

**ADMINISTRAÇÃO DE SISTEMAS**

SPRINT B

TURMA 3NA

GRUPO 74

| 1140858 Carlos Moutinho

| 1171602 Rui Marinho

| 1181882 Rafael Soares

| 1181892 Sara S. Silva

| 1181895 Fábio Silva

# **Introdução**

O presente trabalho foi desenvolvido no âmbito da disciplina de Administração de Sistemas (ASIST) no âmbito do projeto integrador de LAPR5.

Neste relatório será efetuada o estudo das seguintes UC’s:

* Como administrador da infraestrutura quero que o servidor Windows e Linux forneçam endereços IP (na segunda placa de rede) da família 192.168.X.0/24 aos postos clientes, onde X é obtido por 100 + número\_do\_grupo (exemplo, para o grupo 99, X=199); para o efeito devo alterar o endereço dessa placa assignado nas aulas PL.
* Como administrador da infraestrutura quero que os serviços acima referidos funcionem em failover, com um deles a facultar endereços de 192.168.X.50 a 192.168.X.150 e o outro de 192.168.X.151 a 192.168.X.200.
* Como administrador da infraestrutura quero os servidores Windows e Linux estejam disponíveis apenas para pedidos HTTP e HTTPS. Tal não deve impedir o acesso por SSH ou RDP aos administradores (o grupo).
* Como administrador da infraestrutura quero impedir o IP spoofing na minha rede.
* Como administrador da infraestrutura quero que os utilizadores registados no Linux com UID entre 6000 e 6500 só consigam aceder via SSH se esse acesso for a partir de uma máquina listada em /etc/remote-hosts.
* Como administrador da infraestrutura quero que o acesso ao sistema seja inibido aos utilizadores listados em /etc/bad-guys.
* Como administrador da infraestrutura quero que as mensagens pré-login e pós-login bem-sucedido sejam dinâmicas (por exemplo, “[Bom dia] | [Boa tarde] username”, etc.).
* Como administrador da infraestrutura quero que o servidor Linux responda e envie pedidos ICMP para teste de conectividade apenas e só aos computadores dos elementos do grupo.

# **21 - Como administrador da infraestrutura quero que o servidor Windows e Linux forneçam endereços IP (na segunda placa de rede) da família 192.168.X.0/24 aos postos clientes, onde X é obtido por 100 + número\_do\_grupo (exemplo, para o grupo 99, X=199); para o efeito devo alterar o endereço dessa placa assignado nas aulas PL**

CONFIGURAÇÕES:

* servidor **Linux** com ip fixo **192.168.174.1**
* servidor **Windows** com ip fixo **192.168.174.2**.

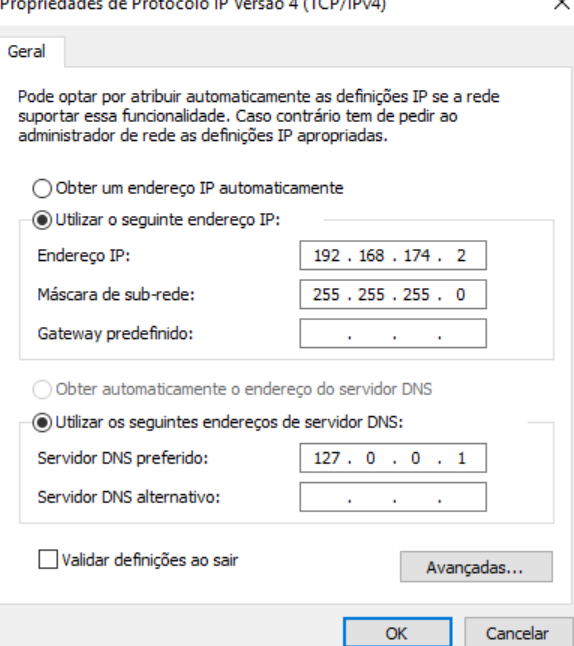
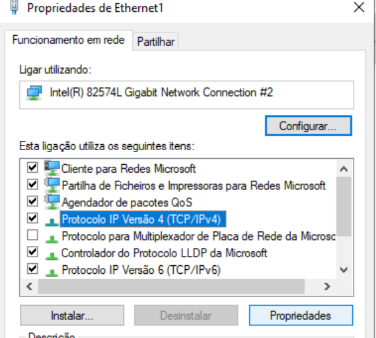
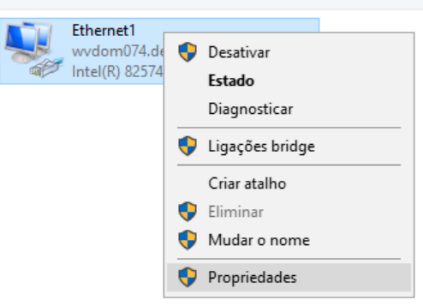
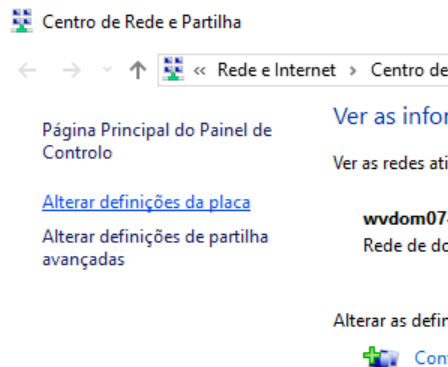
##### **LINUX**:

Text

Description automatically generatedEditamos o ficheiro **/etc/network/interfaces** com o comando **nano /etc/network/interfaces** e alteramos as linhas correspondentes à segunda placa - no nosso caso, a placa **ens33**. Configuramos com o ip **192.168.174.1** e a respetiva máscara **255.255.255.0.** como representado na imagem abaixo.

