

Relatório de ASIST

Sprint 1

Turma 3DI - Grupo 50

1191296 - Gabriel Gonçalves

1191369 - Tiago Leite

1201305 - Tiago Afonso

1211304 - Francisco Bogalho

Data: 29/10/2023

Índice

Índice de quadros, figuras, abreviaturas	3
Distribuição de tarefas	5
User Story 1.....	6
User Story 2.....	10
User Story 3.....	13
User Story 4.....	15
User Story 5.....	19
User Story 6.....	20
User Story 7.....	22
User Story 8.....	23

Índice de quadros, figuras, abreviaturas

Figura 1 - Texto inserido no ficheiro /etc/issue	6
Figura 2 - Informação apresentada antes de realizar a autenticação	7
Figura 3 - Texto inserido no ficheiro /etc/issue.net.....	8
Figura 4 - Ficheiro /etc/ssh/sshd_config com configuração do Banner	9
Figura 5 - Mensagem antes de autenticação através de conexão SSH.....	9
Figura 6 - Mensagem de login de origem	10
Figura 7 - Desativação da MOTD, no ficheiro /etc/pam.d/login.....	10
Figura 8 - Desativação do último login, no ficheiro /etc/pam.d/login.....	10
Figura 9 - Desativação da MOTD e último login para SSH, no ficheiro /etc/ssh/sshd_config	11
Figura 10 - Bash script que imprime a mensagem de login	11
Figura 11 - Exemplo da nova mensagem de login	12
Figura 12 - Habilitação das quotas.....	13
Figura 13 - Script para os utilizadores não excederem os 300 ficheiros na sua área de trabalho	14
Figura 14 - Visualização das quotas de todos os utilizadores	14
Figura 15 - Local de criação da pasta partilhada.....	15
Figura 16 - Onde aceder às definições avançadas de partilha.....	15
Figura 17 - Onde se cria uma nova partilha	16
Figura 18 - Onde se gera as permissões da nova partilha	16
Figura 19 - Onde cria-se um novo template de quota	17
Figura 20 - Onde e como se configura um novo template de quota	17
Figura 21 - O email enviado quando o uso da pasta partilhada atinge determinado limite	18
Figura 22 - Tentativa de ultrapassar a quota	18
Figura 23 - Novos campos no ficheiro /etc/pam.d/sshd para impor as restrições.....	19
Figura 24 - Endereço IP adicionado no ficheiro /etc/remote-hosts	20
Figura 25 - Comando adicionado no ficheiro /etc/pam.d/sshd.....	20
Figura 26 - Teste de uma conexão SSH bem-sucedida	21
Figura 27 - Teste de uma conexão SSH malsucedida	21
Figura 28 - Os conteúdos do ficheiro bad-guys	22
Figura 29 - O comando para bloquear os users identificados no ficheiro bad-guys	22
Figura 30 - Comando para o reinício do sshd	22
Figura 31 - Prova da funcionalidade da US	22

Figura 32 - Configuração predefinida para a biblioteca pam_cracklib	24
Figura 33 - Configuração de requisitos mínimos para as palavra-passe com a biblioteca pam_cracklib...	24
Figura 34 - Configuração completa do pam_cracklib e pam_unix para definir a complexidade das palavra-chave	25
Figura 35 - Configuração da frequência de expiração das palavra-passe.....	26
Figura 36 - Tentativa de alteração de palavra-passe que não cumpre os requisito mínimos	26
Figura 37 - Alteração de palavra-passe para uma que cumpre os requisitos mínimos	27

Distribuição de tarefas

As *User Stories* foram distribuídas pelos membros do grupo da seguinte forma:

User Story	Membro
1	Tiago Leite (1191369)
2	Tiago Afonso (1201305)
3	Gabriel Gonçalves (1191296)
4	Francisco Bogalho (1211304)
5	Tiago Afonso (1201305)
6	Gabriel Gonçalves (1191296)
7	Francisco Bogalho (1211304)
8	Tiago Leite (1191369)

User Story 1

A autenticação ao nosso sistema Linux pode ser feita de duas formas: através do *vcenter*, que representa a ligação local, e por ligação SSH. O objetivo desta *user story* é personalizar a informação apresentada no terminal antes da autenticação do utilizador, tanto quando tenta uma conexão local, como uma ligação SSH. É requerido que esta informação contenha obrigatoriamente a data e o número de utilizadores ativos no sistema nesse mesmo momento.

Acesso através do *vcenter* (conexão local)

Para alterar a informação que aparece antes de se realizar a autenticação de um utilizador local através do *vcenter*, editamos a informação presente no ficheiro `/etc/issue`. Para fazer essa edição, utilizou-se o comando `'nano /etc/issue'`.

Ao editar este ficheiro, colocamos a informação que é requerida. Na imagem seguinte, é possível visualizar o texto escrito no ficheiro indicado.



```
GNU nano 5.4 /etc/issue
-----
--0000000000--0000000000--0000000000--000--000--0000000000--0000000000--0000000000--
--0000000000--0000000000--0000000000--000--000--0000000000--0000000000--0000000000--
--000--000--000--000--000--000--000--000--000--000--000--000--000--000--000--000--
--000--000000--000--000--000--000--000--000--000--000--0000000000--000--000--
--000--000000--000000000--000--000--000--000--0000000000--0000000000--000--000--
--000--000--0000000000--000--000--000--000--0000000000--0000000000--000--000--000--
--0000000000--000--000--0000000000--0000000000--000--0000000000--0000000000--
--0000000000--000--000--0000000000--0000000000--000--0000000000--0000000000--
-----
Welcome comrade, this is our server \n!
There are \U active in our system in this moment.
Date: \d.
_
```

Figura 1 - Texto inserido no ficheiro `/etc/issue`

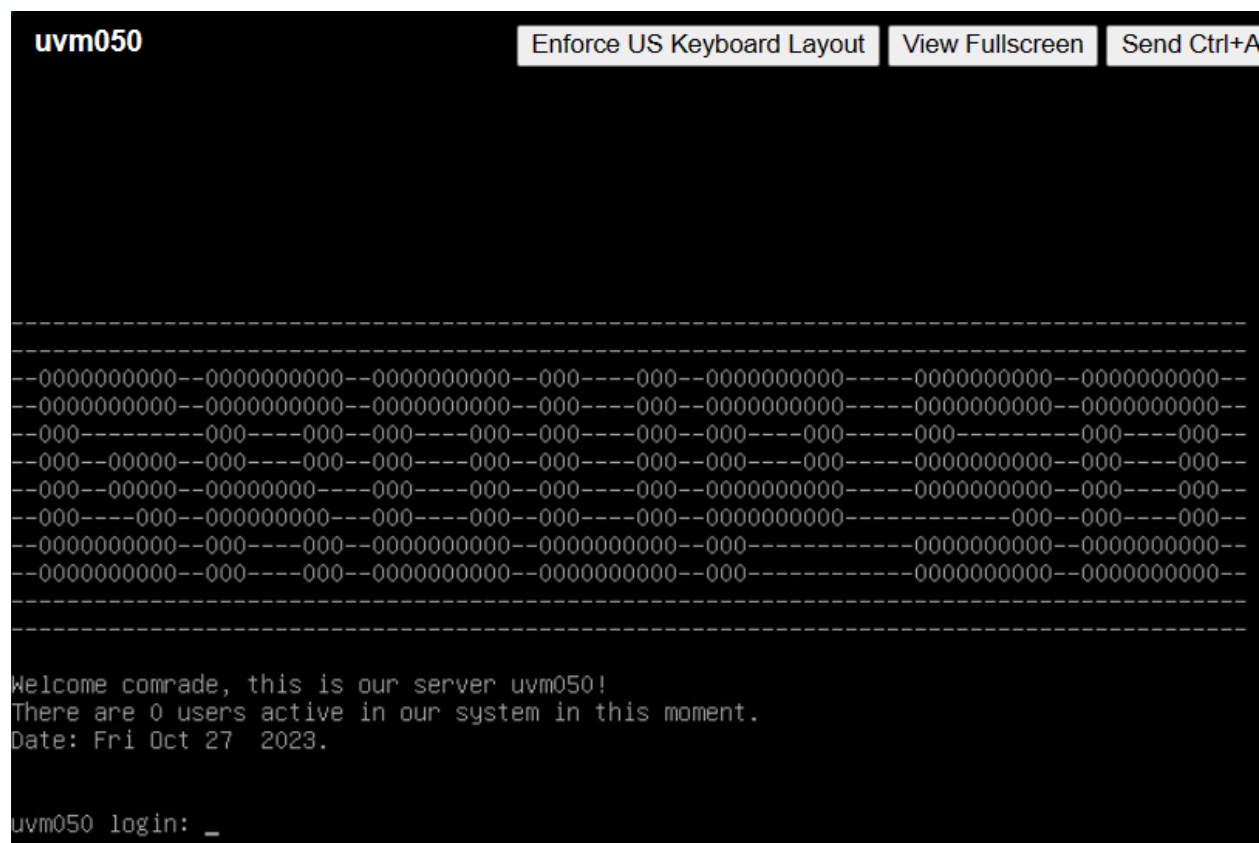
Através da visualização desta imagem, podemos observar que existem algumas letras com um *escape* “\” atrás de uma dada letra, tal como, \n, \u e \d. Estas sequências têm um significado especial neste contexto, pois são interpretados de uma forma diferente pelo sistema. Contudo, vale a pena ressaltar que este é apenas um ficheiro de texto normal, no entanto, aceita alguns *escapes* que representam algo. O significado das sequências utilizadas é o seguinte:

