64 bytes from 2804:1f38:2001:700::4ca: icmp_seq=3 ttl=61 time=0.179 ms
^C
--- 2804:1f38:2001:700::4ca ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2035ms
rtt min/avg/max/mdev = 0.154/0.167/0.179/0.010 ms
root@cpd:/tmp/pycore.1/cpd.conf#
```

Figura 25: Testando conexão da Internet com os servidores internos e externos (IPv6) [Autores 2024].

5.3. Controle da Rede das Impressoras

A rede das impressoras possui restrições de uso, como mencionado anteriormente. Para demonstrar essas restrições, utilizamos o comando *ping* a partir da sub-rede dos professores e da sub-rede do laboratório de *hardware*. Os resultados mostram que os professores têm acesso às impressoras, enquanto o laboratório de *hardware* não possui essa permissão. A Figura 26 mostra os resultados obtidos.


```

root@h22:/tmp/pycore.1/h22.conf# ping 172.16.8.2
PING 172.16.8.2 (172.16.8.2) 56(84) bytes of data.

root@h14:/tmp/pycore.1/h14.conf# ping 172.16.8.2
PING 172.16.8.2 (172.16.8.2) 56(84) bytes of data.
64 bytes from 172.16.8.2: icmp_seq=1 ttl=62 time=0.263 ms
64 bytes from 172.16.8.2: icmp_seq=2 ttl=62 time=0.100 ms
64 bytes from 172.16.8.2: icmp_seq=3 ttl=62 time=0.109 ms
^C
--- 172.16.8.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.100/0.157/0.263/0.074 ms
root@h14:/tmp/pycore.1/h14.conf# █

```

Figura 26: Testando a conexão com a rede das impressoras [Autores 2024].

Em seguida foi utilizado o *script* para liberar a sub-rede do laboratório de *hardware* para acesso à impressoras, conforme apresenta a Figura 27.

```

root@Impressoras:/home/core/Desktop/controle# sh liberar_impressoras.sh
Digite um endereço IPv4 para liberar o acesso à impressora:
ATENÇÃO: O endereço pode ser um host ou uma sub-rede no formato CIDR.
172.16.11.0/24
/
SUB-REDE 172.16.11.0/24 COM ACESSO LIBERADO AS IMPRESSORAS.
root@Impressoras:/home/core/Desktop/controle#

```

Figura 27: Testando a conexão com a rede das impressoras [Autores 2024].

Após a execução desse comando, conforme mostrado na Figura 28, pode-se observar que o *host* do laboratório conseguiu estabelecer a conexão com as impressoras.

```

root@h22:/tmp/pycore.1/h22.conf# ping 172.16.8.2
PING 172.16.8.2 (172.16.8.2) 56(84) bytes of data.
64 bytes from 172.16.8.2: icmp_seq=1 ttl=62 time=0.108 ms
64 bytes from 172.16.8.2: icmp_seq=2 ttl=62 time=0.128 ms
64 bytes from 172.16.8.2: icmp_seq=3 ttl=62 time=0.110 ms
64 bytes from 172.16.8.2: icmp_seq=4 ttl=62 time=0.105 ms
64 bytes from 172.16.8.2: icmp_seq=5 ttl=62 time=0.396 ms
64 bytes from 172.16.8.2: icmp_seq=6 ttl=62 time=0.107 ms
^C
--- 172.16.8.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5116ms
rtt min/avg/max/mdev = 0.105/0.159/0.396/0.106 ms
root@h22:/tmp/pycore.1/h22.conf# █

```

Figura 28: Testando a conexão com a rede das impressoras após a liberação para o laboratório de *hardware* [Autores 2024].

O mesmo processo funciona para IPv6, já que a rede foi projetada para a implementação de pilha dupla, não só para a entrega de endereços e resolução de nomes como também os demais itens, como *firewall* por exemplo.

5.4. Controle da Rede dos Laboratórios

É necessário bloquear a conexão com a Internet dos laboratórios conforme necessário. A Figura 29 mostra os *hosts* 25 e 26 pingando a rede externa após o bloqueio da conexão com a Internet do laboratório de informática 1. É importante destacar que o *host* 25 pertence ao Lab-info 1, enquanto o *host* 26 pertence ao Lab-info 2.

```
root@h25:/tmp/pycore.1/h25.conf# ping 172.16.0.20
PING 172.16.0.20 (172.16.0.20) 56(84) bytes of data.
[...]
```

```
root@h26:/tmp/pycore.1/h26.conf# ping 172.16.0.20
PING 172.16.0.20 (172.16.0.20) 56(84) bytes of data.
64 bytes from 172.16.0.20: icmp_seq=1 ttl=61 time=0.211 ms
64 bytes from 172.16.0.20: icmp_seq=2 ttl=61 time=0.130 ms
64 bytes from 172.16.0.20: icmp_seq=3 ttl=61 time=0.119 ms
^C
--- 172.16.0.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.119/0.153/0.211/0.041 ms
root@h26:/tmp/pycore.1/h26.conf#
```

Figura 29: Teste de conexão com a Internet: um laboratório bloqueado e outro não
[Autores 2024].

6. Conclusão

O desenvolvimento deste projeto de redes demonstrou a importância de um planejamento meticuloso, a necessidade de uma configuração precisa de servidores e serviços, e a implementação de fortes medidas de segurança. Através de uma abordagem sistemática e detalhada, foi possível criar uma rede eficiente, segura e escalável que atende às necessidades específicas do CTISM. Este projeto não apenas destaca a complexidade envolvida no design e implementação de redes modernas, mas também a importância da contínua adaptação e otimização para atender às demandas crescentes e dinâmicas de conectividade e segurança.

Artefatos disponíveis em: github.com

Referências