##### **WINDOWS**:

Para o Windows, fomos às definições de rede, às propriedades da placa pretendida (segunda placa: **Ethernet1**), selecionamos a ligação que pretendemos, neste caso IPv4, e alteramos o ip para o ip pretendido (**192.168.174.2**).



1

2

3

4

# **22 - Como administrador da infraestrutura quero que os serviços acima referidos funcionem em failover, com um deles a facultar endereços de 192.168.X.50 a 192.168.X.150 e o outro de 192.168.X.151 a 192.168.X.200**

Para os sistemas funcionarem em failover, configuramos:

* servidor **Linux** para atribuir ips entre **192.168.174.50 e 192.168.174.150**
* servidor **Windows** entre **192.168.174.151 e 192.168.174.200**.

##### **LINUX**:

Para configurar o DHCP no servidor Linux, foi necessário instalar o serviço **isc-dhcp-server** com o comando **sudo apt install isc-dhcp-server.**

Text

Description automatically generatedUma vez instalado definimos a subnet e o range como solicitado e para tal, foi necessário editar o ficheiro **dhcpd.conf** através do comando **nano /etc/dhcp/dhcpd.conf** e inserir os ips e a máscara pretendida.

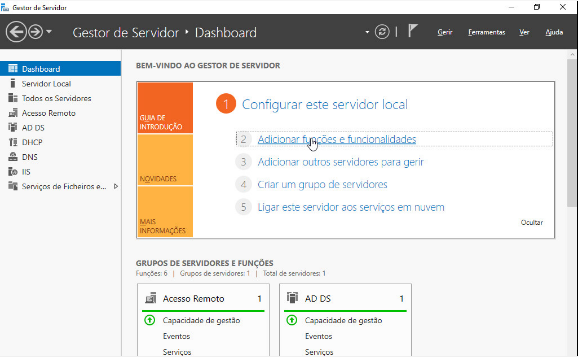
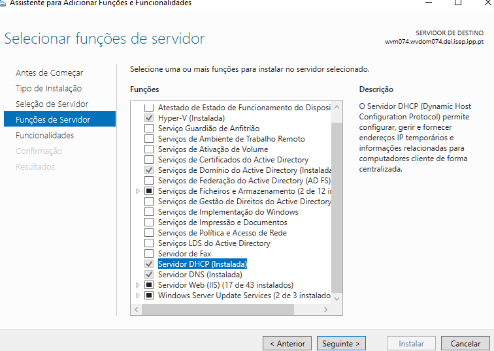
Text

Description automatically generatedUma vez configurado, definimos a segunda pplaca, ens33, como sendo a interface com o DHCP, editando o ficheiro **isc-dhcp-server** com o comando **nano /etc/default/isc-dhcp-server.**

Finalmente, arrancamos com o respetivo serviço com o comando **Systemctl start isc-dhcp-server.**

##### **WINDOWS:**

Em primeiro lugar, instalamos o servidor DHCP através da opção **adicionar funções e funcionalidades** no gestor de servidor.

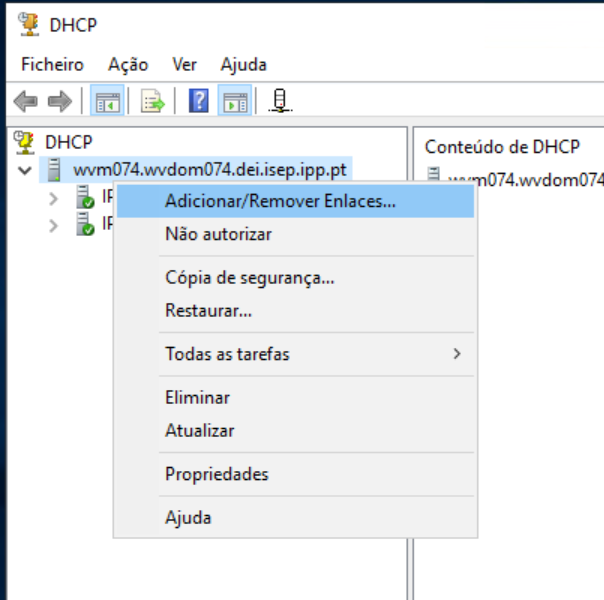
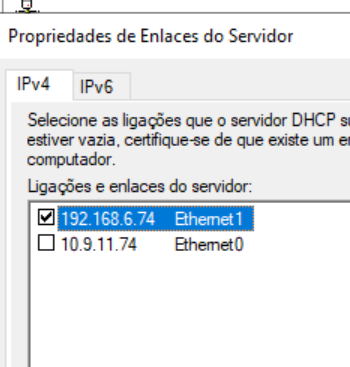
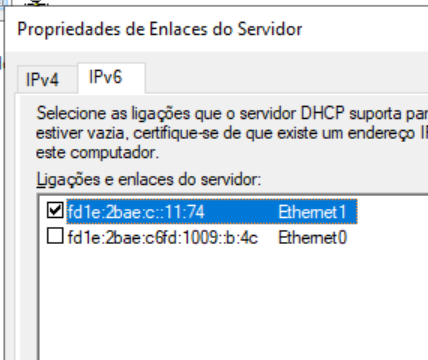


1

2

Depois de instalado, na pesquisa apareceu a aplicação **DHCP**. Selecionando-a, a janela mostrou os servidores presentes no sistema.