“\n” – Representa o nome do *host*, ou seja, o nome do nó da máquina;

“\U” – Representa o número de utilizadores atualmente ativos no sistema e escreve a *string* “user” ou “users” dependendo do número de utilizador ativos nesse momento. Se utilizarmos a variante “\u” apenas escreve o número sem a *string*;

“\d” – Representa a data atual.

Agora quando tentamos fazer a autenticação local através do *vcenter*, podemos verificar que a nova informação personalizada aparece da forma desejada.



```
uvm050 Enforce US Keyboard Layout View Fullscreen Send Ctrl+A

-----
--0000000000--0000000000--0000000000--000---000--0000000000-----0000000000--0000000000--
--0000000000--0000000000--0000000000--000---000--0000000000-----0000000000--0000000000--
--000-----000---000---000---000---000---000---000---000---000---000-----000-----000---000--
--000--00000--000---000--000---000--000---000--000---000---0000000000--000---000---
--000--00000--000000000--000---000--000---000--0000000000-----0000000000--000---000---
--000---000--000000000--000---000--000---000--0000000000-----0000000000--000---000---
--0000000000--000---000--0000000000--0000000000--000-----0000000000--0000000000--
--0000000000--000---000--0000000000--0000000000--000-----0000000000--0000000000--
-----
Welcome comrade, this is our server uvm050!
There are 0 users active in our system in this moment.
Date: Fri Oct 27 2023.

uvm050 login: _
```

Figura 2 - Informação apresentada antes de realizar a autenticação

Nota: Este ficheiro tem permissões 644, ou seja, apenas o dono, neste caso o *root*, tem permissões para alterar o conteúdo, enquanto o grupo e os outros utilizadores apenas têm permissões de leitura, para que não possam alterar o conteúdo do ficheiro.

Acesso através de uma ligação SSH

Agora ainda falta alterar a informação que aparece antes de o utilizador se autenticar por conexão SSH. Podemos ter duas abordagens aqui, em que uma será criar um ficheiro em qualquer lugar com o texto desejado, ou então utilizamos o ficheiro que se encontra no caminho */etc/issue.net*. Neste caso, vamos utilizar o último ficheiro mencionado, pois é aquele que é utilizado normalmente para este fim. Então através do comando *'nano /etc/issue.net'* começamos a edição do ficheiro.

```
GNU nano 5.4 /etc/issue.net
-----
--0000000000--0000000000--0000000000--000--000--0000000000--0000000000--0000000000--
--0000000000--0000000000--0000000000--000--000--0000000000--0000000000--0000000000--
--000-----000-----000-----000-----000-----000-----000-----000-----000-----000--
--000--00000--000-----000--000-----000--000-----000--000-----000-----0000000000--000--000--
--000--00000--0000000000--000-----000--000-----000--0000000000--0000000000--000-----000--
--000-----000--0000000000--000-----000--000-----000--0000000000--0000000000--000-----000--
--0000000000--000-----000--0000000000--0000000000--000-----0000000000--0000000000--
--0000000000--000-----000--0000000000--0000000000--000-----0000000000--0000000000--
-----
Welcome comrade, this is our server uvm050!
```

Figura 3 - Texto inserido no ficheiro `/etc/issue.net`

Como podemos ver pela imagem anterior, o documento encontra-se com uma mensagem completamente estática, pois para as conexões SSH não é possível indicar informações dinâmicas antes da autenticação, como é requerido pela *User Story* e como foi feito anteriormente para o acesso local pelo *vcenter*. Isto acontece porque aqui não temos tantas garantias de que a pessoa que está a tentar aceder é quem diz ser. Isto é o contrário do que acontece na ligação pelo *vcenter* que é utilizado como uma ligação local ao sistema, sendo esta a razão pela qual devemos estar conectados à VPN do DEI, ou à rede ISEPWLAN no ISEP para conseguir utilizar o *vcenter*, o que já implica autenticação prévia para garantirmos que pertencemos a esta mesma organização. Se fosse utilizado, por exemplo, uma script, isso poderia comprometer a segurança do sistema, pois o utilizador que está a tentar aceder ao sistema poderia explorar vulnerabilidades executando alguma script maliciosa. Acima de tudo, devemos ter em mente que o *banner* de *login* tem como função apresentar uma mensagem de boas-vindas ou algum tipo de políticas/regras, por isso deve ser mantido com um formato simples.

Dado isto, após editarmos o ficheiro com o texto desejado, precisamos de indicar na configuração do SSH, que queremos que esta mensagem neste ficheiro seja apresentada ao utilizador. O ficheiro a editar é o que se encontra no caminho `/etc/ssh/sshd_config`.

Ao editar este ficheiro, procuramos pela tag *'Banner'*, em que esta se encontrava comentada. Então, retiramos o símbolo de comentário desta tag, e à sua frente alteramos o valor de *none* para o caminho em que se encontra o ficheiro com a mensagem a apresentar, que neste caso é `/etc/issue.net`. Na imagem seguinte, é possível visualizar como fica o ficheiro após estas alterações.


```
GNU nano 5.4 /etc/ssh/sshd_config
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
PrintLastLog no
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
Banner /etc/issue.net_
```

Figura 4 - Ficheiro /etc/ssh/sshd_config com configuração do Banner

Após guardar estas alterações, devemos reiniciar o serviço de SSH para que estas alterações sejam efetivamente aplicadas, e para isso utilizamos o comando `'sudo systemctl restart ssh'`.

Vale a pena ressaltar que caso tivéssemos criado um outro ficheiro noutra local e com outro nome, na tag `'Banner'` deveríamos indicar o caminho para esse mesmo ficheiro desejado, pois não é obrigatório a utilização do ficheiro `'/etc/issue.net'`.

Agora quando fazemos uma conexão SSH ao servidor, podemos observar que já aparece a mensagem pretendida.

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\ti_ma>ssh luser1@uvm050.dei.isep.ipp.pt
-----
--0000000000--0000000000--0000000000--000---000--0000000000-----0000000000--0000000000--
--0000000000--0000000000--0000000000--000---000--0000000000-----0000000000--0000000000--
--000-----000---000--000---000--000---000--000---000-----000-----000---000---
--000--00000--000---000--000---000--000---000--000---0000000000--0000000000--000---000---
--000--00000--000000000--000---000--000---000--0000000000-----0000000000--000---000---
--000---000--000000000--000---000--000---000--0000000000-----000---000---000---
--0000000000--000---000--0000000000--0000000000--000-----0000000000--0000000000--
--0000000000--000---000--0000000000--0000000000--000-----0000000000--0000000000--
-----
Welcome comrade, this is our server uvm050!

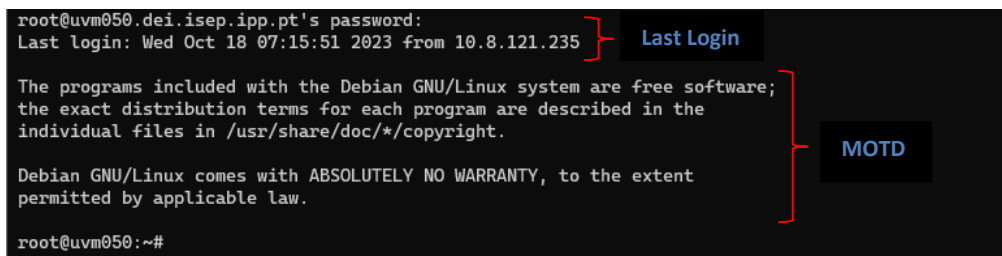
luser1@uvm050.dei.isep.ipp.pt's password:
```

Figura 5 - Mensagem antes de autenticação através de conexão SSH

User Story 2

Esta User Story irá alterar as informações exibidas no terminal após a autenticação, mostrando dados mais atraentes e úteis para o utilizador. A mensagem incluirá o nome do utilizador, a data e hora, bem como o seu último login bem-sucedido. Além disso, terá uma borda adaptada ao tamanho da mensagem e um doodle de um gato.

Passo 1: Remover mensagens de login original:

A terminal window showing the original login messages. The first line is the password prompt. The second line is the 'Last login' message, which is annotated with a red bracket and the label 'Last Login'. The third line is the MOTD (Message of the Day), which is annotated with a red bracket and the label 'MOTD'. The terminal prompt is 'root@uvm050:~#'.

