

Uma imagem com alimentação

Descrição gerada automaticamente

**Relatório do Sprint 1**

**Turma 2DH \_ Grupo 43**

1190929 \_ Patrícia Barbosa

1190947 \_ Pedro Fraga

1190956 \_ Pedro Garcia

1190963 \_ Pedro Preto

**Professor:**

André Moreira, ASC

**Unidade Curricular:**

Arquitetura de Sistemas

**Data: 04/12/2021**

Índice

[INTRODUÇÃO 3](#_Toc89563440)

[USER STORY 1 E 2 4](#_Toc89563441)

[LINUX 4](#_Toc89563442)

[WINDOWS **Erro! Marcador não definido.**](#_Toc89563443)

[USER STORY 3 6](#_Toc89563444)

[LINUX 6](#_Toc89563445)

[WINDOWS 7](#_Toc89563446)

[USER STORY 4 8](#_Toc89563447)

[LINUX 8](#_Toc89563448)

[WINDOWS 9](#_Toc89563449)

[USER STORY 5 10](#_Toc89563450)

[LINUX 10](#_Toc89563451)

[USER STORY 6 11](#_Toc89563452)

[LINUX 11](#_Toc89563453)

[USER STORY 7 12](#_Toc89563454)

[LINUX 12](#_Toc89563455)

[USER STORY 8 14](#_Toc89563456)

[LINUX 14](#_Toc89563457)

# INTRODUÇÃO

O presente relatório funcionará como apoio e explicação de execução das User Stories pedidas para o SPRINT 1 de ASIST. Em cada User Story, está dividido pelas tarefas realizadas em Linux e em Windows.

Nas user stories 1 e 2, não conseguimos resolver no servidor Windows, devido a um erro ao tentar configurar o dhcp.

# USER STORY 1 E 2

Uma imagem com texto, pessoa, captura de ecrã, documento

Descrição gerada automaticamente

Figura 1: User stories 1 e 2

Para a realização de ambas as stories, fizemos uma configuração do DHCP, tanto em Linux, como em Windows.

## LINUX

Em Linux, recorremos ao isc-dhcp-server, que já tinha sido previamente instalado, aquando da realização da ficha PL 5.

Começámos por modificar os parâmetros do DHCP server no ficheiro “dhcpd.conf”, adicionando o IP da subnet, a range de endereços, a máscara e o Broadcast address no final do ficheiro:

Uma imagem com texto

Descrição gerada automaticamente

Figura 2: Alteração efetuado em /etc/dhcp/dhcp.conf

De seguida, definimos a NIC a ser usada pelo isc-dhcp-server. Para tal especificamos que a NIC a ser usada é a “ens33” (placa de rede secundária) em “/etc/default/isc-dhcp-server”.

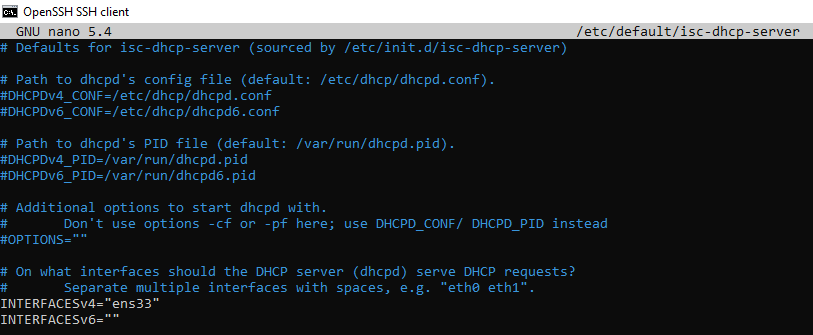


Figura 3: Alteraçao efetuada em /etc/default/isc-dhcp-server

Por fim, reiniciámos o serviço de DHCP e verificamos o status, para confirmar que está ativo, através dos comandos:

* **systemctl restart isc-dhcp-server**
* **systemctl status isc-dhcp-server**

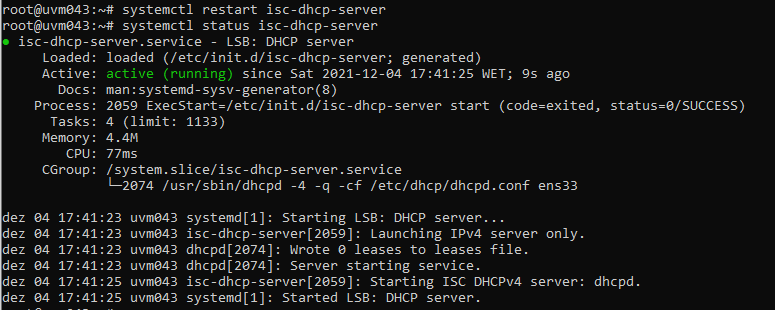


Figura 4: Status do isc-dhcp-server

# USER STORY 3

Uma imagem com texto

Descrição gerada automaticamente

Figura 5: User story 3

## LINUX

Para esta user story, recorremos ao comando iptables para gerir o tráfego de pacotes, através da criação de regras. Contudo, estas regras não persistem após reinício do servidor se apenas utilizarmos o comando iptables. A solução encontrada pelo nosso grupo passou pela criação de um script, que contém a especificação das regras de entrada através sequencialmente. No final do script, rejeita todos os pacotes que não fizeram correspondência com nenhuma das regras, e guarda as configurações. Esse script é chamado através de um “service descriptor” (unit file), criado em “/etc/systemd/system”, que executa o script sempre que o sistema é iniciado (reboot).

