Sistemas Informáticos

Bloque 4
Práctica 3: Seguridad en cuentas de usuario y contraseñas en Linux con Docker

Instrucciones:

- Crea y accede a un contenedor Docker basado en Ubuntu que lleve tu nombre tanto el contenedor como el hostname de la máquina virtualizada.

docker run --hostname guillermoDominguez --name guillermoDominguez -it ubuntu:latest /bin/bash

Instalamos utilidades que necesitamos para la tarea y actualizamos paquetes:

apt-get update && apt-get install -y nano passwd libpam-pwquality

```
PS C:\Users\guill> docker run --hostname guillermoDominguez --name donGuillermo -it ubuntu:latest /bin/bash
root@guillermoDominguez:/# apt-get update && apt-get install -y nano passwd libpam-pwquality
```

Tareas:

- Crea dos usuarios nuevos: usuario1 y usuario2 (sin contraseña inicialmente).

useradd -m usuario1 && useradd -m usuario2

- Asigna una contraseña segura a cada usuario con el comando passwd.

echo "usuario1:Password123!" | chpasswd

echo "usuario2:Password123!" | chpasswd

- Configura la expiración de contraseñas: (comando chage)
    - Haz que la contraseña de usuario1 expire en 30 días.

chage -M 30 usuario1

    - Obliga a usuario2 a cambiar su contraseña en el próximo inicio de sesión.

chage -d 0 usuario2

    - Verifica la configuración de seguridad de cada usuario.

(Esto lo veremos al final del ejercicio al ejecutar el Script)

- Configura reglas estrictas de seguridad de contraseñas modificando el archivo /etc/login.defs para que:
    - Las contraseñas de los futuros usuarios expiren después de 90 días.

sed -i 's/^PASS_MAX_DAYS.*/PASS_MAX_DAYS\t90/' /etc/login.defs

    - El usuario deba esperar al menos 1 día antes de cambiar su contraseña nuevamente.

sed -i 's/^PASS_MIN_DAYS.*/PASS_MIN_DAYS\t1/' /etc/login.defs

    - Muestre una advertencia 7 días antes de que la contraseña expire.

sed -i 's/^PASS_WARN_AGE.*/PASS_WARN_AGE\t7/' /etc/login.defs

- Instala libpam-pwquality para reforzar la seguridad de contraseñas y Modifica las políticas de complejidad de contraseñas en /etc/security/pwquality.conf para que las contraseñas tengan estos requisitos mínimos:
    - Longitud mínima de 12 caracteres.
    - Requiere al menos un número.
    - Requiere al menos una mayúscula.
    - Requiere al menos una minúscula.
    - Requiere al menos un carácter especial.

echo -e "minlen = 12\ndcredit = -1\nucredit = -1\nlcredit = -1\nocredit = -1" >> /etc/security/pwquality.conf

```
root@guillermoDominguez:/# useradd -m usuario1 && useradd -m usuario2
root@guillermoDominguez:/# echo "usuario1:Password123!" | chpasswd
echo "usuario2:Password123!" | chpasswd
root@guillermoDominguez:/# chage -M 30 usuario1
root@guillermoDominguez:/# chage -d 0 usuario2
root@guillermoDominguez:/# sed -i 's/^PASS_MAX_DAYS.*/PASS_MAX_DAYS\t90/' /etc/login.defs
sed -i 's/^PASS_MIN_DAYS.*/PASS_MIN_DAYS\t1/' /etc/login.defs
sed -i 's/^PASS_WARN_AGE.*/PASS_WARN_AGE\t7/' /etc/login.defs
root@guillermoDominguez:/# echo -e "minlen = 12\ndcredit = -1\nucredit = -1\nlcredit = -1\nocredit = -1" >> /etc/security/pwquality.conf
root@guillermoDominguez:/# cat << 'EOF' > /correccion.sh
```

- Comprueba los requisitos tratando de realizar cambios de contraseña.

(Esto lo veremos al final del ejercicio al ejecutar el Script)

cat << 'EOF' > /correccion.sh

#!/bin/bash

echo "Realizando auditoría en tu sistema, por favor espere."

sleep 5

echo "El resultado del análisis es:"

for user in usuario1 usuario2; do

```
    if id "$user" &>/dev/null; then
        echo "✓ El usuario $user existe."
    else
        echo "ERROR: El usuario $user no existe."
    fi
done
if [[ $(chage -l usuario1 | grep "Maximum number of days between password change" | awk '{print $NF}') -eq 30 ]]; then
    echo "✓ La contraseña de usuario1 expira en 30 días."
else
    echo "ERROR: La expiración de la contraseña de usuario1 no es de 30 días."
fi
if [[ $(chage -l usuario2 | grep -i "password must be changed") ]]; then
    echo "✓ Usuario2 debe cambiar su contraseña en el próximo inicio de sesión."
else
    echo "ERROR: Usuario2 no tiene activada la obligación de cambio de contraseña."
fi
errors=0
if grep -Eq "^PASS_MAX_DAYS[[:space:]]+90" /etc/login.defs; then
    echo "✓ Las contraseñas expiran en 90 días."
else
        echo "ERROR: PASS_MAX_DAYS no está configurado en 90 días."; errors=$((errors+1))
fi
if grep -Eq "^PASS_MIN_DAYS[[:space:]]+1" /etc/login.defs; then
    echo "✓ Se requiere un mínimo de 1 día antes de cambiar contraseña."
else
```

```
        echo "ERROR: PASS_MIN_DAYS no está configurado en 1 día."; errors=$((errors+1))

fi

if grep -Eq "^PASS_WARN_AGE[[:space:]]+7" /etc/login.defs; then

   echo "✓ Se muestra advertencia 7 días antes del vencimiento de la contraseña."

else

        echo "ERROR: PASS_WARN_AGE no está configurado en 7 días."; errors=$((errors+1))

fi

dpkg -l | grep -q libpam-pwquality

if [[ $? -eq 0 ]]; then

   echo "✓ libpam-pwquality está instalado."

else

   echo "ERROR: libpam-pwquality no está instalado."

fi

pwq_config_file="/etc/security/pwquality.conf"

if grep -q "minlen = 12" "$pwq_config_file" && \

  grep -q "dcredit = -1" "$pwq_config_file" && \

  grep -q "ucredit = -1" "$pwq_config_file" && \

  grep -q "lcredit = -1" "$pwq_config_file" && \

  grep -q "ocredit = -1" "$pwq_config_file"; then

        echo "✓ Las reglas de complejidad de contraseña están configuradas correctamente."

else

      echo "ERROR: No todas las reglas de complejidad de contraseña están configuradas correctamente."

fi

echo "Revisión completada."
```

