

# SPRINT2 ASIST

## **Grupo 46**

José Mota (1161263)

Pedro Real (1170689)

João Flores (1171409)

Patrick Timas (1171352)

Dezembro, 2019

## Casos de uso

| Casos de uso |   |
|--------------|---|
| 1            | Como administrador da infraestrutura quero que seja criada uma SAN iSCSI nos servidores Linux e Windows disponíveis para qualquer utilizador autenticado  |
| 2            | Como administrador da infraestrutura quero que a SAN anterior esteja disponível sem necessidade de intervenção humana após um reboot de qualquer dos servidores   |
| 3            | Como administrador do servidor Linux quero que semanalmente seja verificado se todos os utilizadores registados em <b>/etc/passwd</b> possuem uma entrada no <b>/etc/shadow</b> , se o grupo primário existe, se a homedir existe e pertence ao dono e grupo correto. Qualquer inconformidade deve ser registada em <b>/tmp/auth_errors</b> |
| 4            | Como administrador da infraestrutura quero que todos os utilizadores registados no DC Windows tenham a sessão bloqueada ao fim de 1 minuto de inatividade   |

# Índice

|                                     |    |
|-------------------------------------|----|
| Caso de uso 1 e Caso de uso 2 ..... | 4  |
| Caso de uso 3 .....                 | 14 |
| Caso de uso 4 .....                 | 16 |

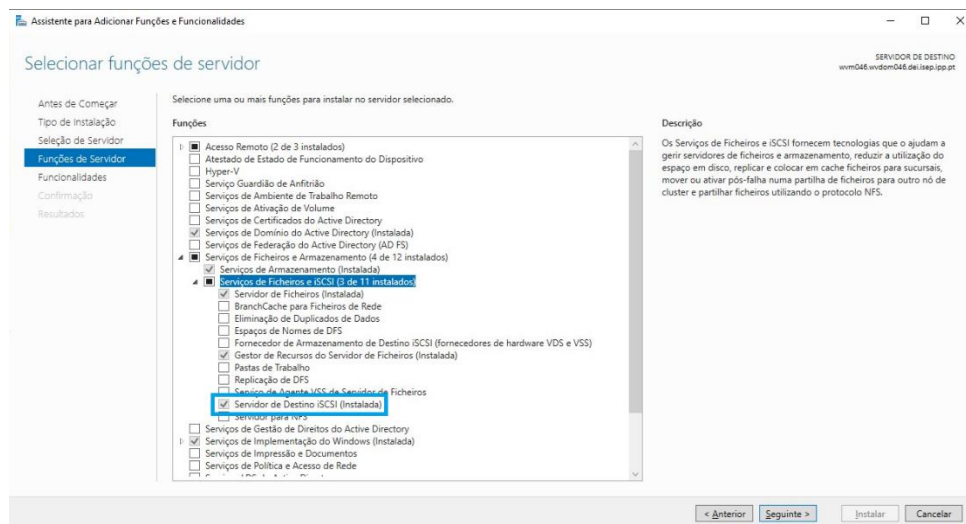
## Caso de uso 1 / 2

Para a realização desse Caso de uso foi necessária a criação de duas SANs, sendo que para isto foram configurados:

- 1 servidor de storage (target) para servidor (Windows e Linux)
- 1 cliente de storage (initiator) para servidor (Windows e Linux)