```
root@uvm050.dei.isep.ipp.pt's password:
Last login: Wed Oct 18 07:15:51 2023 from 10.8.121.235

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

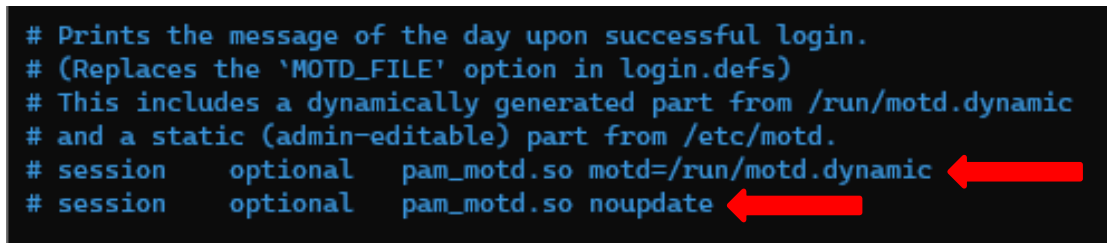
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

root@uvm050:~#
```

Figura 6 - Mensagem de login de origem

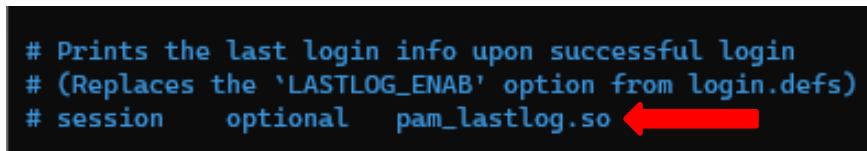
Para desativar a mensagem de login foi alterado o ficheiro **/etc/pam.d/login**. Este é o ficheiro de configuração do *Pluggable Authentication Module* (PAM), usado para autenticar utilizadores em sistemas UNIX.

Foram comentadas as seguintes linhas para desativar a *message of the day* (MOTD) e o ultimo login com sucesso, respetivamente.

A terminal window showing the contents of the /etc/pam.d/login file. The file contains several lines of configuration. Two lines are highlighted with red arrows pointing to them, indicating they should be commented out: '# session optional pam_motd.so motd=/run/motd.dynamic' and '# session optional pam_motd.so nouupdate'.

```
# Prints the message of the day upon successful login.
# (Replaces the 'MOTD_FILE' option in login.defs)
# This includes a dynamically generated part from /run/motd.dynamic
# and a static (admin-editable) part from /etc/motd.
# session optional pam_motd.so motd=/run/motd.dynamic
# session optional pam_motd.so nouupdate
```

Figura 7 - Desativação da MOTD, no ficheiro /etc/pam.d/login

A terminal window showing the contents of the /etc/pam.d/login file. The file contains several lines of configuration. One line is highlighted with a red arrow pointing to it, indicating it should be commented out: '# session optional pam_lastlog.so'.

```
# Prints the last login info upon successful login
# (Replaces the 'LASTLOG_ENAB' option from login.defs)
# session optional pam_lastlog.so
```

Figura 8 - Desativação do último login, no ficheiro /etc/pam.d/login

No entanto, logins **SSH** ainda exibem mensagem de login. Para desativá-la, foi necessário alterar o ficheiro **/etc/ssh/sshd_config**. Este é o ficheiro de configuração do **OpenSSH SSH daemon**.

Para tal, foram alterados os seguintes parâmetros, para desativar a MOTD e o último login com sucesso respetivamente

```
PrintMotd no
PrintLastLog no
```

Figura 9 - Desativação da MOTD e último login para SSH, no ficheiro `/etc/ssh/sshd_config`

Após essas alterações, é necessário reiniciar o serviço SSH com o comando **service ssh restart**.

Passo 2: imprimir uma nova mensagem customizada

Para imprimir a nossa mensagem, foi desenvolvido o seguinte **script Bash**:

```
# Welcome message
if [ -t 1 ] && [ ! -f ~/.hushlogin ]; then
    usr=$(whoami)
    last_login=$(last -F $(usr) | grep 'still logged in' | head -1 |
        awk '{print $6, $5 " ", " $8 ", " $7}' | sed 's/\([0-9][0-9]:[0-9][0-9]\):[0-9][0-9]/\1/')
    echo

    message="Welcome, $usr!
    Today is $(date +%A, %d %b, %Y)".
    It is currently $(date +%H:%M).
    I haven't seen you since $last_login.
    Have a nice day!"

    # Calculate the box width based on the longest line
    longest_line_length=0
    while IFS= read -r line; do
        line_length=$(echo -n -e "$line" | wc -c)
        if [ $line_length -gt $longest_line_length ]; then
            longest_line_length=$line_length
        fi
    done <<< "$message"
    box_width=$((longest_line_length + 2))

    # Print top border
    echo "      | \_ _ / , |      ( \ \ \ \"
    echo "      _ . | o o | _   ) )"

    line=("+")
    for ((i = 0; i < box_width; i++)); do
        if ((i >= 3 && i <= 5)) || ((i >= 9 && i <= 11)); then
            line+ "("
        else
            line+ "-"
        fi
    done
    line+ "+"

    echo $line

    # Print message with borders and line
    echo -e "$message" | while IFS= read -r line; do
        printf "| %-*s | \n" "${(box_width - 2)}" "$line"
    done

    # Print bottom border
    line= "+"
    for ((i = 0; i < box_width; i++)); do
        line+ "-"
    done
    line+ "+"
    echo $line
    echo
fi
```

Figura 10 - Bash script que imprime a mensagem de login

```
      | \_ _ / , |   ( \
    _ . | o o   | _   ) )
+---(((---(((-----+
| Welcome, root!
| Today is Wednesday, 18 Oct, 2023.
| It is currently 05:31.
| I haven't seen you since 18 Oct, 2023, 05:31.
| Have a nice day!
+-----+-----+-----+

```

Figura 11 - Exemplo da nova mensagem de login

Este script foi colocado na pasta **/etc/profile.d/**, que contem scripts que são executados na inicialização pela Shell, com o nome **welcome_message.sh**.

Como o script foi desenvolvido em Bash, os utilizadores também terão de usar a **Bash Shell**.

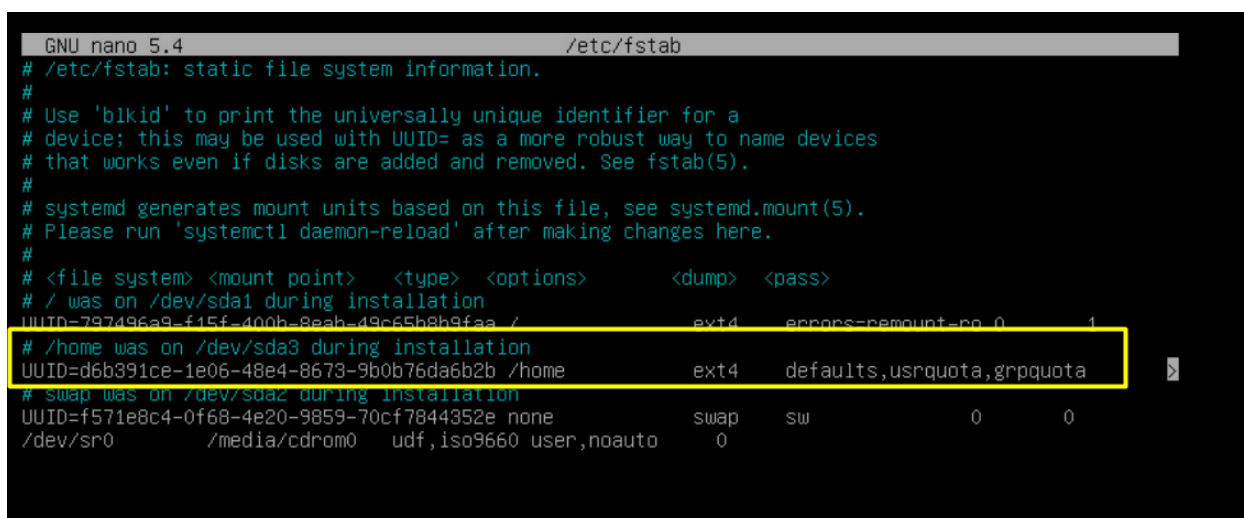
User Story 3

Na *User Story 3* pretende-se implementar uma gestão de quotas no sistema *Linux* para que os utilizadores não possam exceder os 300 ficheiros na sua área de trabalho (*home directory*). Para a automatização deste processo será criado um *script*.

Para a realização desta *US*, deve-se estar com a sessão iniciada como **root** na máquina virtual, seguidamente realizar-se os seguintes passos:

1. Habilitar as quotas no ficheiro de configuração **/etc/fstab** através do comando **"nano /etc/fstab"**.

A opção **usrquota** deve estar definida para a partição onde o diretório **home** está localizado;