Para adicionar o range na placa que pretendemos, tivemos que selecionar a opção **Adicionar/Remover Enlaces** e de seguida selecionar a placa Ethernet1 (segunda placa), tanto para Ipv4 como para Ipv6**.**



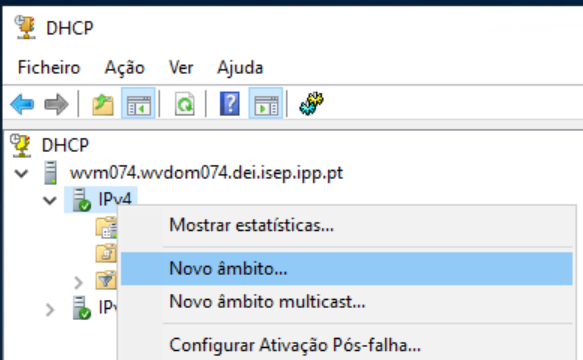
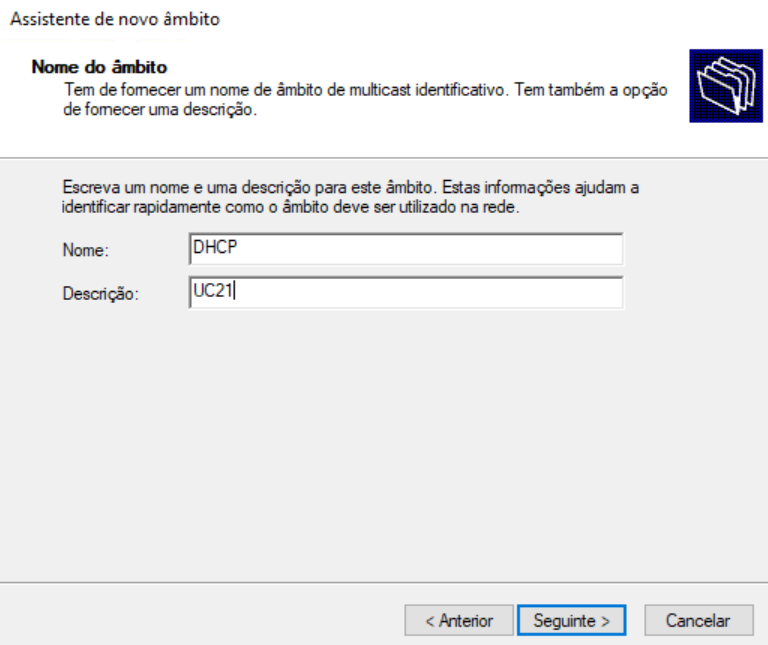
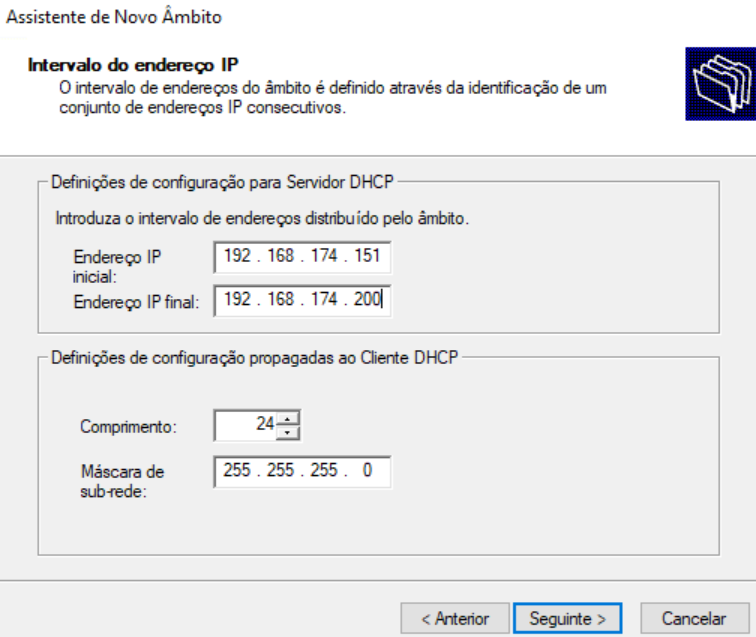
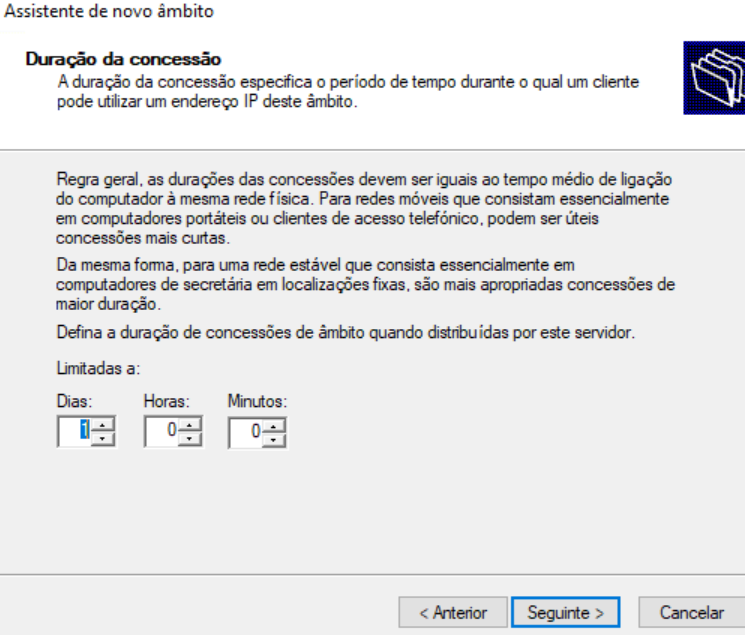
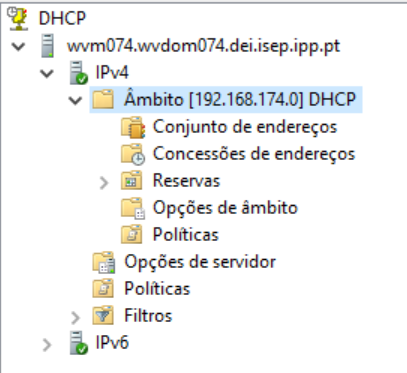
1