### Target Windows

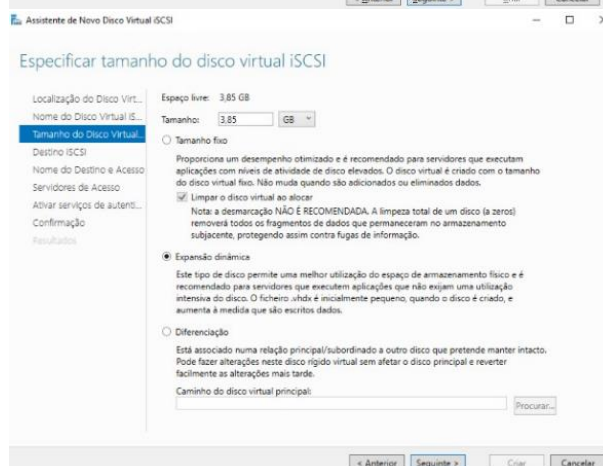
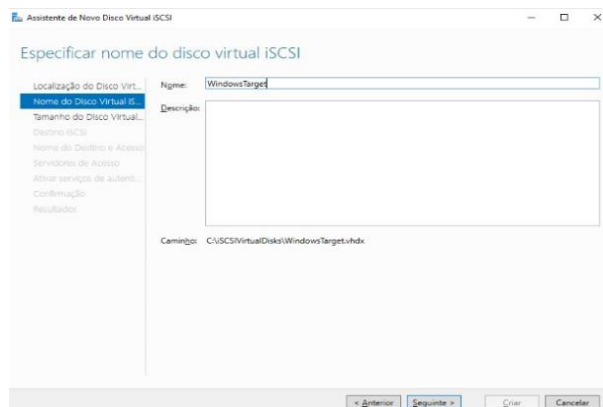
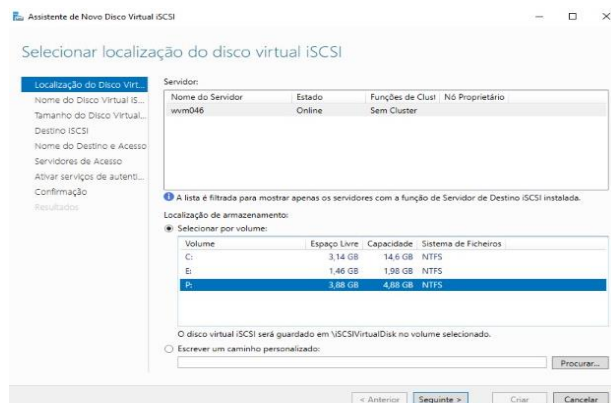
Na configuração do Target no servidor Windows deve verificar se a função de “Servidor de Destino iSCSI” está instalada, caso contrário, deve ser instalada.



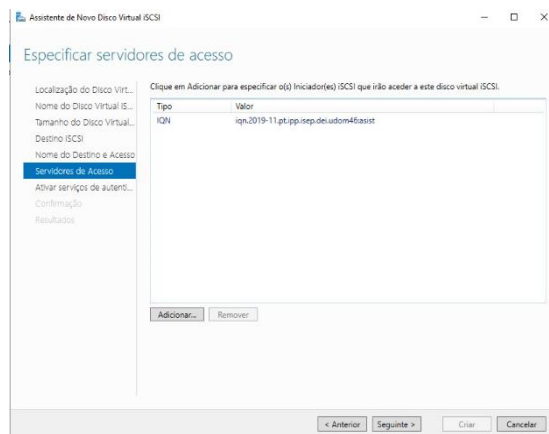
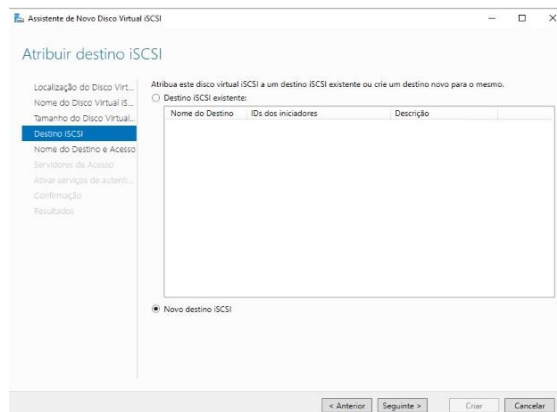
Com a função instalada, passa-se para a configuração do Target para que possa ser utilizado pelos clientes (Initializers).

Para isto, cria-se um disco virtual iSCSI no “Gestão de Discos” que vai ser disponibilizado para utilização por parte dos Initializers.

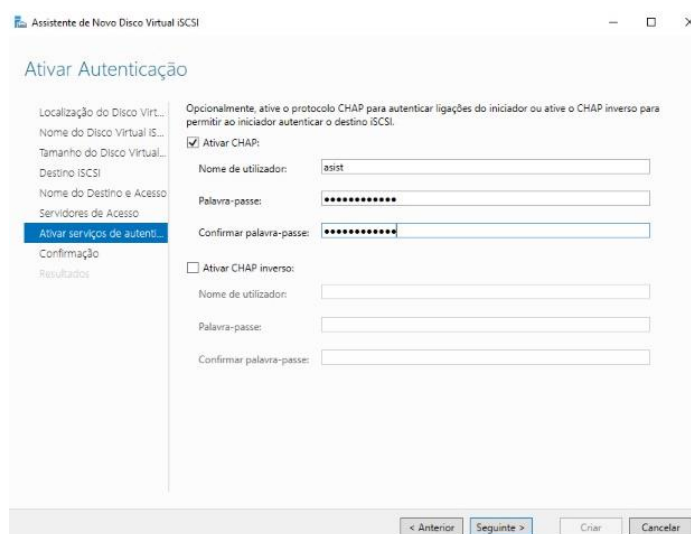
Escolhe-se o disco a qual se pretende na utilização do Initiator, dar um nome e atribuir um tamanho que será disponibilizado.



