Instituto Superior de **Engenharia** do Porto

Relatório Sprint 1&2

Turma 3DD - Grupo 23

1220879 – Rafael Brandao

Professor:

André Moreira, ASC

Unidade Curricular:

Arquitetura de Sistemas

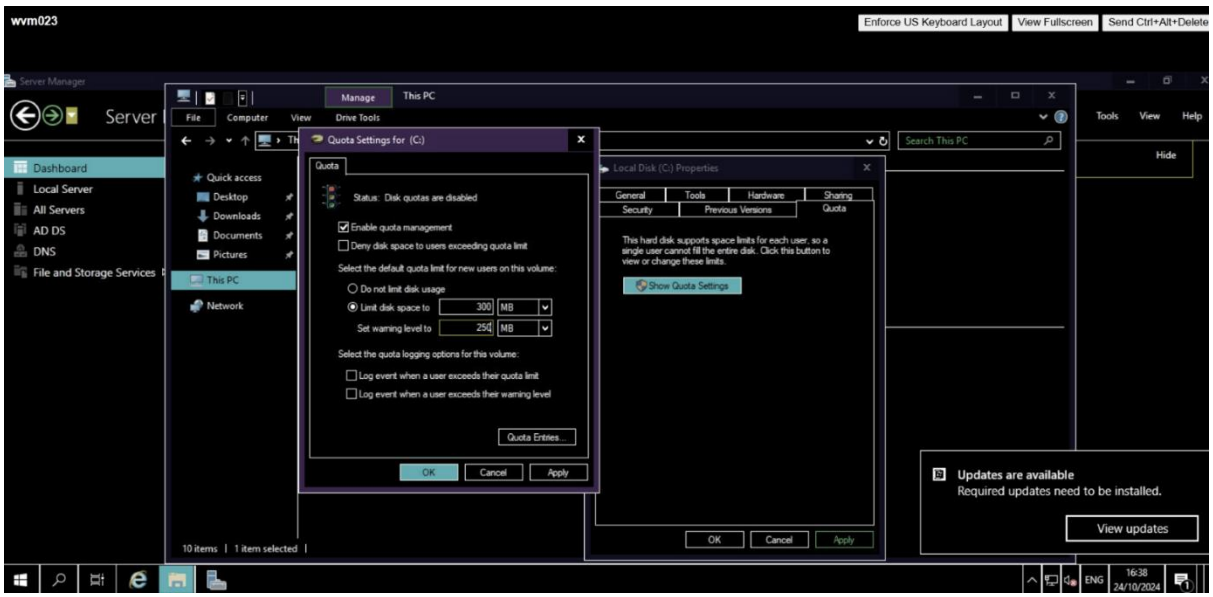
Índice

Sprint 1