2

3

4

Para adicionar/configurar o range de atribuição de ips, tivemos que selecionar a versão de ip (IPv4) e selecionar **Novo Âmbito**. Para finalizar a configuração, seguimos os passos e definimos as configurações necessárias, tais como, **Nome**, **Descrição**, **Range** de ips e respetiva **máscara**, **duração** da concessão e ips a excluir (no nosso caso não foi necessário especificar).



1

2

3

4

5

#### **23 - Como administrador da infraestrutura quero os servidores Windows e Linux estejam disponíveis apenas para pedidos HTTP e HTTPS. Tal não deve impedir o acesso por SSH ou RDP aos administradores (o grupo).**

##### **LINUX**:

A nossa solução passou por bloquear todas as portas e apenas libertar as que pretendíamos. Para isso, as regras de permissão foram inseridas antes da regra de bloqueio, de forma a analisar o tráfego primeiro e só depois bloquear. Para isso foi criado um ficheiro para as iptables com o nome **iptables.sh** criado em **/etc/save** com o comando nano **/etc/save/iptables.sh**.

Graphical user interface

Description automatically generated with medium confidence

Para memorizar permanentemente as configurações das iptables, foi executado o comando **iptables-save > /etc/save/iptables.sh**.

De forma a carregar as iptables quando o servidor é desligado ou reiniciado, foi necessário adicionar o comando **iptables-restore > /etc/save/iptables.sh** ao ficheiro **crontab**.

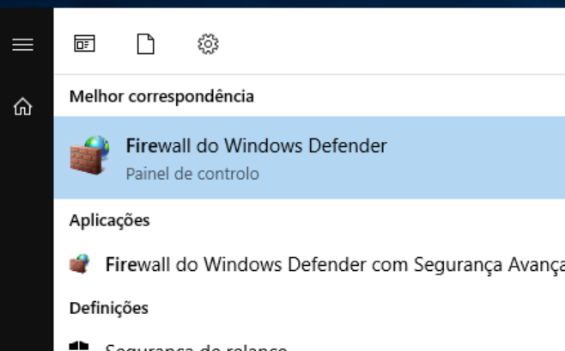
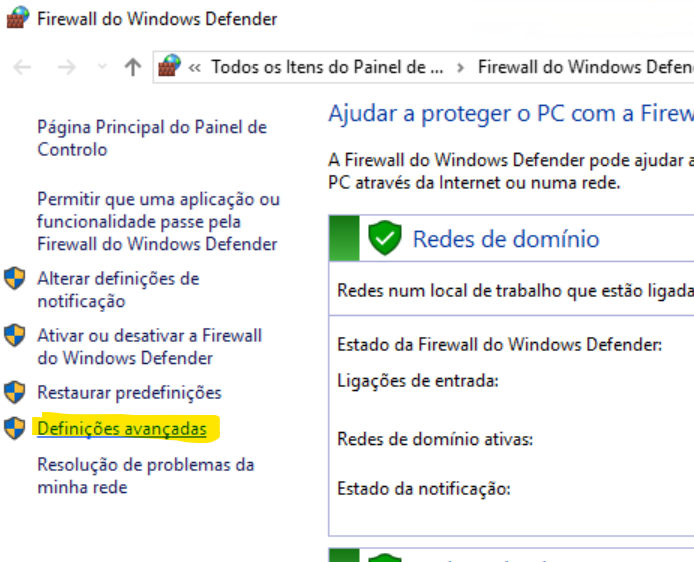
O ficheiro **crontab** no Linux é um daemon que executa tarefas editadas pelo usuário em horários e eventos específicos. Neste caso, não temos horário definido é apenas no arranque da máquina. Alteramos o ficheiro e colocamos o comando.