Cria-se um novo destino iSCSI onde será do tipo IQN e tem de ter um valor. No nosso caso valor foi “iqn-2019-11.pt.ipp.isep.dei.udom46:asist” este que será usado no Initiator do Linux.



Para ativar a autenticação deve-se ativar o CHAP (mecanismo de autenticação utilizado) e criar as credenciais pretendidas.



O Target em Windows está configurado, sendo agora necessário criar o Initiator em Linux e conectar este ao Target configurado.

## Initiator Linux

Na configuração do Initiator em Linux, instala-se o package “open-iscsi” com o comando “**apt-get install open-iscsi**”.

De seguida, edita-se o ficheiro **/etc/iscsi/initiatorname.iscsi** para introduzir o nome do Initiator (nome este o que foi utilizado para indicar o initiator na configuração do Target Windows configurado anteriormente).

```
GNU nano 2.9.3 /etc/iscsi/initiatorname.iscsi
## DO NOT EDIT OR REMOVE THIS FILE!
## If you remove this file, the iSCSI daemon will not start.
## If you change the InitiatorName, existing access control lists
## may reject this initiator. The InitiatorName must be unique
## for each iSCSI initiator. Do NOT duplicate iSCSI InitiatorNames.
#InitiatorName=iqn.1993-08.org.debian:01:219b5d4f819a
InitiatorName=iqn.2019-11.pt.1pp.isep.dei.udom46:asist
```

Após a definição do nome, edita-se o ficheiro **/etc/iscsi/iscsid.conf** para configurar o método e credenciais de autenticação (CHAP) e também a

```
GNU nano 2.9.3 /etc/iscsi/iscsid.conf
# To enable CHAP authentication set node.session.auth.authmethod
# to CHAP. The default is None.
node.session.auth.authmethod = CHAP
# To set a CHAP username and password for initiator
# authentication by the target(s), uncomment the following lines:
node.session.auth.username = asist
node.session.auth.password = asist2019gp46
# To set a CHAP username and password for target(s)
# authentication by the initiator, uncomment the following lines:
#node.session.auth.username_in = username_in
#node.session.auth.password_in = password_in
# To enable CHAP authentication for a discovery session to the target
```

Para ter uma ligação automática ao Target quando o sistema inicia, altera-se o startup para o automático.

```
GNU nano 2.9.3 /etc/iscsi/iscsid.conf
#*****
# Startup settings
#*****
# To request that the iscsi initd scripts startup a session set to
# node.startup = automatic
#
# To manually startup the session set to "manual". The default is
node.startup = automatic
# For "automatic" startup nodes, setting this to "Yes" will try to
# available iface until one succeeds, and then stop. The default
# logins on all available ifaces simultaneously.
node.leading_login = No
```

Concluída a edição dos ficheiros, executa-se o comando **“systemctl restart iscsid open-iscsi”** para aplicar as alterações efetuadas.

Para conectar o Initiator ao Target, executa-se o comando **“iscsiadm -m discovery -t sendtargets -p TARGETIP”** (onde o TARGETIP é o ip do servidor onde se encontra o target desejado, no caso é 10.9.11.46) para detetar os targets disponíveis.

```
asist@uvm046:/$ sudo iscsiadm -m discovery -t sendtargets -p 10.9.11.46
10.9.11.46:3260,1 iqn.2019-11.pt.ipp.isep.dei.udom46:asistw
asist@uvm046:/$
```

Sendo agora possível ver os detalhes da sessão estabelecida através do comando **“iscsiadm -m session -o show”**.

```
asist@uvm046:/$ sudo iscsiadm -m discovery -t sendtargets -p 10.9.11.46
10.9.11.46:3260,1 iqn.2019-11.pt.ipp.isep.dei.udom46:asistw
asist@uvm046:/$ iscsiadm -m session -o show
tcp: [1] 10.9.11.46:3260,1 iqn.2019-11.pt.ipp.isep.dei.udom46:asistw (non-flash)
asist@uvm046:/$
```

Estabelecida a conexão ao Target, segue-se a configuração do disco a utilizar. Para isso, foi criada uma partição primária **/dev/sdb1** com recurso ao comando **“fdisk”** e seus recursos, depois criamos um filesystem nessa partição do tipo ext4 com o comando **“mkfs.ext4 /dev/sdb1”**. Para finalizar executamos o comando **“mount /dev/sdb1 /asist/disk1”** e adicionamos uma entrada para este disco no ficheiro **“/etc/fstab”** para que o mount deste disco seja executado automaticamente no arranque do servidor.

```
GNU nano 2.9.3 /etc/fstab
# UUID=86f5331f-21cc-49b4-bd8d-27dc149b5cee none swap sw 0 0
UUID=bc5f2f9a-d8c9-4172-a033-3983f3dc0736 / ext4 defaults 0 0
UUID=512d5bf1-1411-4539-8162-be7e9daf1062 /home ext4 defaults,usrquota,grpquota 0 0
/dev/sdb1 /asist/disk1 ext4 defaults,auto,_netdev 0 0
/dev/sdb2 /asist/disk2 ntfs defaults,auto,_netdev 0 0
```