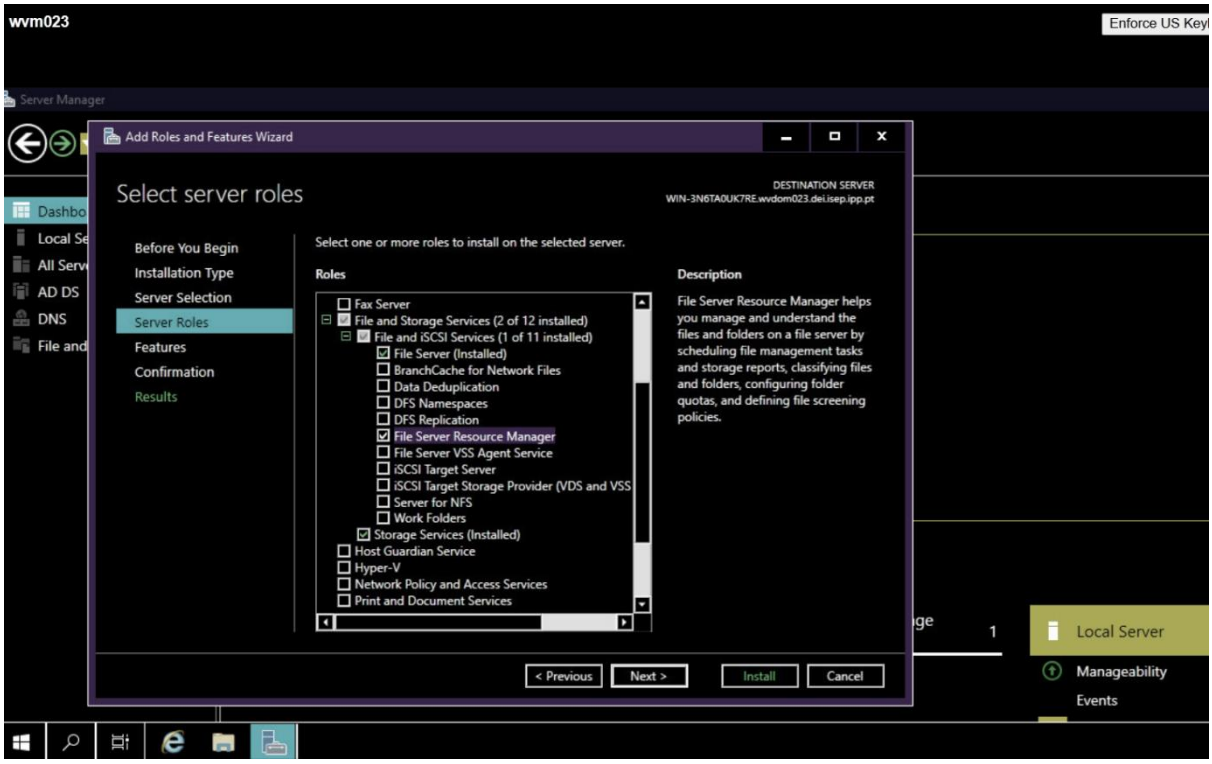
User Story 4

Requisitos: "*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.*"

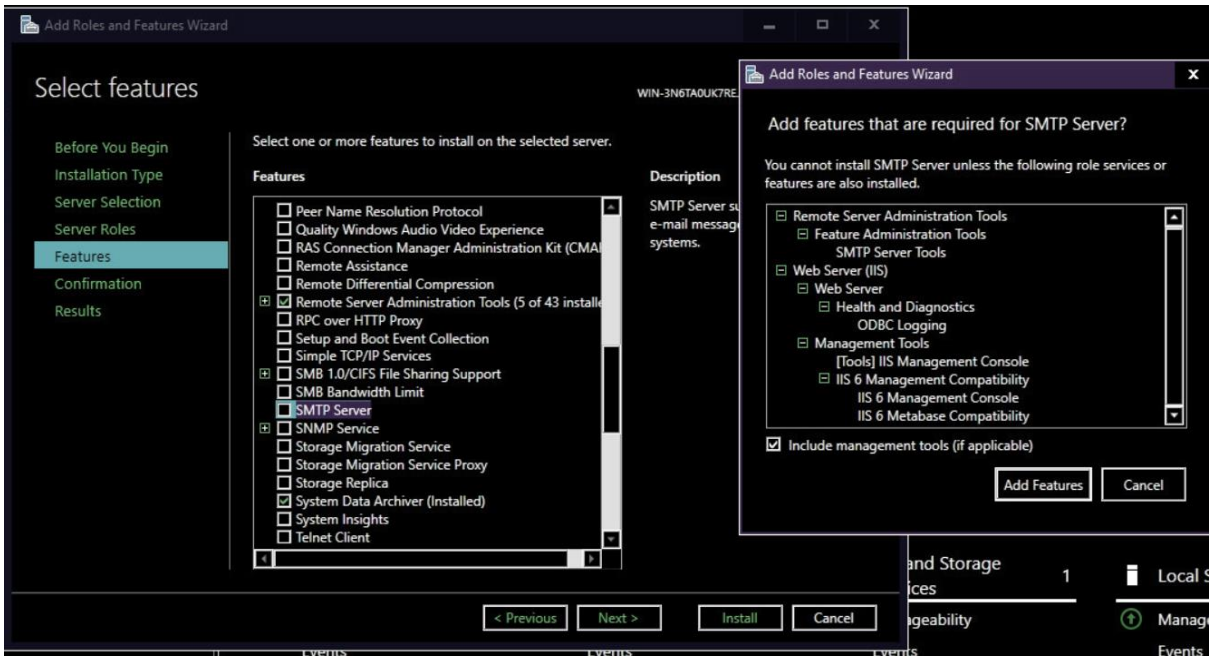
Inicialmente é necessário ativar o Quota Management no disco pretendido (C:):