**Text

Description automatically generated**

##### **WINDOWS:**

No sistema Windows não há ordem de execução de regras na firewall, ou seja, estando a Firewall ativa, é permitido apenas o tráfego indicado nas regras sendo o restante bloqueado por defeito. Para o servidor ficar apenas disponível para pedidos HTTP e HTTPs, foi necessário aceder às definições avançadas da Firewall.

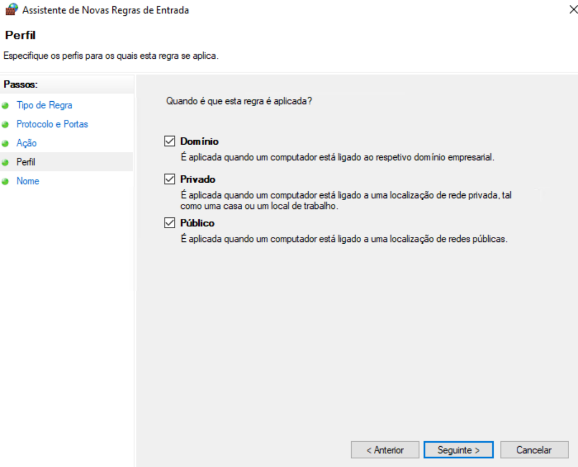
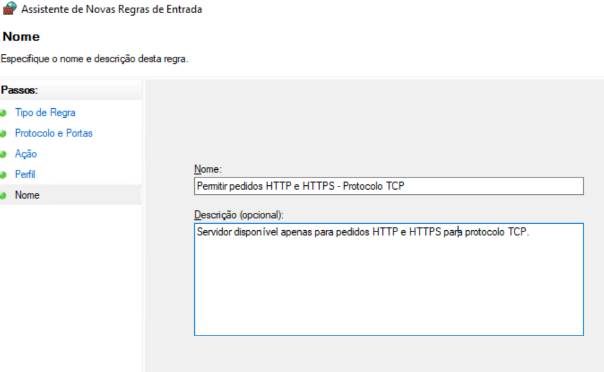
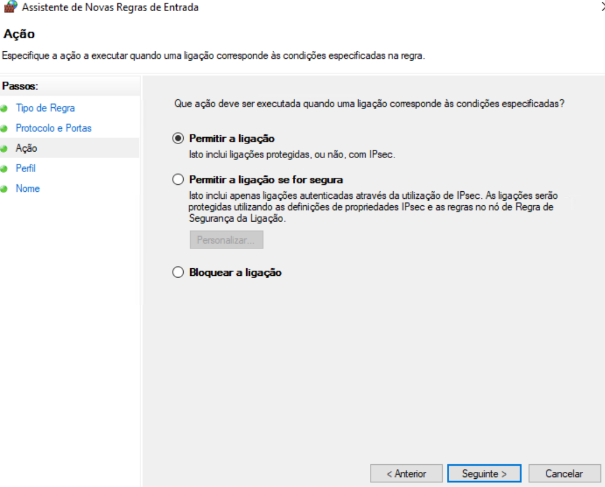
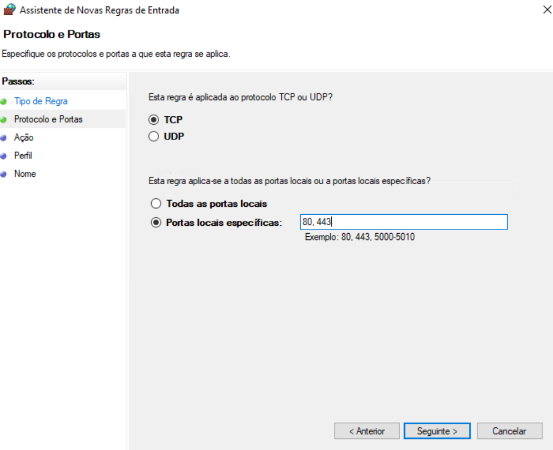
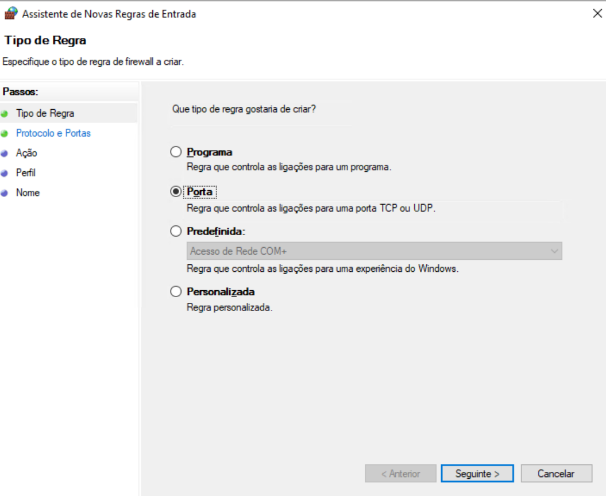
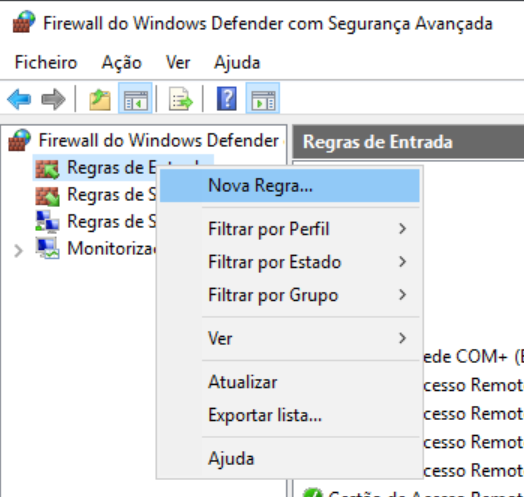


1

2

Dentro da Firewall foi necessário criar as regras de entrada. Criamos duas regras, uma para permitir pedidos HTTP e HTTPs para o protocolo TCP, nas portas correspondentes 80 e 443, e uma segunda com as mesmas configurações para o protocolo UDP.