Para verificar que o mount foi bem sucedido executa-se o comando **“df -h”**

```
asist@uvm046:/$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            211M     0   211M   0% /dev
tmpfs           49M    2.7M   46M    6% /run
/dev/sda3       7.9G   3.6G   3.9G  48% /
tmpfs           241M     0   241M   0% /dev/shm
tmpfs           5.0M     0   5.0M   0% /run/lock
tmpfs           241M     0   241M   0% /sys/fs/cgroup
/dev/loop0      90M    90M     0 100% /snap/core/8039
/dev/loop1      90M    90M     0 100% /snap/core/7917
/dev/sda4        3.9G   17M   3.7G   1% /home
/dev/sdb1       2.3G   7.2M   2.2G   1% /asist/disk1
/dev/sdb2       2.0G   23M   2.0G   2% /asist/disk2
tmpfs           49M     0    49M   0% /run/user/1000
asist@uvm046:/$
```



Depois das configurações feitas, o estado de destino deve estar ligado:

DISCOS VIRTUAIS ISCSI

Todos os discos virtuais ISCSI | 1 no total

TAREFAS

Filtrar

Caminho

Estado

Estado do Disco Virtual

Nome do Destino

Estado do Destino

IP do Iniciador

Tamanho

▼ wvm046 (1)

P:\SCSVirtualDisk\ASIST.vhdx

Ligado

asist

Ligado

192.168.2019-11.pt.ipp.iemp.dei.udon46.asist

4,32 GiB

Última atualização em 01/12/2019 11:34:03

DESTINOS ISCSI

P:\SCSVirtualDisk\ASIST.vhdx em wvm046

TAREFAS

Filtrar

Nome

Nome do Servidor

IQN de Destino

Estado do Destino

IP do Iniciador

Último Início de Sessão

Tempo de Duração

asist

wvm046

iqn-2019-11.pt.ipp.iemp.dei.udon46.asist

Ligado

192.168.2019-11.pt.ipp.iemp.dei.udon46.asist

01/12/2019 11:16:26

00:00:00

Ativar o Windows

Ativar o Windows para usar o Microsoft

## Target Linux

A configuração do Target é necessário a instalação de packages, nomeadamente do package tgt, através do comando **"apt-get install tgt"**.

Também é necessária a criação duma partição que será a disponibilizada ao Initiator, tal como na configuração do Initiator foi utilizado o comando **"fdisk"** e os seus recursos. Criada uma partição primária **"/dev/sdb2"**, criamos nesta partição um filesystem do tipo NTFS, para que a partição em Windows possa ser formatada com NTFS também. De seguida, executamos o mount do disco numa pasta **"/asist/disk2"** através do comando **"mount /dev/sdb2 /asist/disk2"**. E adiciona-se também uma entrada para este disco no ficheiro **"/etc/fstab"** para que este seja montado automaticamente no arranque do servidor.

```
GNU nano 2.9.3 /etc/fstab
UUID=86f5331f-21cc-49b4-bd8d-27dc149b5cee none swap sw 0 0
UUID=bc5f2f9a-d8c9-4172-a033-3983f3dc0736 / ext4 defaults 0 0
UUID=512d5bf1-1411-4539-8162-be7e9daf1062 /home ext4 defaults,usrquota,grpquota 0 0
/dev/sdb1 /asist/disk1 ext4 defaults,auto,_netdev 0 0
/dev/sdb2 /asist/disk2 ntfs defaults,auto,_netdev 0 0
```

Configurado e montado o disco, cria-se uma imagem deste disco, imagem que vai ser disponibilizada ao Initiator para que este possa configurar e utilizar o disco, para este efeito utilizamos o comando **"dd if=/dev/sdb2 of=/asist/disk2/disk2.img count=0 bs=1 seek=2G"**.

```
asist@uvm046:/$ ls asist/disk2/
disk2.img
asist@uvm046:/$
```

De seguida, criamos um ficheiro de configuração na pasta **"/etc/tgt/conf.d"**, no nosso caso chamado **"iscsi.conf"**, para introduzir as informações do Target que vão ser utilizadas pelo sistema para o criar.

```
GNU nano 2.9.3 etc/tgt/conf.d/iscsi.conf
<target iqn.2019-11.pt.ipp.isep.dei.udom46:lun1>
  backing-store /asist/disk2/disk2.img
  initiator-address 10.9.11.46
  incominguser iqn.2019-11.pt.ipp.isep.dei.udom46:asistw1 asist2019gp46
  outgoinguser asist Asist2019Grupo46
</target>
```

Neste ficheiro configuramos o nome do Target, a imagem do disco a partilhar, o ip do initiator e as credenciais de autenticação. De seguida, executa-se o comando **"systemctl restart tgt"** para aplicar as definições escritas no ficheiro.