Uma imagem com texto, captura de ecrã, monitor, ecrã

Descrição gerada automaticamente

Figura 6: Script criado (/scripts/iptables\_commands.sh)

Uma imagem com texto, captura de ecrã, monitor, ecrã

Descrição gerada automaticamente

Figura 7: unit file criado em /etc/systemd/system para realizar o script a cada reboot

Por fim, alteram-se as permissões ao unit file criado (activate\_script\_on\_boot.service) e ativa-se, através dos seguintes comandos:

* chmod 644 /etc/systemd/system/activate\_script\_on\_boot.service
* systemctl enable activate\_script\_on\_boot.service

## WINDOWS

No Windows, a realização da user story passa pela configuração da firewall do Windows defender, onde posteriormente iremos definir as regras de entrada/saída, tal como em Linux.

Para tal, abrimos a “Firewall do Windows Defender com Segurança Avançada”, escolhemos as opções “**Regras de Entrada/Nova Regra”**. De seguida, escolhemos a opção “**/Protocolo e Portas/Porta/TCP/Portas locais específicas”** e especificámos a porta (80 para http, 443 para https ou 3389 para acesso via rdp). Por fim especificámos o nome da regra e adicionámos.

Uma imagem com texto

Descrição gerada automaticamente

Figura 8: Regras de entrada adicionadas

# USER STORY 4



Figura 9: User story 4

O IP spoofing é uma técnica maliciosa que ocorre quando o endereço de IP origem de pacotes é modificado para esconder a identidade de quem envia o pacote, fazendo-se passar por outro computador ou sistema.

## LINUX

Em Linux, de forma semelhante à user story 3, utilizamos um comando iptables para rejeitar todo o tráfego exterior através da placa de rede primária (ens32), cujo IP de origem provém da segunda placa de rede (ens33 – 192.168.143.0/24). Esse comando foi adicionado ao script já criado também “iptables\_commands.sh”.

Uma imagem com texto, captura de ecrã, monitor, ecrã

Descrição gerada automaticamente

Figura 10: Script /scripts/iptables\_commands.sh com linha de prevenção de IP spoofing adicionada

Para verificar as regras adicionadas, tanto nesta user story, como na user story 3, podemos utilizar o comando iptables, com as opções -L.

Uma imagem com texto, placa

Descrição gerada automaticamente

Figura 11: Verificação das regras adicionadas

## WINDOWS

De forma semelhante à user story 3 e seguindo a mesma lógica utilizada em linux, adicionamos uma regra de entrada, neste caso que bloqueia o acesso. Colocámos o endereço remoto “192.168.143.0/24” (placa de rede secundária), e o protocolo e porta local não são especificados.

Uma imagem com texto

Descrição gerada automaticamente

Figura 12: Regra de entrada adicionada para prevenção do ip spoofing

# USER STORY 5

Uma imagem com texto

Descrição gerada automaticamente

Figura 13: User story 5

## LINUX

De modo a restringir o acesso aos users com id entre 6000 e 6500 via SSH, apenas se o acesso for a partir de uma máquina listada em “/etc/remote-hosts”, começámos por criar o ficheiro “/etc/remote-hosts”.



Figura 14: Ficheiro /etc/remote-hosts com os hosts dos nossos computadores

De seguida, adicionámos no topo do ficheiro “/etc/pam.d/sshd” as configurações para restringir o acessoo aos utilizadores com id entre 6000 e 6500.

Uma imagem com texto, monitor, interior, ecrã

Descrição gerada automaticamente

Figura 15: Configurações no ficheiro /etc/pam.d/sshd

Para testar, colocámos os nossos hosts (ver Figura 8) em “etc/remote-hosts” e tentamos conectar com o user “luser2”, que foi criado na resolução da ficha PL2 e cujo id é 6001. Tivemos sucesso no login por SSH quando o nosso host estava listado no ficheiro criado, mas quando retirámos os hosts, o acesso passa a ser negado. Para um user com id fora do intervalo restrito, o acesso é garantido sem qualquer restrição.

# USER STORY 6

Uma imagem com texto

Descrição gerada automaticamente

Figura 16: User story 6

## LINUX

Na resolução desta user story, começamos por criar o ficheiro “/etc/bad-guys”, e adicionamos o user “luser1”, para mais tarde testar.