```
GNU nano 5.4 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=797496a9-f15f-400b-8eab-49c65b8b9faa / ext4 errors=remount-ro 0 1
# /home was on /dev/sda3 during installation
UUID=d6b391ce-1e06-48e4-8673-9b0b76da6b2b /home ext4 defaults,usrquota,grpquota 1
# swap was on /dev/sda2 during installation
UUID=f571e8c4-0f68-4e20-9859-70cf7844352e none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0
```

Figura 12 - Habilitação das quotas

2. Remontar novamente o diretório **/home** com o comando **"mount -o remount /home"**;
3. Efetuar a contabilização inicial usando o comando **"quotacheck -u /home"**. Este comando vai percorrer todo o sistema de ficheiros da partição e contabilizar o espaço que está a ser ocupado por cada utilizador. Na raiz da partição é criado o ficheiro **aquota.user**;
4. Executar o comando **"quotaon -u /home"** que instrui o **kernel** para começar a monitorizar todas as operações de escrita e manter os ficheiros atualizados em conformidade;
5. Criar uma **script**, através do comando **"nano script_quota_us3.sh"**;
6. Nesta script deve-se realizar todo o processo para que os utilizadores não possam exceder os 300 ficheiros na sua área de trabalho. Para isso colocou-se um **for** que percorre todos os utilizadores no diretório **home**, e de seguida verifica através de um **if** se o utilizador existe, em caso afirmativo é realizado o comando **setquota** alterando apenas o parâmetro relativo ao limite **hard** do número de ficheiros para **300**;

7. Tornar o script executável através do comando "**chmod +x script_quota_us3.sh**";
8. Executar o script por meio do comando "**./script_quota_us3.sh**";

```

GNU nano 5.4                                script_quota_us3.sh

#!/bin/bash

# Caminho para o diretório home
home="/home"

# Novo valor para o limite hard do número de ficheiros
hard_limit=300

# Percorrer todos os utilizadores no diretório home
for user in $(ls $home); do
    # Verificar se o utilizador existe
    if id "$user" &>/dev/null; then
        # Aplicar setquota para alterar o limite hard do número de ficheiros
        setquota $user 0 0 0 $hard_limit $home
    fi
done

```

Figura 13 - Script para os utilizadores não excederem os 300 ficheiros na sua área de trabalho

Por fim, deve-se apenas testar a solução implementada para verificar se realiza o pretendido através do comando "**repquota --all**". Este comando permite obter um relatório do estado das quotas de todos os utilizadores numa única operação.

Na seguinte figura é possível verificar que a solução implementada foi bem-sucedida, pois o parâmetro relativo ao limite *hard* do número de ficheiros da área de trabalho dos utilizadores encontra-se com o valor desejado.