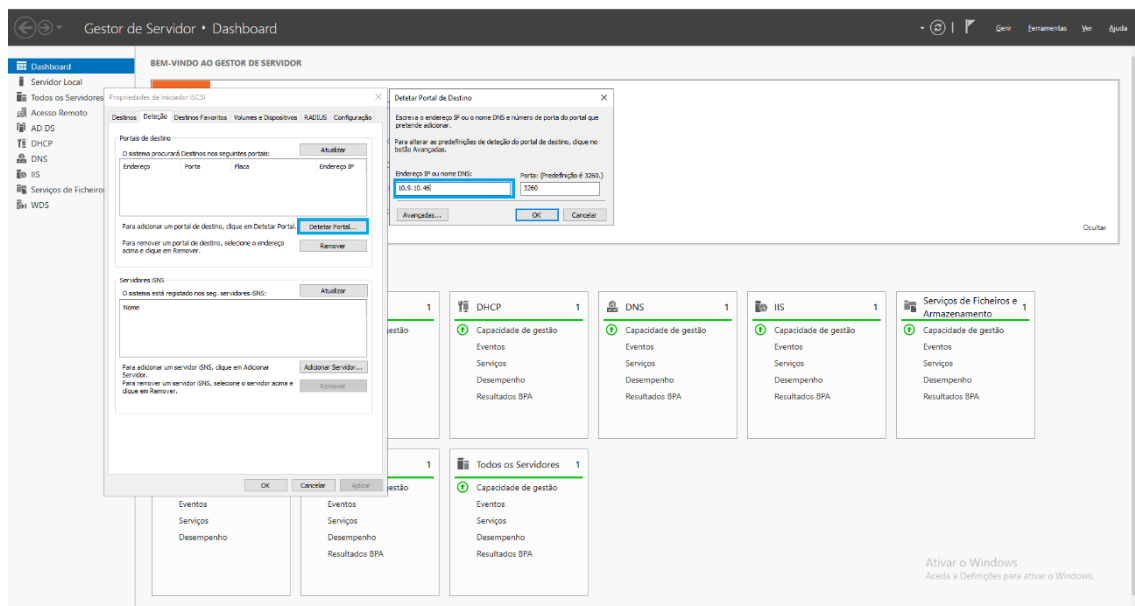
Para verificar o status do Target corre-se o comando “tgtadm --mode target --op show”

```
asist@uvm046:/$ sudo tgtadm --mode target --op show
Target 1: iqn.2019-11.pt.ipp.isep.dei.udom46:lun1
  System information:
    Driver: iscsi
    State: ready
  I_T nexus information:
    I_T nexus: 3
    Initiator: iqn.1991-05.com.microsoft:win-3n6ta0uk7re.wvdom046.dei.isep.ipp.pt alias: none
    Connection: 1
    IP Address: 10.9.11.46
  LUN information:
    LUN: 0
      Type: controller
      SCSI ID: IET      00010000
      SCSI SN: beaf10
      Size: 0 MB, Block size: 1
      Online: Yes
      Removable media: No
      Prevent removal: No
      Readonly: No
      SWP: No
      Thin-provisioning: No
      Backing store type: null
      Backing store path: None
      Backing store flags:
    LUN: 1
      Type: disk
      SCSI ID: IET      00010001
      SCSI SN: beaf11
      Size: 2147 MB, Block size: 512
      Online: Yes
      Removable media: No
      Prevent removal: No
      Readonly: No
      SWP: No
      Thin-provisioning: No
      Backing store type: rdwr
      Backing store path: /asist/disk2/disk2.img
      Backing store flags:
  Account information:
    iqn.2019-11.pt.ipp.isep.dei.udom46:asistw1
    asist (outgoing)
  ACL information:
    10.9.11.46
asist@uvm046:/$
```