Figura 17: Ficheiro /etc/bad-guys criado

De seguida, e à semelhança da user story 5, alterámos o ficheiro “etc/pam.d/sshd”, escrevendo no topo do ficheiro, uma configuração que inibe o acesso aos utilizadoes listados em “/etc/bad-guys”.

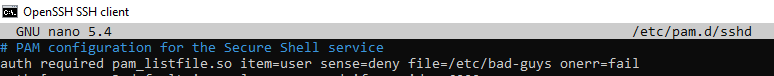


Figura 18: Configuração efetuado no ficheiro /etc/pam.d/sshd

Por fim, tentamos dar login com o user “luser1”, mas o acesso foi negado, devido à presença do user no ficheiro criado.

# USER STORY 7

Uma imagem com texto

Descrição gerada automaticamente

Figura 19: User story 7

## LINUX

Para a mensagem de pré-login, descomentámos a linha referente ao banner, no ficheiro “/etc/ssh/sshd.conf” e adicionámos o caminho para o banner criado, no caso em “/etc/ssh/banner”.



Figura 20: Caminho para o banner em /etc/ssh/sshd.conf

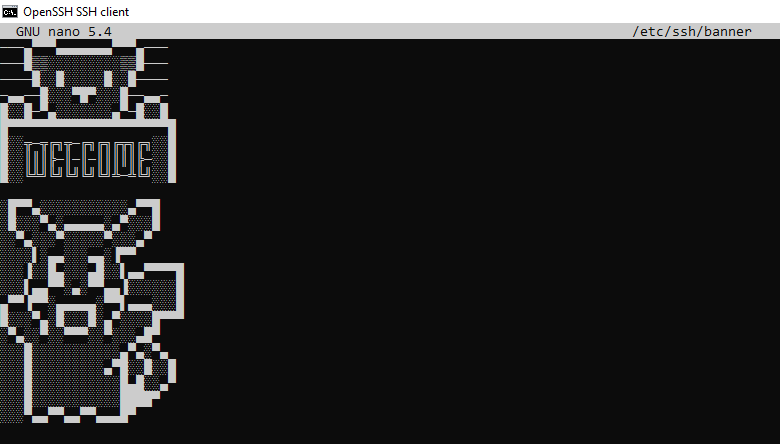


Figura 21: Banner criado

Para a mensagem de pós-login, alterámos o ficheiro “/etc/profile”, criando um script que imprime uma mensagem alterada consoante a altura do dia (manhã, tarde, noite) e o user que acedeu:

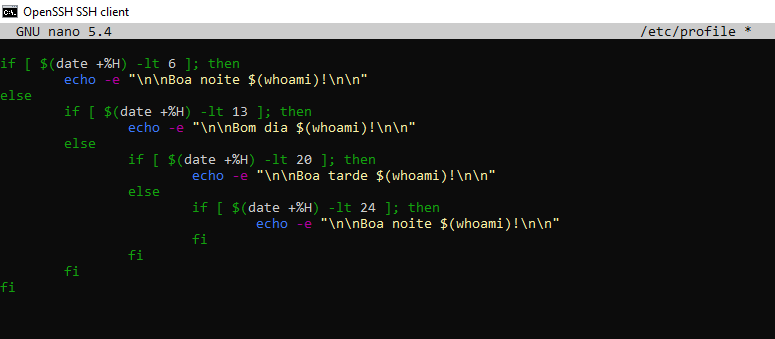


Figura 22: Alteração em /etc/profile



Figura 23: Mensagens pré e pós-login

# USER STORY 8

Uma imagem com texto

Descrição gerada automaticamente

Figura 24: User story 8

## LINUX

Para realização desta user story, recorremos ao script criado na user story 3, adicionando regras através do comando iptables que permitissem respostas e pedidos ICMP por parte do servidor Linux aos computadores dos elementos do grupo. De seguida, demos reboot de modo que o script fosse executado.

Uma imagem com texto, captura de ecrã, monitor, ecrã

Descrição gerada automaticamente

Figura 25: Comandos inseridos no topo de /scripts/iptables\_commands.sh

Contudo o endereço de IP de cada máquina não é sempre o mesmo, pelo que temos noção que esta configuração não seria sempre válida. Tentámos criar um script que adicionaria a regra, utilizando o valor atual do endereço de ip de cada máquina para as opções “-s” e “-d”, mas sem sucesso.

# WEbgrafia

* <https://ixnfo.com/en/iptables-quick-setup-script.html>
* <https://www.baeldung.com/linux/run-script-on-startup>
* <https://linux.die.net/man/8/pam_succeed_if>
* <https://www.howtoforge.com/tutorial/install-and-configure-isc-dhcp-server-in-debian-9/>
* <https://www.veritas.com/support/en_US/article.100027250>