```

root@uvm050:~# repquota --all
*** Report for user quotas on device /dev/sda3
Block grace time: 7days; Inode grace time: 7days

```

User		Block limits				File limits			
		used	soft	hard	grace	used	soft	hard	grace
root	--	20	0	0		2	0	0	
asist	--	20	0	0		5	0	300	
luser1	--	20	0	0		5	0	300	
luser2	--	4	0	0		1	0	300	
luser3	--	4	0	0		1	0	300	
luser4	--	4	0	0		1	0	300	
luser5	--	4	0	0		1	0	300	
luser6	--	4	0	0		1	0	300	

Figura 14 - Visualização das quotas de todos os utilizadores

User Story 4

Esta US pede como administrador do sistema quero implementar uma gestão de quotas no sistema Windows para que uma pasta de partilha de ficheiros (que deve ser criada) não possa conter mais do que 10MB de informação, avisando-me por email se estiver prestes a ser alcançado esse limite.

Criar pasta

Para isso temos de criar uma pasta destinada a esta função, chamar-se á “pasta xpto das partilhas” e estará localizada em “C:\”.

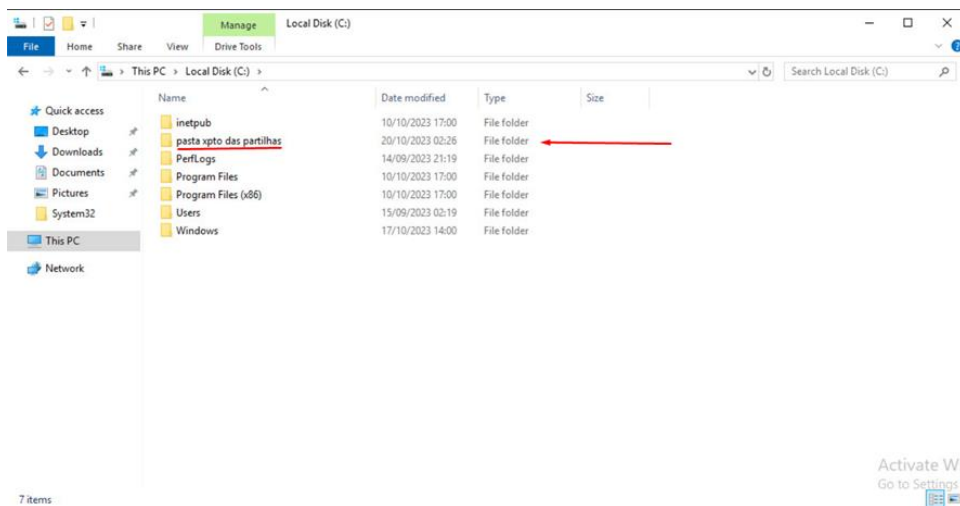


Figura 15 - Local de criação da pasta partilhada

Partilhar a pasta

Depois clica-se com o botão direito em cima da pasta criada e acesse às propriedades da mesma. Na aba “sharing” clica-se em “Advanced Sharing”:

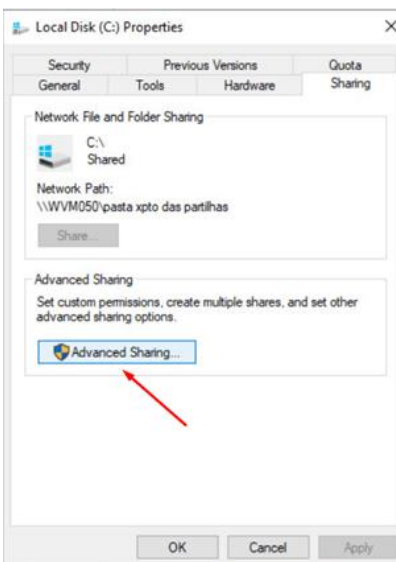


Figura 16 - Onde aceder às definições avançadas de partilha

De seguida criei uma “New Share”

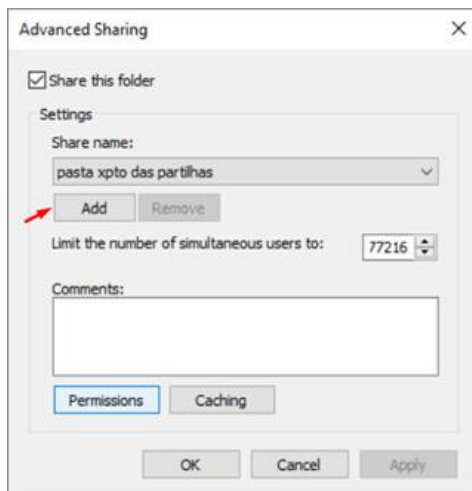


Figura 17 - Onde se cria uma nova partilha

e atribui-se as permissões adequadas.

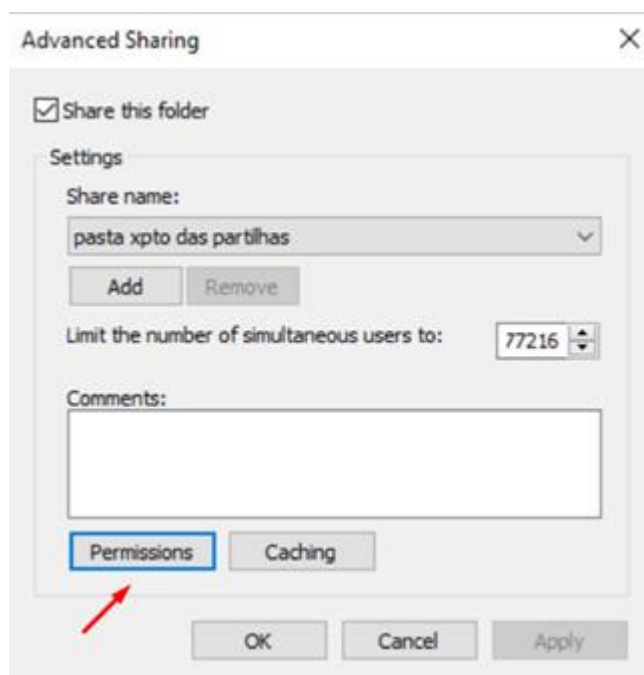


Figura 18 - Onde se gera as permissões da nova partilha

Criar quota e mensagem do email:

Para criar uma quota é necessário ir ao “File server Resource Manager” e ao abrir a aba “Quota Management” selecionar “Quota Templates” e selecionar “Create Quota Template”:

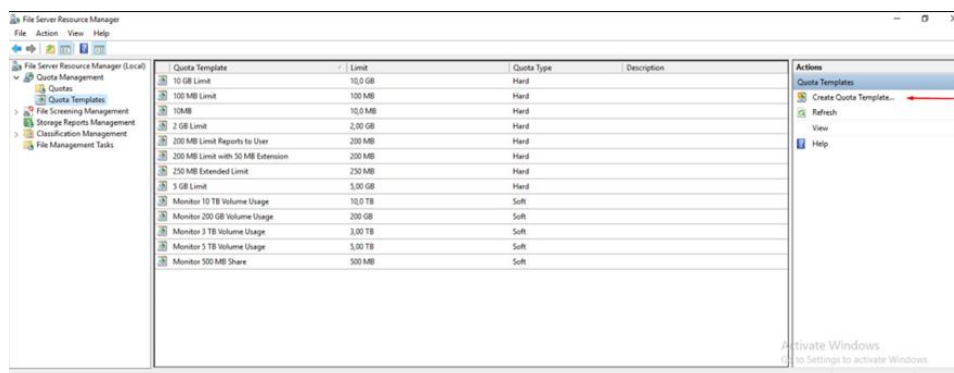


Figura 19 - Onde cria-se um novo template de quota

Cria-se um limite personalizado de 10MB:

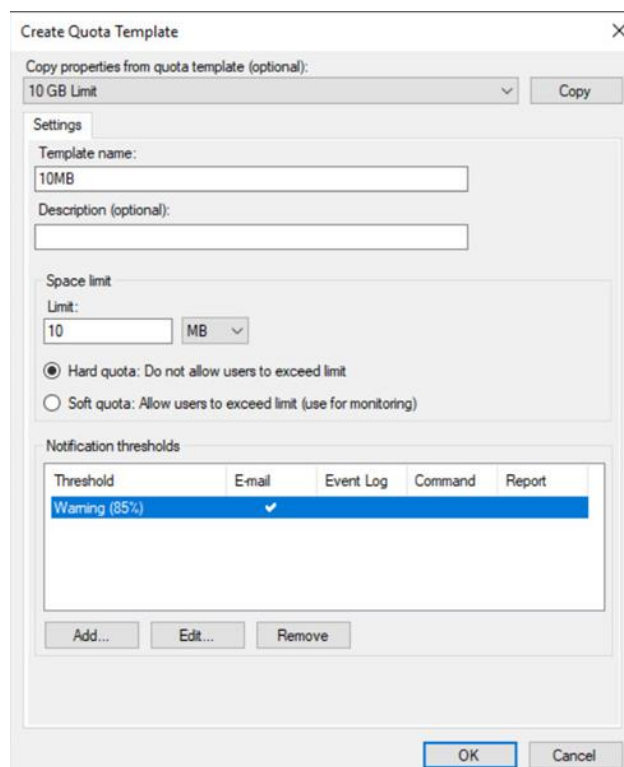


Figura 20 - Onde e como se configura um novo template de quota

Que manda um email personalizado, quando se chegar a 85% de espaço ocupado, como se vê na figura:

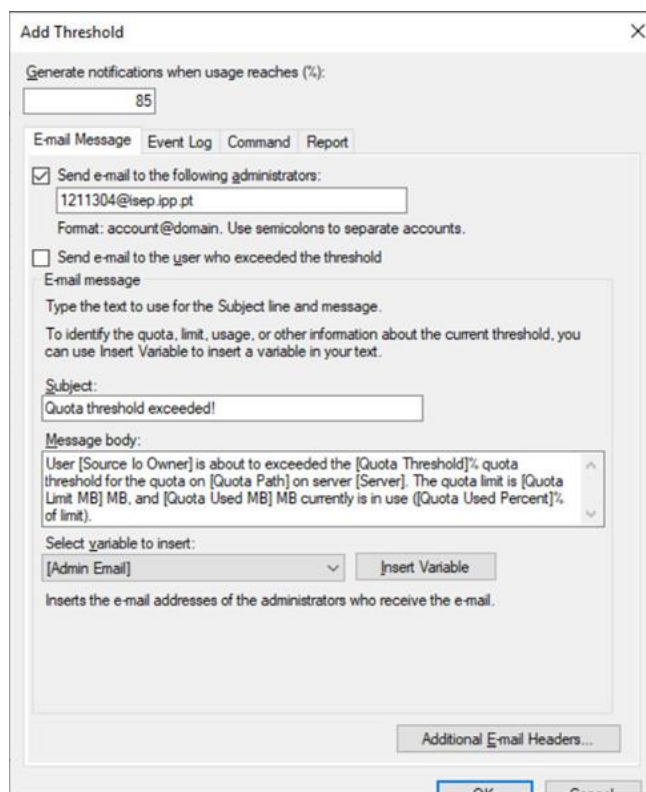


Figura 21 - O email enviado quando o uso da pasta partilhada atinge determinado limite

Para este trabalho o email não é efetivamente enviado por causa do servidor smtp, ficando o email na queue do windows.

Testar a quota:

Como se pode ver na figura depois se criar um ficheiro uma segunda vez o qual faria ultrapassar o limite de 10MB, a ação é negada pois não é possível passar o limite.

```
C:\pasta xpto das partilhas>fsutil file createnew test.txt 9000000
File C:\pasta xpto das partilhas\test.txt is created

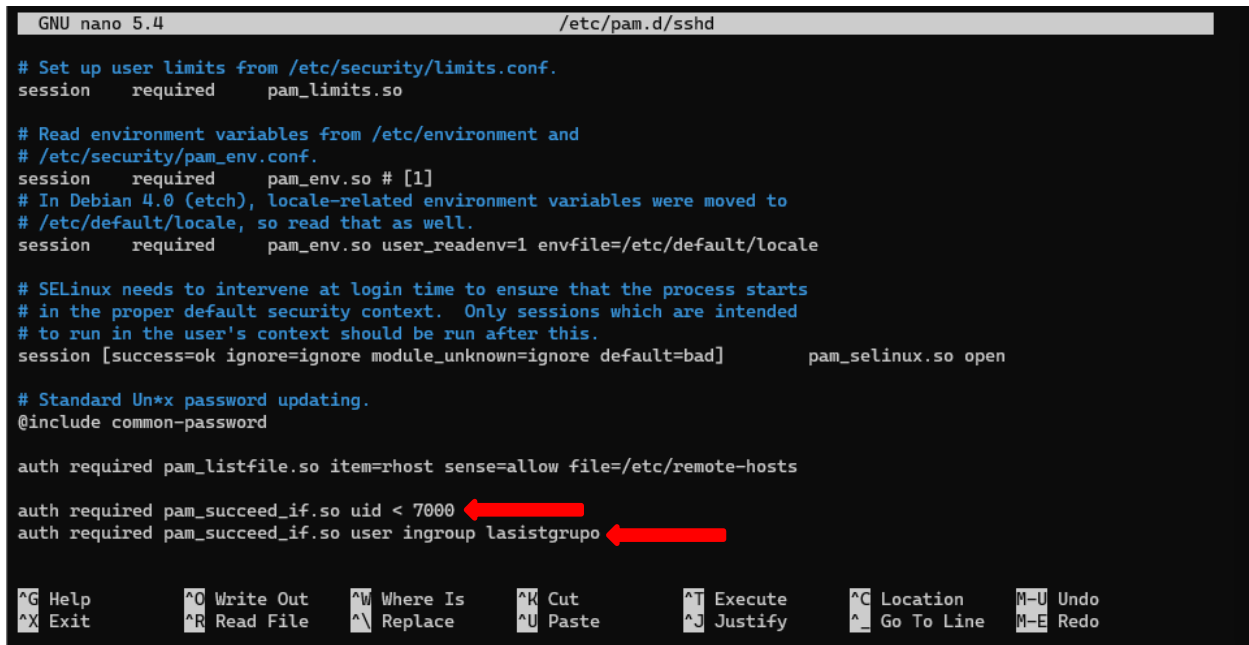
C:\pasta xpto das partilhas>fsutil file createnew test1.txt 9000000
Error: There is not enough space on the disk.
```

Figura 22 - Tentativa de ultrapassar a quota

User Story 5

A User Story tem o objetivo de limitar o acesso ao sistema, permitindo apenas o acesso a utilizadores com um **UID inferior a 7000** e que pertençam ao **grupo "lasistgrupo"**.

Para impor essas restrições, o seguinte conteúdo foi adicionado ao ficheiro **/etc/pam.d/sshd**. Estas restrições apenas aplicam-se apenas a acessos via SSH.