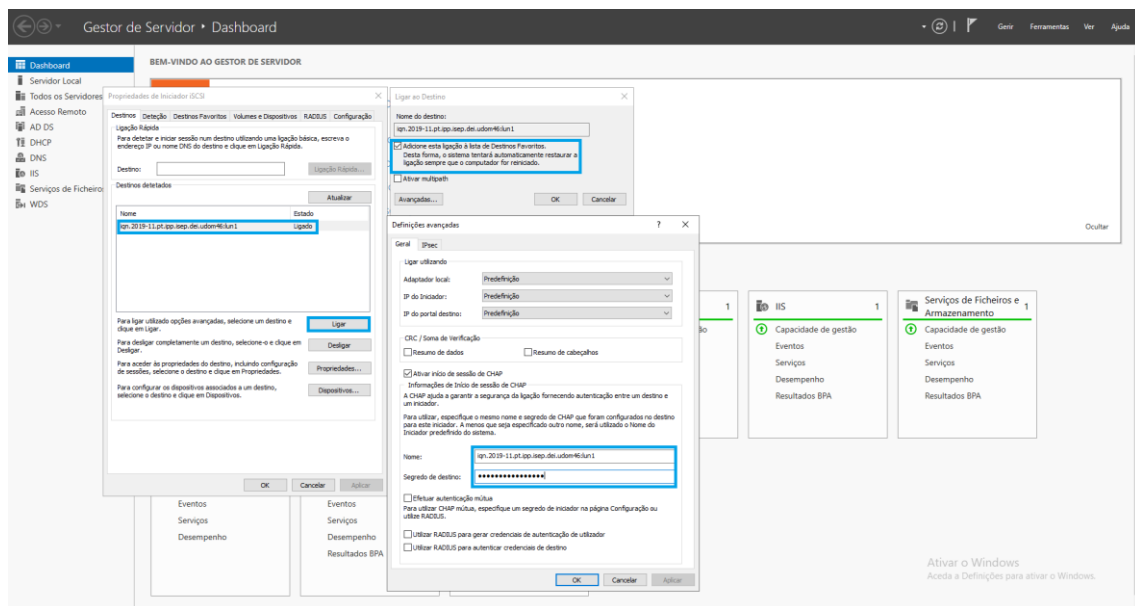
O Target Linux está agora configurado, falta apenas a configuração do Initiator Windows para poder conectá-lo a este Target.

## Initiator Windows

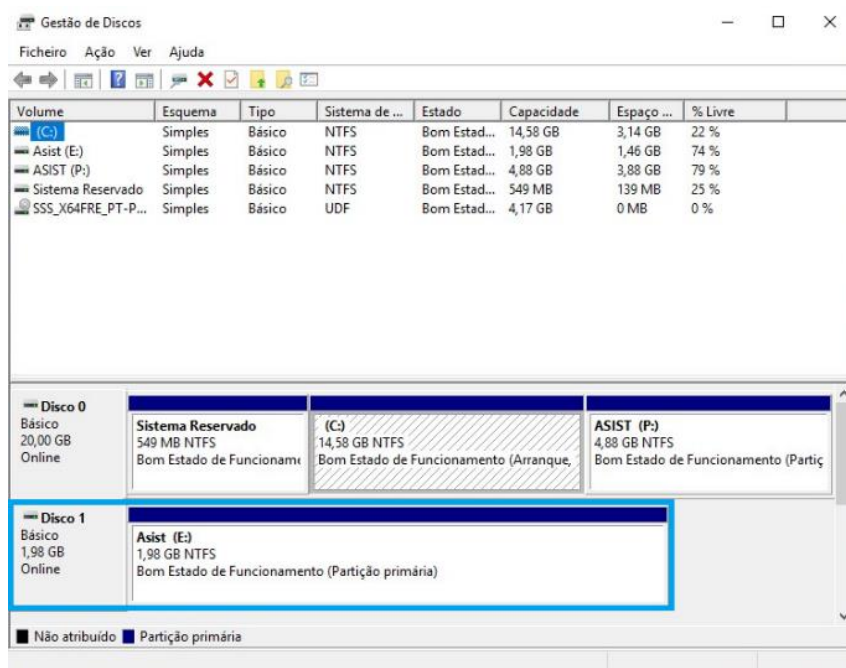
A configuração do Initiator e a partir do serviço Iniciador iSCSI. Com o serviço iniciado, começamos por indicar o ip do servidor Target



Ao estabelecer ligação com o servidor do Target, este deve ser detetado. De seguida estabelece-se uma conexão com ele, definindo as credenciais do CHAP de acordo com as definidas no target. Adiciona-se o Target nos “Destinos Favoritos” para que o Initiator se conecte automaticamente ao Target no arranque do servidor.



Concluída a ligação ao Target, vai aparecer automaticamente no gestor de discos do servidor Initiator, um novo disco não formatado com o tamanho especificado no Target, para concluir, basta colocar o disco online, inicializá-lo e criar um novo volume primário com formatação NTFS.