Exemplo para protocolo TCP:



1

2

3

4

5

6

Por fim, as regras criadas:

Graphical user interface, text, application

Description automatically generated

#### **24 - Como administrador da infraestrutura quero impedir o IP spoofing na minha rede**

##### **LINUX:**

Para evitar o *spoofing* na nossa rede, que pode ser definida pela apropriação de uma identidade numa rede de computadores, bloqueamos os pacotes das redes privadas. Isto que significa que as solicitações enviadas com estes endereços de ips seriam impossíveis, pois indicariam que existe uma rede privada dentro de nossa rede. Também bloqueamos o intervalo de ips na nossa rede.



Text

Description automatically generated

A entrada net.ipv4.conf.all.rp\_filter = 1 ativa a verificação do endereço de origem, que é embutida no próprio kernel do Linux e as duas últimas linhas registam todos esses pacotes falsificados no arquivo de log.

Uma imagem com texto

Descrição gerada automaticamente

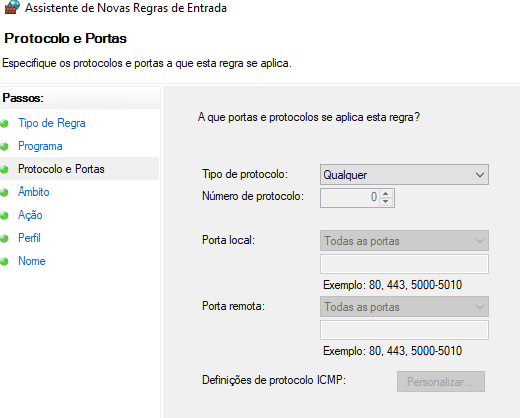
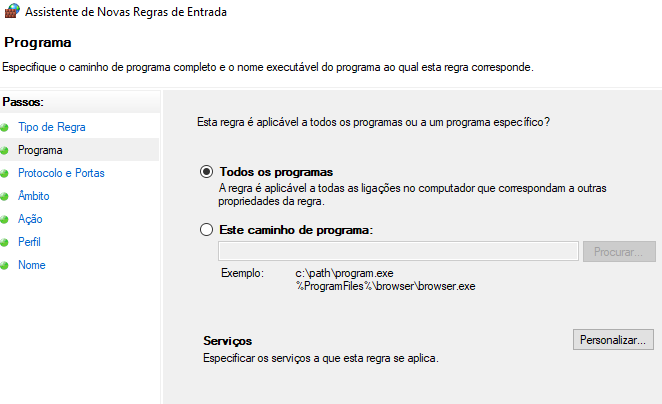
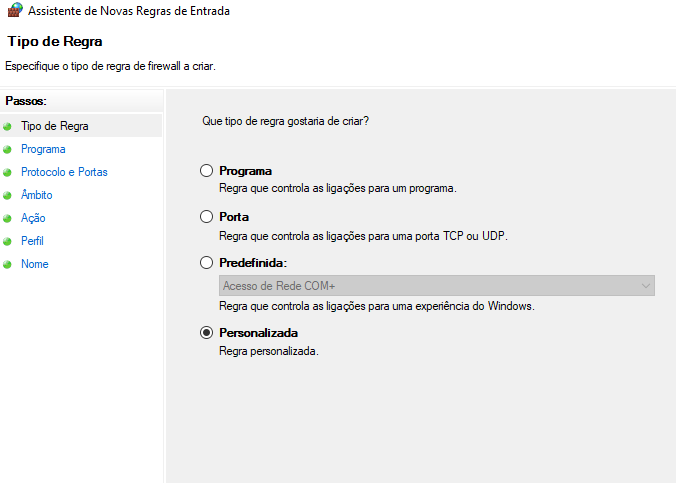
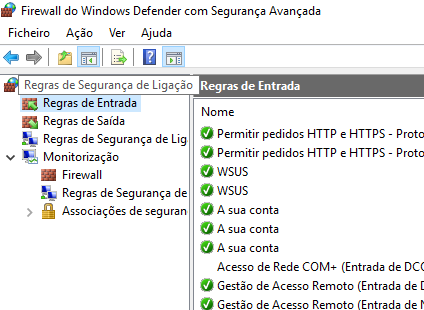
##### **WINDOWS:**

Para impedir o IP spoofing no servidor Windows, foi necessário criar regras de entrada e de saída.

Para as regras de entrada, foi necessário criar uma para aceitar entrada de IPs na nossa gama de IPs **192.168.174.0/24** e uma para bloquear as várias gamas, tais como: **0.0.0.0/8** - Rede atual (válido apenas como endereço de origem), **127.0.0.0/8** – ip privado, **10.0.0.0/8** – ip privado,

* **172.16.0.0/12** – ip privado, **192.168.0.0/16**, **224.0.0.0/3**, **192.168.174.2/24** (IP do próprio servidor).