```
GNU nano 5.4 /etc/pam.d/sshd

# Set up user limits from /etc/security/limits.conf.
session required pam_limits.so

# Read environment variables from /etc/environment and
# /etc/security/pam_env.conf.
session required pam_env.so # [1]
# In Debian 4.0 (etch), locale-related environment variables were moved to
# /etc/default/locale, so read that as well.
session required pam_env.so user_readenv=1 envfile=/etc/default/locale

# SELinux needs to intervene at login time to ensure that the process starts
# in the proper default security context. Only sessions which are intended
# to run in the user's context should be run after this.
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open

# Standard Unix password updating.
@include common-password

auth required pam_listfile.so item=rhost sense=allow file=/etc/remote-hosts

auth required pam_succeed_if.so uid < 7000
auth required pam_succeed_if.so user ingroup lasistgrupo

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

Figura 23 - Novos campos no ficheiro **/etc/pam.d/sshd** para impor as restrições

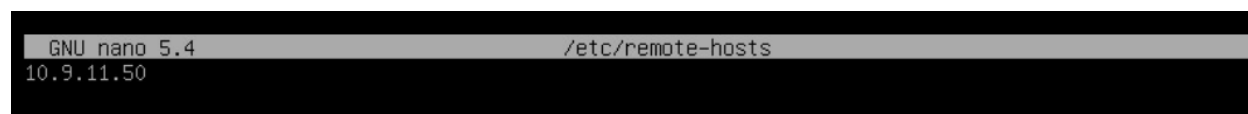
Estes campos limitam o restringem o acesso a utilizadores com **UID inferior a 7000**, e a utilizadores que pertencem ao grupo **"lasistgrupo"** respetivamente.

User Story 6

Na *User Story 6* pretende-se condicionar o acesso ao sistema *Linux*, permitindo apenas as máquinas remotas que constem no ficheiro */etc/remote-hosts*. Este condicionamento aplica-se apenas aos acessos via *SSH*.

Para a realização desta *US*, deve-se estar com a sessão iniciada como **root** na máquina virtual, seguidamente realiza-se os seguintes passos:

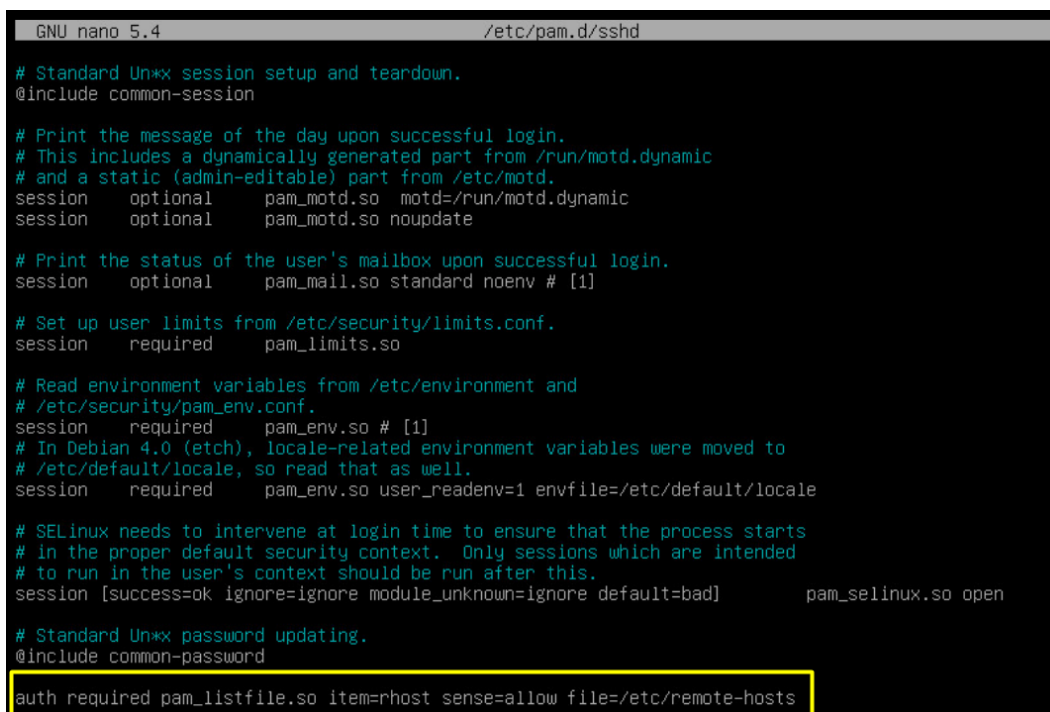
1. Criar o ficheiro */etc/remote-hosts*, através do comando **"nano /etc/remote-hosts"**;
2. Neste ficheiro deve-se adicionar os endereços *IP* (um por linha) confiáveis que vamos permitir aceder ao sistema. Neste caso adicionou-se o *IP* do nosso servidor *Windows*;



```
GNU nano 5.4 /etc/remote-hosts
10.9.11.50
```

Figura 24 - Endereço IP adicionado no ficheiro */etc/remote-hosts*

3. Aceder ao ficheiro */etc/pam.d/sshd*, através do comando **"nano /etc/pam.d/sshd"**;
4. Neste ficheiro deve-se adicionar ao conteúdo já presente o comando **"auth required pam_listfile.so item=rhost sense=allow file=/etc/remote-hosts"** para permitir o acesso apenas às máquinas remotas que constem no ficheiro */etc/remote-hosts*;



```
GNU nano 5.4 /etc/pam.d/sshd

# Standard Un*x session setup and teardown.
@include common-session

# Print the message of the day upon successful login.
# This includes a dynamically generated part from /run/motd.dynamic
# and a static (admin-editable) part from /etc/motd.
session optional pam_motd.so motd=/run/motd.dynamic
session optional pam_motd.so noupdate

# Print the status of the user's mailbox upon successful login.
session optional pam_mail.so standard noenv # [1]

# Set up user limits from /etc/security/limits.conf.
session required pam_limits.so

# Read environment variables from /etc/environment and
# /etc/security/pam_env.conf.
session required pam_env.so # [1]
# In Debian 4.0 (etch), locale-related environment variables were moved to
# /etc/default/locale, so read that as well.
session required pam_env.so user_readenv=1 envfile=/etc/default/locale

# SELinux needs to intervene at login time to ensure that the process starts
# in the proper default security context. Only sessions which are intended
# to run in the user's context should be run after this.
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open

# Standard Un*x password updating.
@include common-password

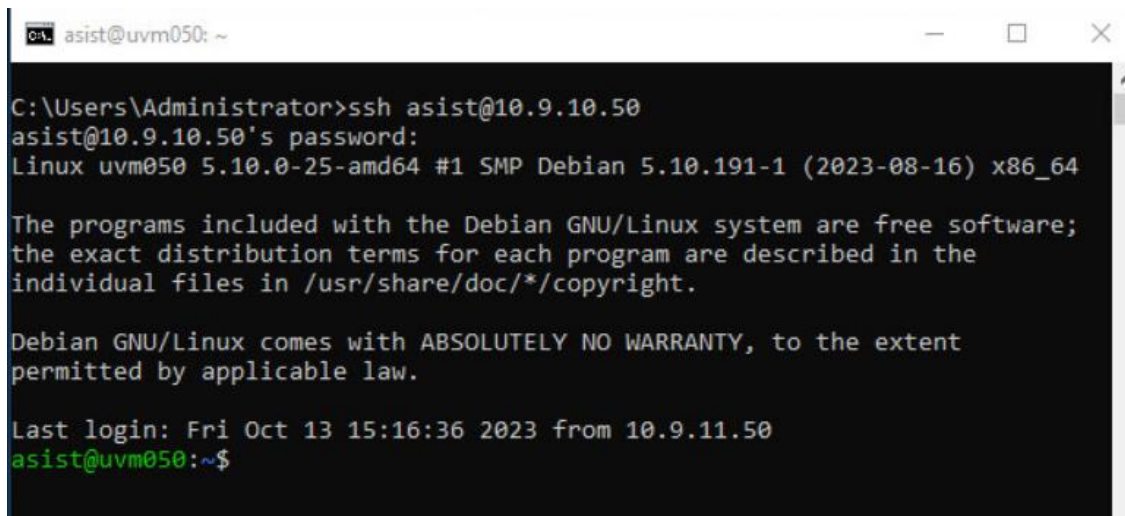
auth required pam_listfile.so item=rhost sense=allow file=/etc/remote-hosts
```

Figura 25 - Comando adicionado no ficheiro */etc/pam.d/sshd*

Por fim, deve-se apenas testar a conexão por *SSH* para verificar se a solução implementada realiza o pretendido.

Como verificado anteriormente, apenas foi autorizado o acesso via *SSH* através do nosso servidor *Windows*. Por esse motivo testa-se nele a realização de uma conexão através do comando "**ssh asist@10.9.10.50**" na linha de comandos. Neste caso está a realizar-se a autenticação com o utilizador **asist**.

Neste 1º teste realizado é expectável o sucesso da conexão após a autenticação do utilizador como verifica-se na seguinte figura.



```
C:\Users\Administrator>ssh asist@10.9.10.50
asist@10.9.10.50's password:
Linux uvm050 5.10.0-25-amd64 #1 SMP Debian 5.10.191-1 (2023-08-16) x86_64

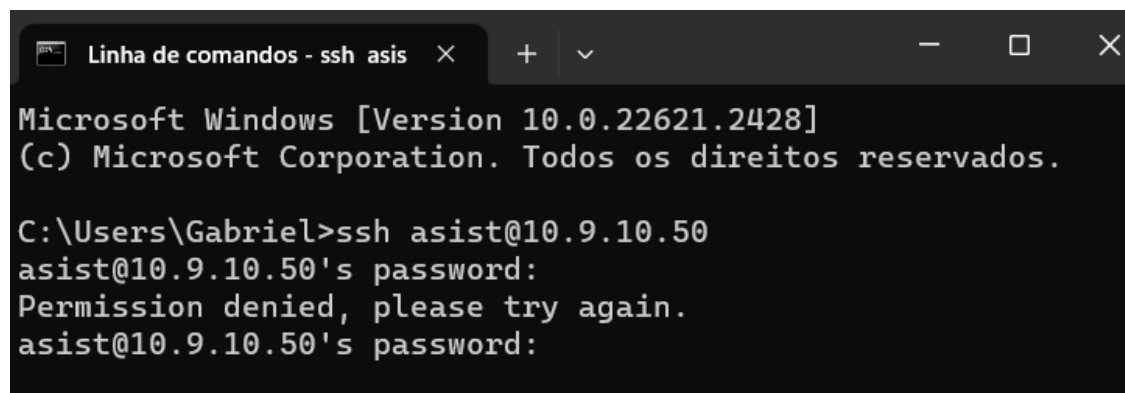
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Fri Oct 13 15:16:36 2023 from 10.9.11.50
asist@uvm050:~$
```

Figura 26 - Teste de uma conexão SSH bem-sucedida

No 2º teste realizado, testa-se a realização de uma conexão através da nossa máquina pessoal. Como o *IP* desta não está inserido no ficheiro **/etc/remote-hosts** é expectável o insucesso da conexão como verifica-se na seguinte figura, pois o acesso é negado após a autenticação do utilizador.



```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Gabriel>ssh asist@10.9.10.50
asist@10.9.10.50's password:
Permission denied, please try again.
asist@10.9.10.50's password:
```

Figura 27 - Teste de uma conexão SSH malsucedida

User Story 7

Na US7 pretende-se, como administrador do sistema usar no sistema Linux o módulo PAM “pam_listfile.so” para condicionar o acesso ao sistema, negando o acesso ao sistema aos utilizadores (um por linha) listados no ficheiro (que deve ser criado) /etc/bad-guys.

Para isso foi criado o ficheiro bad-guys, e nele foi adicionado o user asist:

```
root@uvm050:~# cat /etc/bad-guys
asist
```

Figura 28 - Os conteúdos do ficheiro bad-guys

Depois no ficheiro “/etc/pam.d/sshd” foi adicionado o seguinte comando, para ir ao ficheiro bad-guys buscar os users para bloquear:

```
#US7 - restrict users on file bad-guys
auth required pam_listfile.so \
onerr==succeed item=user sense=deny file=/etc/bad-guys
```

Figura 29 - O comando para bloquear os users identificados no ficheiro bad-guys

E depois reiniciou-se o sshd, para as alterações entrarem em efeito:

```
root@uvm050:~# systemctl restart sshd
```

Figura 30 - Comando para o reinício do sshd

E desta forma o acesso ao servidor pelo user asist foi negado:

```
C:\Windows\system32>ssh asist@10.9.10.50
-----
--0000000000--0000000000--0000000000--000--000--0000000000-----0000000000--0000000000--
--0000000000--0000000000--0000000000--000--000--0000000000-----0000000000--0000000000--
--000-----000--000--000--000--000--000--000--000--000--000-----000--000--000--
--000--000000--000--000--000--000--000--000--000--000--000-----0000000000--000--000--
--000--000--000000000--000--000--000--000--0000000000-----0000000000--000--000--000--
--0000000000--000--000--0000000000--0000000000--000-----0000000000--0000000000--
--0000000000--000--000--0000000000--0000000000--000-----0000000000--0000000000--
-----
Welcome comrade, this is our server uvm050!

asist@10.9.10.50's password:
Permission denied, please try again.
```

Figura 31 - Prova da funcionalidade da US

User Story 8

Na *User Story 8* é pedido que seja utilizado o módulo PAM “*pam_cracklib.so*”, para obrigar os utilizadores a terem uma palavra-passe complexa. Mas antes de prosseguirmos, precisamos de definir primeiro o que é uma palavra-passe complexa.

Normalmente quando nos referimos a uma palavra-passe complexa, estamos a assumir que será difícil de adivinhar, e que esta ajuda a prevenir que pessoas não autorizadas entrem em certa conta ou sistema. Para se definir uma palavra-passe complexa, temos de impor alguns requisitos mínimos aquando da sua criação. Estes requisitos podem ser a combinação de letras maiúsculas e minúsculas, números, símbolos especiais (!, @, \$, entre outros) e um comprimento mínimo significativo. Com a definição destes requisitos, conseguimos eliminar algumas palavra-passes fáceis de adivinhar, como por exemplo, sequências numéricas (por exemplo “1235678”) e palavras reais e comuns. (por exemplo, nomes de animais de estimação, datas de nascimento, entre muitas outras).

Para além da obrigação de utilização de uma palavra-passe complexa, devem ser adotados outros mecanismos que ajudem a manter uma conta segura. Alguns desses mecanismos podem ser:

- a expiração da palavra-passe de X em X tempo, obrigando a que esta seja trocada frequentemente;
- evitar utilizar palavra-passes antigas, ou seja, que já foram utilizadas anteriormente;
- limitar o número de tentativas de acesso à conta, ou seja, se errar a palavra-passe X vezes, fica com a conta bloqueada.

Dados isto, vamos então agora definir os requisitos mínimos de palavra-passe que queremos que os utilizadores do nosso sistema Linux sigam. Os requisitos serão então:

- comprimento mínimo de 8 caracteres;
- pelo menos 1 letra minúscula;
- pelo menos 1 letra maiúscula;
- pelo menos 1 número;
- pelo menos 1 símbolo especial.

Para aplicar estes requisitos, tivemos de instalar o package da biblioteca *pam_cracklib*, em que para isso, executamos o seguinte comando: `'sudo apt-get install libpam-cracklib'`.