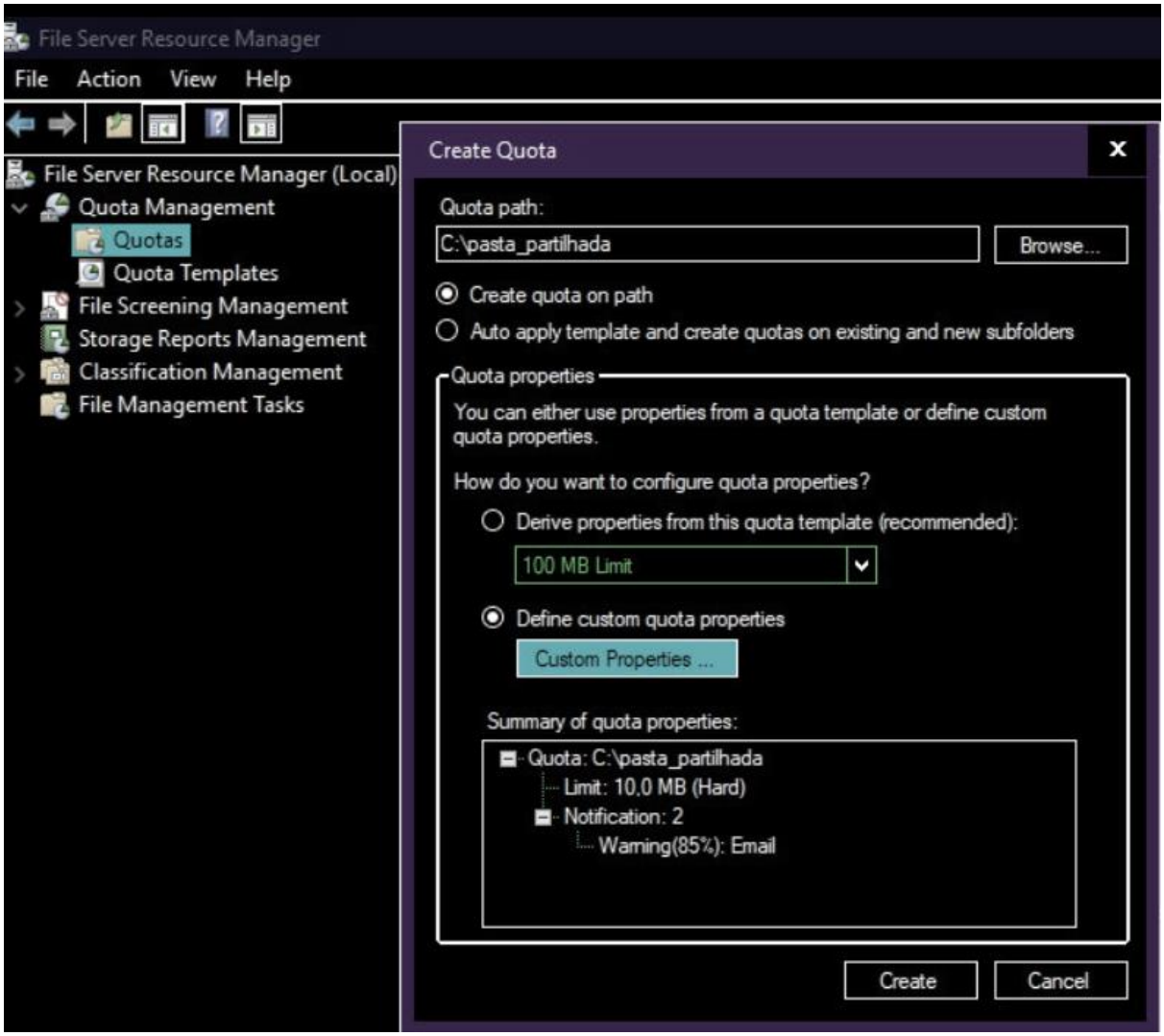


Após a criação da "pasta_partilhada" dentro do disco em que ativamos o Quota Management procedemos a adicionar o File Server Resource Manager para podermos também fazer a configuração das quotas na pasta que pretendemos usar.