### Caso de uso 3

Para obter semanalmente os requisitos, inicialmente modificamos o ficheiro “crontab” com os seguintes parâmetros: o primeiro conjunto de parâmetros restringe uma data, sendo que esta quando for atingida ativa o script que leva à sua execução, o segundo parâmetro restringe o usuário e o terceiro representa o diretório onde se encontra o ficheiro a executar.

```
0 0 * * 0 root /etc/weeklyCheck.py
```

Este script não foi escrito em bash mas sim na linguagem python, devido ao seu alto nível e à existência de mais documentação.

```
def validateUserInShadowFile(user): #check if user was an entry in Shadow file
    with open("/etc/shadow") as file: #open file with path /etc/shadow
        for line in file: #loop that runs all lines until find user or file ends
            userName = line.split(':') #since the shadow file splits all params by :
            shadowUserName = userName[0]
            if user == shadowUserName:
                return True
    return False
```

Esta função serve a “main” de modo a verificar se o user recebido por parâmetro existe no ficheiro shadow, verificando se existe alguma ocorrência do user em alguma linha no primeiro valor, caso este existe a função devolve true, senão a resposta é false.

```
def validateGroupInGroupFile(group): #check if user was as entry in Group File
    with open("/etc/group") as file: #open file with path /etc/group
        for line in file: #loop that runs all lines until find user or file ends
            groupLine = line.split(':') #split since all values are all split by :
            groupUser = groupLine[0]
            if group == groupUser:
                return True
    return False
```

Esta função serve a “main” de modo parecido à função descrita a cima, varia no ficheiro sendo que este em vez do shadow é o de group.

```
def check_user_homedir(home, user, group, errorFile):
    path = '../..' + home
    if (os.path.exists(path)):
        line = subprocess.check_output(['ls', '-ld', path]).rstrip()
        lineSplit = line.split()
        owner = lineSplit[2]
        if not owner == user:
            errorFile.write('Home directory folder ({} belongs to wrong User {}.
            folderGroup = lineSplit[3]
            if not folderGroup == group:
                errorFile.write('Home directory folder ({} belongs to wrong group {}.
    else:
        errorFile.write('User {} doesn\'t have a home directory\n'.format(user))
```

Esta função verifica se o grupo primário, o diretório primário e a homedir existe e se o dono do grupo é o correto. Inicialmente cria o caminho para o diretório da home, verifica

se o caminho é válido, e de seguida faz as várias verificações conforme os parâmetros da função, verificando o dono e o grupo.

Esta função serve para criar o ficheiro de erros conforme o caminho do ficheiro, cria com permissões de escrita.

```
def createFile(filePath):  
    if not os.path.isfile(filePath):  
        open(filePath, 'w').close()  
    return filePath
```

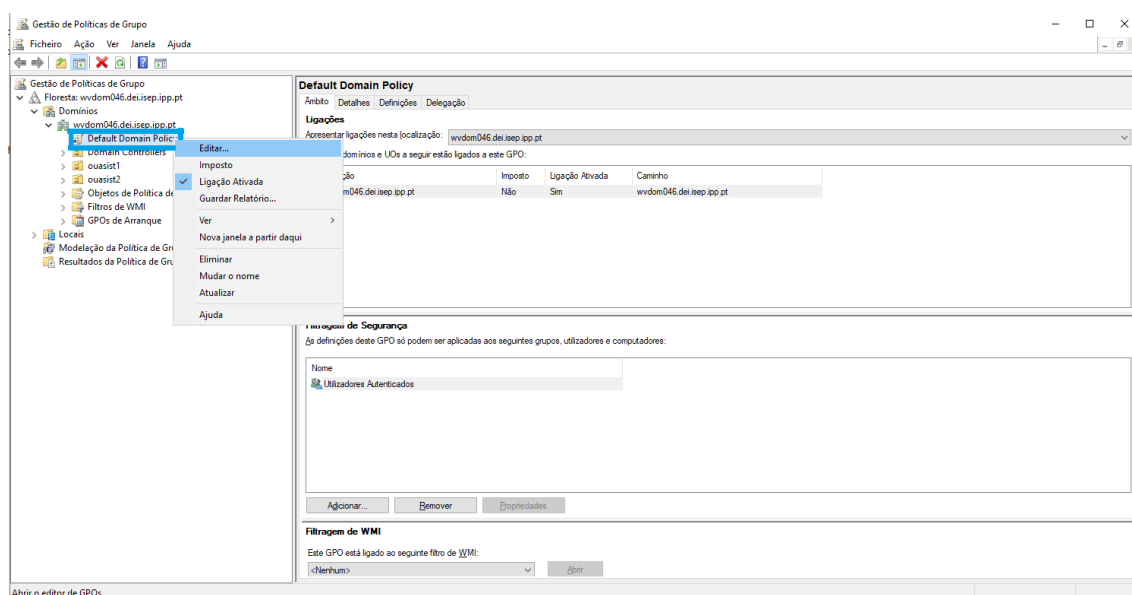
A “main” chama as várias funções descritas a cima, cria o ficheiro de erros com caminho “/tmp” e nome “auth\_error”.