Após instalar esta biblioteca, editamos o ficheiro `'etc/pam.d/common-password'`. No ficheiro, já se encontrava uma configuração predefinida, como podemos ver pela imagem abaixo.

```
# here are the per-package modules (the "Primary" block)
password      requisite      pam_cracklib.so retry=3 minlen=8 difok=3
password      [success=2 default=ignore] pam_unix.so obscure use_authok try_first_pass yescrypt
```

Figura 32 - Configuração predefinida para a biblioteca *pam_cracklib*

A primeira linha após o comentário é a responsável por ativar a biblioteca *pam_cracklib*, enquanto a segunda linha invoca o módulo standard *pam_unix*.

Definimos então os requisitos anteriormente combinados para as palavra-passe.

```
# here are the per-package modules (the "Primary" block)
password      requisite      pam_cracklib.so minlen=8 lcredit=-1 ucredit=-1 dcredit=-1 ocredit=-1
password      [success=2 default=ignore] pam_unix.so obscure use_authok try_first_pass yescrypt
```

Figura 33 - Configuração de requisitos mínimos para as palavra-passe com a biblioteca *pam_cracklib*

Através da imagem anterior, na primeira linha, podemos ver que definimos novos campos para o módulo *pam_cracklib*, sendo este que irá fazer as validações das palavra-passe:

- *minlen* – Define o tamanho mínimo de caracteres;
- *lcredit* – Define a quantidade obrigatória de letras minúsculas;
- *ucredit* – Define a quantidade obrigatória de letras maiúsculas;
- *dcredit* – Define a quantidade de valores numéricos obrigatórios;
- *ocredit* – Define a quantidade de caracteres especiais obrigatórios.

Podemos observar que é utilizada a opção -1 nestes 4 últimos casos (*lcredit*, *ucredit*, *dcredit*, *ocredit*). Fazemos isto porque ao atribuir um número negativo estamos a definir a quantidade de valores obrigatórios que devem ser inseridos pelo utilizador nesse mesmo requisito. Se colocarmos um valor positivo, estamos a dizer quantos créditos no máximo a inserção daquele tipo de caractere irá valer. Por exemplo, se o *lcredit* tivesse o valor 2, o utilizador ao inserir duas letras minúsculas, elas iam contar como 4 caracteres para o tamanho mínimo da palavra-passe (*minlen*), pois cada uma iria valer mais um crédito do que aquilo que vale. Mas se por exemplo, em vez duas letras minúsculas fossem inseridas três, no total iam contar como cinco, pois o máximo de créditos já tinha sido atingido, ou seja, os dois primeiros caracteres, contavam em conjunto como quatro, mas depois o outro apenas contaria como um, pois o máximo de créditos tinha sido atingido.

Já na configuração do módulo *pam_unix*, temos definido o valor *yescrypt* que ativa o algoritmo *yescrypt* para calcular a *hash* para as palavra-passe. Temos também o *use_authok*, em que este diz ao módulo *pam_unix* que já não precisa de fazer verificações (muitas vezes duplicadas) na palavra-passe, e sim apenas aceitá-la depois de ter sido verificada pelo *pam_cracklib*. A opção *obscure* será removida, pois isto faz com que este módulo faça algumas verificações que já não são necessárias devido a estarmos a utilizar

o módulo *pam_cracklib*. Já o *try_first_pass* é usado para indicar que o PAM deve tentar usar a palavra-passe fornecida anteriormente durante a mesma autenticação para autenticar o utilizador.

Adicionamos também mais alguns requisitos que ajudam a manter a palavra-passe mais segura, sendo estes:

- o número de caracteres que devem diferir entre a palavra-passe atual e a nova;
- o número máximo de tentativas para alterar a palavra-passe antes de abortar o processo;
- o número de palavra-passes antigas que são recordadas e que não podem ser repetidas enquanto não forem esquecidas pelo sistema;
- por fim, definir de quanto em quanto tempo o utilizador é obrigado a alterar a sua palavra-passe.

No módulo *pam_cracklib*, começamos então por adicionar o campo *'difok=4'*, em que este define que o utilizador ao alterar a sua palavra-passe tem de colocar pelo menos 4 caracteres diferentes dos que tem na sua palavra-passe atual (a ordem também conta). Adicionamos também o campo *'retry=3'*, em que apenas permite que o utilizador tente 3 vezes alterar a palavra-passe para uma que respeite os requisitos mínimos. De notar que caso ele falhe essas 3 vezes, ele poderá voltar a usar o comando *passwd* para tentar alterar a sua palavra-passe.

Para que o utilizador não possa repetir as suas 2 últimas palavra-passe, adicionamos o campo *'remember=2'*, sendo que este terá de ser na parte do módulo *pam_unix*. Para fazer isto, tivemos também de garantir que o ficheiro *'etc/security/opasswd'* existia, pois é aqui que as palavra-passe antigas do utilizador serão armazenadas. No nosso caso, este ficheiro já estava criado, caso contrário teria de ser criado e deveriam ser atribuídas as permissões 600 (*read* e *write* para o *owner*, que é o *root*, e sem permissões para os restantes).

```
# here are the per-package modules (the "Primary" block)
password      requisite      pam_cracklib.so minlen=8 lcredit=-1 ucredit=-1 dcredit=-1 ocredit=-1 difok=4 retry=3
password      [success=2 default=ignore] pam_unix.so use_authtok try_first_pass yescrypt remember=2
```

Figura 34 - Configuração completa do *pam_cracklib* e *pam_unix* para definir a complexidade das palavra-chave

Para obrigar os utilizadores a alterarem a sua palavra-passe de X em X tempo, editamos o ficheiro *'/etc/login.defs'* alterando alguns parâmetros, sendo esses:

- *PASS_MAX_DAYS* – A frequência, em dias, em que o utilizador tem de alterar a sua palavra-passe;
- *PASS_MIN_DAYS* – O tempo mínimo, em dias, que o utilizador tem de estar com a palavra-passe nova que acabe de definir;
- *PASS_WARN_AGE* – Representa o número de dias antes da palavra-passe expirar, em que o utilizador é avisado.

No nosso caso, deixamos configurado para que o utilizador não tenha de alterar a palavra-passe tão cedo, devido a não ser necessário pois poderíamos perder facilmente o rasto às palavra-passe dos utilizadores que nos poderão ser necessário aceder nos futuros *sprints*/trabalhos.

```
#
# Password aging controls:
#
#     PASS_MAX_DAYS   Maximum number of days a password may be used.
#     PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#     PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
```

Figura 35 - Configuração da frequência de expiração das palavra-passe

Nota: Estes parâmetros agora aplicados à expiração de palavra-passe, depois de alterada, apenas é aplicada aos utilizadores que são criados a partir desse momento com o comando *useradd*. Caso se queira aplicar a utilizadores já existentes, essa configuração terá de ser alterada manualmente na conta do próprio utilizador.

Agora ao tentar mudar a palavra-passe, podemos verificar que o sistema irá recusar a alteração caso esta não tenha em conta os novos requisitos para as palavra-passe. Para além disso, o sistema também dá *feedback* ao utilizador sobre a nova palavra-passe inserida caso esta não cumpra os requisitos.

```
luser1@uvm050:~$ passwd
Changing password for luser1.
Current password:
New password:
BAD PASSWORD: it is WAY too short
New password:
BAD PASSWORD: is too simple
New password:
BAD PASSWORD: is too similar to the old one
passwd: Have exhausted maximum number of retries for service
passwd: password unchanged
luser1@uvm050:~$
```

Figura 36 - Tentativa de alteração de palavra-passe que não cumpre os requisito mínimos

```
luser1@uvm050:~$ passwd
Changing password for luser1.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

Figura 37 - Alteração de palavra-passe para uma que cumpre os requisitos mínimos

À parte: Na minha opinião, por vezes é mais benéfico deixar o utilizador livre para escolher as suas palavra-passe, e em vez de lhe impor requisitos mínimos, como por exemplo o tamanho mínimo, devemos antes educá-lo a utilizar boas práticas na escolha das suas palavra-passe. Isto é justificado pela razão de que se alguém tentar fazer um ataque de força bruta e souber os requisitos mínimos de palavra-passe, pode eliminar algumas tentativas no seu algoritmo, fazendo com que o ataque seja feito de forma mais eficiente. Um exemplo disso é, se houver um mínimo de 4 caracteres para a palavra-passe, o algoritmo de força bruta apenas necessitará de começar a gerar tentativas a partir dos 4 caracteres. Claro que este tópico não é trivial, mas pode ser uma boa abordagem a aplicar em certas situações e com o grupo certo de pessoas.