Adicionamos também o SMTP Server, para o ADMIN ser avisado quando estiver perto de ultrapassar o limite definido de quota.





User Story 7

Requisitos: "*Como administrador do sistema quero 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.*"

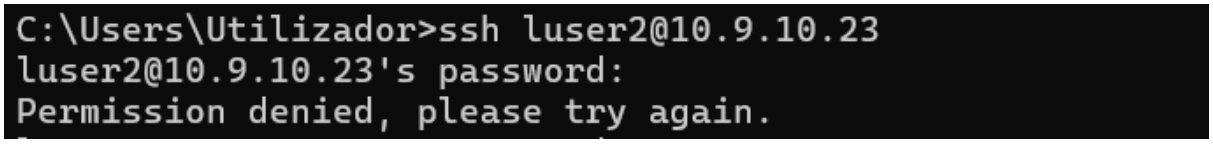
Primeiramente, é necessário criar um ficheiro /etc/bad-guys.

De seguida, listamos os users aos quais pretendemos negar o acesso e após isso configuraramos o arquivo PAM no ficheiro /etc/pam.d/sshd.

Parâmetros utilizados e os seus significados:

- onerr = succeed: No caso de surgir erros, como, por exemplo, se o arquivo '/etc/bad-guys' não existir, a autenticação será bem-sucedida.
- item=user: Especifica que o item a ser verificado é um user.
- sense=deny: Indica que o acesso deve ser negado caso o utilizador esteja na lista especificada.
- file=/etc/bad-guys: Este é o caminho para o ficheiro que contém a lista de utilizadores indesejados.

Assim, se um user incluído no ficheiro /etc/bad-guys tentar efetuar login, não irá conseguir aceder ao sistema após estas alterações.



Sprint 2

User Story 6

Requisitos: "*Como administrador do sistema quero que seja proposta, justificada e implementada uma estratégia de cópia de segurança que minimize o RPO (Recovery Point Objective) e o WRT (Work Recovery Time).*"

Estratégia e Justificação

1. Backup Total ao Domingo:
    - Garante a criação de um backup completo, o que facilita a restauração rápida de todo o sistema em caso de qualquer falha severa.
    - Realizamos o backup às 15:00h de modo a minimizar o impacto no sistema durante horas críticas.
2. Backups Incrementais Diários:
    - Registam-se apenas as alterações realizadas desde o último backup de modo a garantir que apenas essas sejam perdidas em situações de falha, reduzindo significativamente o RPO.
    - Reduz o tempo de recuperação (WRT) do backup diário pois complementam os dados restaurados sem necessidade de processamento intensivo.
3. Automação via Crontab:
    - Permite uma execução automática e sem intervenção humana, de modo a garantir alguma consistência no processo de backup.

Implementação

1. Criamos um script nomeado de *backup.sh* que faz o seguinte:
    - Backup Total: Exporta todas as tabelas do banco de dados para um arquivo SQL e compacta-o. Arquivos da semana anterior são movidos para a pasta "backups".
    - Backups Incrementais: Gera backups incrementais com base em alterações desde o último backup, utilizando as datas de modificação das tabelas. Também compacta os dados incrementais para minimizar espaço utilizado.