Obtém o ficheiro “passwd”, percorre o ficheiro obtendo o username e fazendo a várias validações recorrendo as funções a cima descritas. Conforme o retorno true ou false, não adiciona ou adiciona ao ficheiro preparado para guardas os erros semanais.

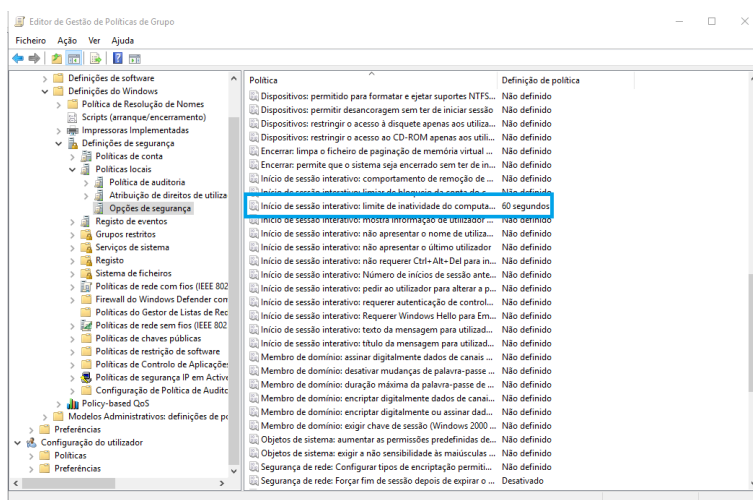
```
errorFilePath = createFile('../tmp/auth_error')  
errorFile = open(errorFilePath, 'r+')  
errorFile.write('Validations of user script runned at: {}\\n\\n'.format(datetime.now()))  
with open("/etc/passwd") as passwd:  
    for passwdLine in passwd:  
        userPasswdSplit = passwdLine.split(':')  
        user = userPasswdSplit[0]  
        if not validateUserInShadowFile(user):  
            errorFile.write('User {} doesnt have entry in /etc/shadow\\n'.format(user))  
        primaryGroup = subprocess.check_output(['id', '-gn', user]).rstrip()  
        if not validateGroupInGroupFile(primaryGroup):  
            errorFile.write('Primary Group {} of User {} doesn\\'t exist\\n'.format(primaryGroup, user))  
        home = userPasswdSplit[5]  
        check_user_homedir(home, user, primaryGroup, errorFile)  
errorFile.write('\\n')  
errorFile.close()
```

## Caso de uso 4

Para bloquear a sessão de todos os utilizadores registados no DC do Windows depois de 1 minuto de inatividade, primeiramente entrar no “Gestor do Servidor->Gestão de Políticas de Grupo” e editar o “Default Domain Policy” do domínio.



De seguida ir no “Configuracao do computador->Políticas->Definicoes do Windows->Definicoes de seguranca->Políticas locais->Opcoes de seguranca” e procurar pelo “Início de sessão interativo: limite de inatividade do computador” e definir o tempo para 60 segundos.



Para a alteracao anterior surtir efeito, a política deve ser atualizada atraves da linha de comando com o seguinte comando “gpupdate /force”.



## Webgrafia

1. <https://help.ubuntu.com/lts/serverguide/iscsi-initiator.html>
2. <https://www.tecmint.com/setup-iscsi-target-and-initiator-on-debian-9/>
3. [https://www.server-world.info/en/note?os=Ubuntu\\_18.04&p=iscsi&f=2](https://www.server-world.info/en/note?os=Ubuntu_18.04&p=iscsi&f=2)
4. [https://www.server-world.info/en/note?os=Ubuntu\\_18.04&p=iscsi&f=3](https://www.server-world.info/en/note?os=Ubuntu_18.04&p=iscsi&f=3)
5. <https://www.informaticar.net/how-to-setup-windows-server-2016-as-iscsi-target/>
6. <http://www.virtualizationblog.com/configuring-iscsi-target-in-windows-server-2016/>
7. [https://www.youtube.com/watch?v=sFmP6LzO5KE&fbclid=IwAR2Etm35nEALbGd8IkUgw\\_hHR7B3MHKZMmOe2lMZbNPs4frZIX0ajue9O9Y](https://www.youtube.com/watch?v=sFmP6LzO5KE&fbclid=IwAR2Etm35nEALbGd8IkUgw_hHR7B3MHKZMmOe2lMZbNPs4frZIX0ajue9O9Y)