EOF

```
> #!/bin/bash
echo "Realizando auditor\303\255a en tu sistema, por favor espere."
sleep 5
echo "El resultado del an\303\241lisis es:"
for user in usuario1 usuario2; do
    if id "$user" &>/dev/null; then
        echo "\342\234\223 El usuario $user existe."
    else
        echo "ERROR: El usuario $user no existe."
    fi
done
if [[ $(chage -l usuario1 | grep "Maximum number of days between password change" | awk '{print $NF}') -eq 30 ]]; then
    echo "\342\234\223 La contrase\303\261a de usuario1 expira en 30 d\303\255as."
else
    echo "ERROR: La expiraci\303\263n de la contrase\303\261a de usuario1 no es de 30 d\303\255as."
fi
if [[ $(chage -l usuario2 | grep -i "password must be changed") ]]; then
    echo "\342\234\223 Usuario2 debe cambiar su contrase\303\261a en el pr\303\263ximo inicio de sesi\303\263n."
else
    echo "ERROR: Usuario2 no tiene activada la obligaci\303\263n de cambio de contrase\303\261a."
fi
errors=0
if grep -Eq "^PASS_MAX_DAYS[[:space:]]+90" /etc/login.defs; then
    echo "\342\234\223 Las contrase\303\261as expiran en 90 d\303\255as."
else
    echo "ERROR: PASS_MAX_DAYS no est\303\241 configurado en 90 d\303\255as."; errors=$((errors+1))
fi
if grep -Eq "^PASS_MIN_DAYS[[:space:]]+1" /etc/login.defs; then
    echo "\342\234\223 Se requiere un m\303\255nimo de 1 d\303\255a antes de cambiar contrase\303\261a."
else
    echo "ERROR: PASS_MIN_DAYS no est\303\241 configurado en 1 d\303\255a."; errors=$((errors+1))
fi
if grep -Eq "^PASS_WARN_AGE[[:space:]]+7" /etc/login.defs; then
    echo "\342\234\223 Se muestra advertencia 7 d\303\255as antes del vencimiento de la contrase\303\261a."
else
    echo "ERROR: PASS_WARN_AGE no est\303\241 configurado en 7 d\303\255as."; errors=$((errors+1))
fi
dpkg -l | grep -q libpam-pwquality
if [[ $? -eq 0 ]]; then
    echo "\342\234\223 libpam-pwquality est\303\241 instalado."
else

EOFo "Revisi\303\263n completada."as de complejidad de contrase\303\261a est\303\241n configuradas correctamente."
```

Usamos ***chmod +x /correccion.sh*** para cambiar los permisos, y ***bash /correccion.sh***.

El resultado del script:

```
root@guillermoDominguez:/# chmod +x /correccion.sh
root@guillermoDominguez:/# bash /correccion.sh
Realizando auditoría en tu sistema, por favor espere.
El resultado del análisis es:
✓ El usuario usuario1 existe.
✓ El usuario usuario2 existe.
✓ La contraseña de usuario1 expira en 30 días.
✓ Usuario2 debe cambiar su contraseña en el próximo inicio de sesión.
✓ Las contraseñas expiran en 90 días.
✓ Se requiere un mínimo de 1 día antes de cambiar contraseña.
✓ Se muestra advertencia 7 días antes del vencimiento de la contraseña.
✓ libpam-pwquality está instalado.
✓ Las reglas de complejidad de contraseña están configuradas correctamente.
Revisión completada.
```

Varios intentos aparte de cambiar la contraseña sin cumplir los requisitos mínimos:

```
BAD PASSWORD: The password contains less than 1 digits
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password contains less than 1 uppercase letters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password contains less than 1 lowercase letters
Retype new password:
Sorry, passwords do not match.
passwd: Have exhausted maximum number of retries for service
passwd: password unchanged
```

Copiamos el archivo del contenedor a la carpeta personal:

docker cp donGuillermo:/correccion.sh C:\Users\guill\Desktop\1DAM

```
PS C:\Users\guill>  docker cp donGuillermo:/correccion.sh C:\Users\guill\Desktop\1DAM\
Successfully copied 4.1kB to C:\Users\guill\Desktop\1DAM\
```