Exemplo de criação de regra de entrada para impedir acesso:

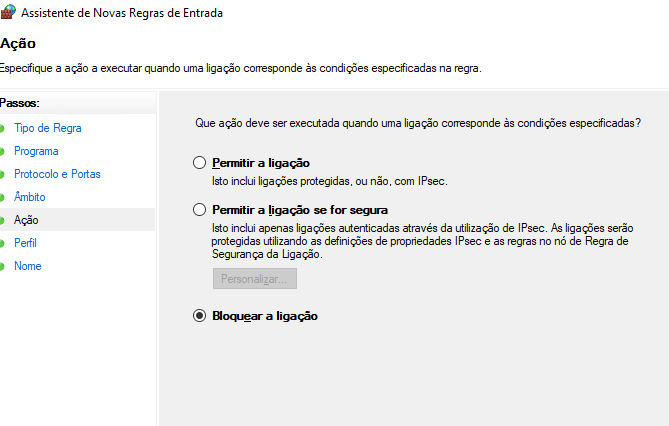
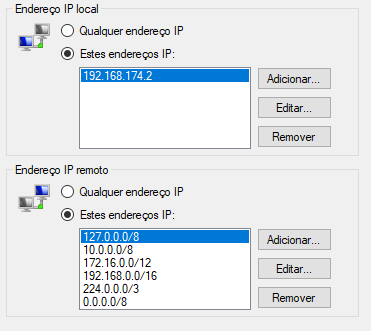


1

2

3

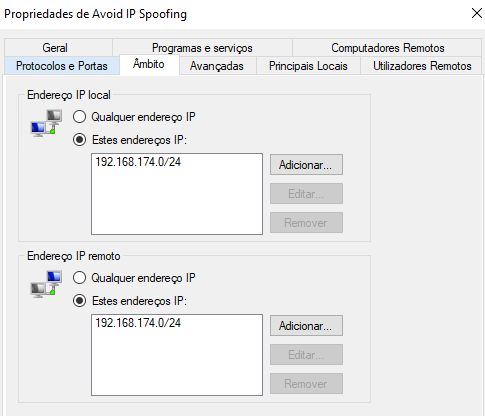
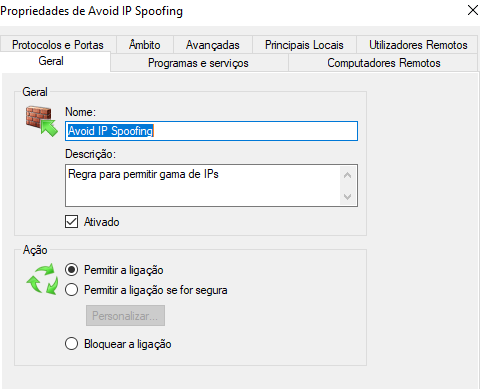
4



5

6

Regra de entrada para permitir acesso a gama de IPs **192.168.174.0/24**:



1

2

Resultado das regras de entrada:

Graphical user interface, application

Description automatically generated

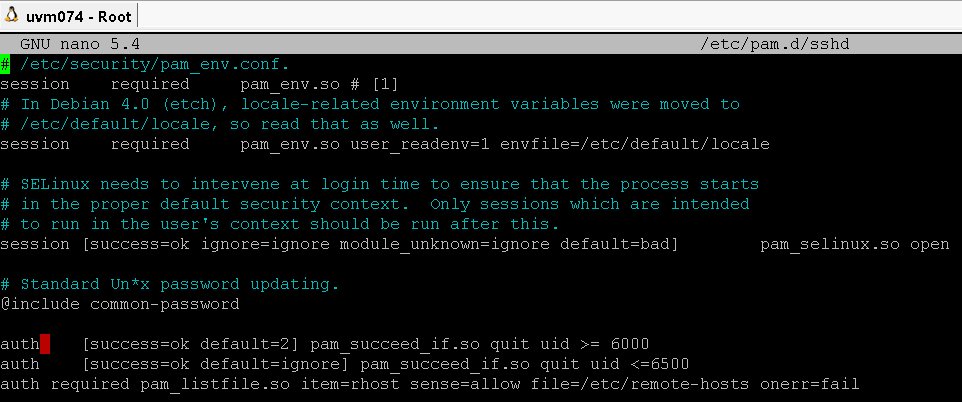
Para as regras de saída, foram criadas as regras com as mesmas configurações:

Graphical user interface, application

Description automatically generated

#### **25 - Como administrador da infraestrutura quero que os utilizadores registados no Linux com UID entre 6000 e 6500 só consigam aceder via SSH se esse acesso for a partir de uma máquina listada em /etc/remote-hosts**

O módulo PAM, pam\_limits, define os limites dos recursos do sistema que podem ser obtidos numa sessão do utilizador ou seja, PAM é o modulo que nega sempre os acessos, salvo indicação do contrário. Tendo isto em mente, abrimos o ficheiro **/etc/pam.d/sshd** e inserimos no final três linhas de código (ver imagem abaixo). A primeira linha irá testar se o UID do user em questão é igual ou superior a 6000, caso não seja, por defeito, irá saltar as duas linhas, descartando assim a necessidade de verificar se o IP está na lista de IPs autorizados. Caso seja superior a 6000, irá verificar na segunda linha se o UID é igual ou inferior a 6500 o que, caso não seja, irá ignorar e tentar a terceira linha. A terceira linha irá verificar o ficheiro /**etc/remote-hosts** para ver se o IP da sessão que está a ser testada pertente à lista de IPs aceites.



Após definir este ficheiro, criamos então o ficheiro **/etc/remote-hosts** e inserimos um IP de uma das nossas máquinas para testarmos.

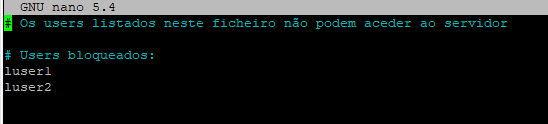


Uma imagem com texto, relógio

Descrição gerada automaticamente

#### **26 - Como administrador da infraestrutura quero que o acesso ao sistema seja inibido aos utilizadores listados em /etc/bad-guys**

Iniciamos o processo criando o ficheiro bad-guys com os users que pretendíamos inibir o acesso.



1

2

Para configurar e inibir o respetivo acesso, foi necessário editar o ficheiro **common-auth** com o comando **nano /etc/pam.d/common-auth** e adicionar a seguinte linha:

**auth required pam\_listfile.so item=user sense=deny file=/etc/bad-guys onerr=succeed**

Text

Description automatically generated

O **pam\_listfile** é um módulo **PAM** que dá a possibilidade de negar ou permitir serviços baseados num ficheiro arbitrário. Nessa mesma configuração, o argumento **item** **= user** indica a **pam\_listfile** que este deve encontrar nomes de utilizadores em **//etc/bad-guys**.

O argumento **sense = deny** indica a **pam\_listfile** que o ficheiro **bad-guys**,em **file=//etc/bad-guys,** é uma lista de negação, ou seja, qualquer utilizador nesse ficheiro faz com que o **pam\_listfile** falhe, e por consequente, fazendo com que a autenticação falhe.

O argumento **onerr = succeed** indica que a condição de sucesso é erro.

Por último, foi necessário reiniciar os serviços através dos comandos **sudo service ssh restart** e **sudo service sshd restart.**



Como podemos ver, ao utilizador luser2 que estava presente na lista, não é permitido iniciar sessão.

A screenshot of a computer

Description automatically generated with medium confidence

#### **27 - Como administrador da infraestrutura quero que as mensagens pré-login e pós-login bem-sucedido sejam dinâmicas (por exemplo, “[Bom dia] | [Boa tarde] username”, etc.)**

##### **LINUX:**

Para resolvermos esta UC, dividimos a questão em duas partes.

A primeira parte consistiu em definir a mensagem de pré login (mensagem apresentada aos utilizadores que estão prestes a entrar e a efetuar o login). Para isso criamos um ficheiro no diretório **/etc/ssh/** com o nome “ssh\_banner\_pre\_login”.

Uma imagem com texto

Descrição gerada automaticamente

Neste ficheiro colocamos a mensagem que gostaríamos de ver ao inicializarmos a máquina.

Uma imagem com texto

Descrição gerada automaticamente

Depois da criação e edição do ficheiro, vamos ao ficheiro **sshd\_config,** localizado no diretório /etc/ssh/, e descomentamos a linha que indica o banner (posteriormente estava a banner none) e inserimos o banner definido anteriormente.



Uma imagem com texto

Descrição gerada automaticamente

Com disto, damos como concluído a criação do banner de pré-login. Para testarmos, reiniciamos a máquina com o comando **init 6** e comprovamos o resultado.



Uma imagem com texto

Descrição gerada automaticamente

Para realizar a segunda etapa desta UC, a definição customizada do pós-login, fomos ao ficheiro **/etc/profile** e inserimos um script que irá verificar as horas atuais e consoante essas horas irá dar as boas-vindas dizendo ‘bom dia’, ‘boa tarde’ ou ‘boa noite’ ao utilizador logado.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Por fim, ao fazermos o login podemos comprovar que já temos a mensagem de pós-login a funcionar como solicitado.

Uma imagem com texto

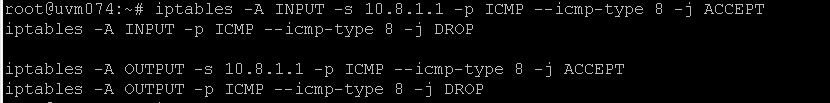
Descrição gerada automaticamente

#### **28 - Como administrador da infraestrutura quero que o servidor Linux responda e envie pedidos ICMP para teste de conectividade apenas e só aos computadores dos elementos do grupo**

Para permitirmos envios de ping de determinadas máquinas podemos usar o comando **iptables**.

Este comando funciona de forma ordenada, ou seja, o primeiro comando tem prioridade sobre os demais. Com isto em mente, colocamos a aceitação dos IPs das nossas máquinas nas primeiras linhas, tanto para input como para output. Nas linhas imediatamente a seguir colocamos o *drop* de qualquer IP tanto para input como para output.

Ao fazermos um ping à nossa máquina em Linux, através de uma das nossas máquinas dos elementos do grupo, iremos conseguir obter uma resposta do ping uma vez que os nossos IPs estão na primeira linha da iptables. Caso não seja o IP autorizado, automaticamente passaria para a segunda linha.



Text

Description automatically generated