GNU nano 7.2 config/backup/backup.sh

```
#!/bin/bash

DB_HOST="vsgate-s1.dei.isep.ipp.pt"
DB_PORT="1234"
DB_NAME="sem5pi"
DB_USER="root"
DB_PASSWORD="trabalhasemuito2425"

BACKUP_DIR="/backup/database"
BACKUP_OLD="$BACKUP_DIR/backups"
LAST_BACKUP_TIME_FILE="$BACKUP_DIR/last_backup_time.txt"

DATE=$(date "+%Y-%m-%d_%H-%M-%S")

# Criar diretorios de backup, caso nao existam
mkdir -p "$BACKUP_DIR"
mkdir -p "$BACKUP_OLD"

# Verificar ultimo backup
if [ ! -f "$LAST_BACKUP_TIME_FILE" ]; then
  echo "1970-01-01 00:00:00" > "$LAST_BACKUP_TIME_FILE"
else
    LAST_BACKUP_TIME=$(cat "$LAST_BACKUP_TIME_FILE")
fi
```

```
# Backup total
if [ "$(date '+%u')" -eq 7 ]; then
  echo "A criar Backup Completo..."

  # Mover backups antigos
  mv "$BACKUP_DIR"/*.tar.gz "$BACKUP_OLD"

  # Realizar backup total
  mysqldump -h "$DB_HOST" --port="$DB_PORT" -u "$DB_USER" -p"$DB_PASSWORD" "$DB_NAME" > "$BACKUP_DIR/full_backup_$DATE.sql"

  # Compactar o backup
  tar -czf "$BACKUP_DIR/full_backup_$DATE.tar.gz" -C "$BACKUP_DIR" "full_backup_$DATE.sql"
  rm "$BACKUP_DIR/full_backup_$DATE.sql"

  # Atualizar o horario
  echo "$(date '+%Y-%m-%d %H:%M:%S')" > "$LAST_BACKUP_TIME_FILE"

  echo "Backup total concluido: $BACKUP_DIR/full_backup_$DATE.tar.gz"
else
  # Backup incremental
  echo "A criar Backup Incremental desde $LAST_BACKUP_TIME..."

  # Listar tabelas
  TABLES=$(mysql -h "$DB_HOST" -P "$DB_PORT" -u "$DB_USER" -p"$DB_PASSWORD" "$DB_NAME" -e "SHOW TABLES;" | tail -n +2)

  # Backup incremental para cada tabela modificada
  for TABLE in $TABLES; do
    QUERY="SELECT * FROM $TABLE WHERE updated_at >= '$LAST_BACKUP_TIME' OR created_at >= '$LAST_BACKUP_TIME';"
    mysql -h "$DB_HOST" -P "$DB_PORT" -u "$DB_USER" -p"$DB_PASSWORD" "$DB_NAME" -e "$QUERY" > "$BACKUP_DIR/incremental_$TABLE_$DATE.sql"
  done

  # Compactar os backups
  tar -czf "$BACKUP_DIR/incremental_backup_$DATE.tar.gz" -C "$BACKUP_DIR" "incremental_"*.sql
  rm "$BACKUP_DIR/incremental_"*.sql

  # Atualizar o horario
  echo "$(date '+%Y-%m-%d %H:%M:%S')" > "$LAST_BACKUP_TIME_FILE"

  echo "Backup incremental concluido: $BACKUP_DIR/incremental_backup_$DATE.tar.gz"
fi

echo "Backup concluido com sucesso"
```

Após a criação do ficheiro tornamo-lo executável com o comando:

```
QMPACHB8:~$ chmod +x /config/backup/backup.sh
```

2. De seguida, usamos o Crontab na automatização diária:

```
GNU nano 5.4                                /tmp/crontab.yPcsOK/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command


0 15 * * * /root/config/backup/backup.sh
```

User Story 8

Requisitos: *"Como administrador do sistema quero obter os utilizadores com mais do que 3 tentativas de acesso incorretas."*

De modo a obter os usuários com mais de 3 tentativas de acesso incorretas a partir dos *logs* de autenticação do sistema, armazenados em /var/log/auth.log, temos de implementar um Shell script, neste caso chamado de *tentativas_falhadas.sh.*

O script encontra o arquivo /var/log/auth.log e procede a filtrar as linhas com o texto "*Failed password*", que indicam tentativas de acesso incorretas.

```bash
#!/bin/bash

declare -A user_counts

while read -r line; do
    if [[ $line == *"Failed password"* ]]; then
        # User
        user=$(echo "$line" | awk '{print $9}')
        # Numero de tentativas falhadas
        ((user_counts["$user"]++))
    fi
done < /var/log/auth.log
```

De seguida, contando o número de falhas por usuário e mostra apenas os usuários com mais de 3 falhas de acesso.

```
echo "Users com mais de 3 tentativas de acesso incorretas:"
for user in "${!user_counts[@]}"; do
    count="${user_counts[$user]}"
    if (( count > 3 )); then
        printf "User: %-8s | Tentativas: %d\n" "$user" "$count"
    fi
done
```

Após tentarmos 3 vezes dar login com a conta luser1, se executarmos o script aparecerá por exemplo: