# Pràctica 2d: Treballant amb contrasenyes d'accés segures i doble autenticació

## 1- Objectius de la pràctica

Aquesta activitat té com objectius:

- La creació dúsuaris pels quals es crearan contrasenyes d'accés segures.
- S'incrementarà la seguretat afegint una segona capa d'autenticació (donle autenticació) utilitzant una aplicació de tipus 2FA App.
- Es provarà el funcionament del doble sistema d'autenticació accedint a un servidor SSH correctament configurat i una aplicació instal·lada al mòbil.

Un altre objectiu d'aquesta activitat es treballar els següents resultats d'aprenentatge i continguts marcats al currículum de la UF al DOGC:

- RA1 i C1 →
  - RA1.7 --> Aplica tècniques criptogràfiques en l'emmagatzematge i la transmissió de la informació.
  - C1.6 → Seguretat Lògica

Per desenvolupar aquests objectius i avaluar si s'han adquirit correctament els conneixemens que es demanen:

- L'alumne haurà de llegir i entendre la documentació. Per resoldre qualsevol dubte sobre la documentació, es durà a terme una sessió a classe per explicar els conceptes més importants i resoldre els dubtes que hagin aparegut durant la lectura de la documentació.
- L'alumne haurà de lliurar una pràctica funcionant d'acord a allò que es demani a l'enunciat.

## 2- Generació de passwords segurs

1. Un password segur hauria de tenir les següents característiques mínimes:
   - Tenir una longitud mínima de 8 caràcters
   - Incloure com a mínim una minúscula
   - Incloure com a mínim una majúscula
   - Incloure com a mínimun número
   - Incloure com a mínim un caràcter especial

2. La distribució Debian del sistema operatiu GNU/Linux permet generar passwords utilitzant l'aplicació **pwgen**. Aquesta ordre permet indicar la quantitat de caràcters, i els tipus de caràcters que volem incloure dins de la contrasenya.

## 3- Sistema de doble autenticació amb 2FA App (2 Factor Authentication App)

1. Un sistema autenticació doble afegeix una capa de seguretat extra quan es vol accedir al sistema perquè demana autenticar a l'usuari seguint 2 passos:
   - Primer, s'ha d'introduir el nom d'usuari i contrasenya
   - Després s'haura de validar l'usuari introduint un codi que es reberà al mòbil via **SMS** o una aplicació específica per rebre aquest codi que s'hauria d'instal·lar dins del mòbil. Aquest tipus d'aplicacions reben el nom de **2FA App** (2 Facto Authentication App).

2. La doble autenticació via SMS, tot i que utilitzada pels bancs, té algunes vulnerabilitats de manera que serà millor utilitzar una 2FA App com les proporcionades per exemple per **Authy** o per **Google Authenticator**.

3. **Google Authenticator** és el més popular dels sistemes de doble autenticació, és gràtis, força fàcil d'instal·lar i utilitzar per qualsevol usuari, està disponible per ordinadors treballant amb sistemes operatius **Linux/Windows/Apple** i per mòbils o tauletes treballant amb **Android/iOS**. Avui dia molts servidors webs, SSH, etc..., que utilitzen aquest sistema per assegurar la doble autenticació.

**4.** L'usuari només necessita instal·lar una aplicació dins del mòbil i tenir un lector de codi QR per poder treballar amb aquest sistema. Per l'administrador del sistema al qual s'ha d'accedir només s'ha d'instal·lar el programari i seguir uns senzills passos de configuració.

## 3- Creació d'un usuari amb una contrasenya que utilitzi els criteris bàsics de segurat

**a)** Instal·la el programa **pwgen**. Executa:

```
sudo aptitude install pwgen
```

**b)** Crea una contrasenya de 8 caràcters, que tingui com a mínim una minúscula, una majúscula, un número i un caràcter especial. Executa:

```
pwgen -y -c -n 8 1
```

**c)** Recorda la contrasenya generada a l'apartat anterior. Crea un usuari del sistema de nom **inf1**. Executa:

```
sudo adduser inf1
```

Quan et demani la contrasenya escriu la generada a l'apartat anterior. Per la resta de preguntes, prem <Enter> per acceptar l'opció per defecte.

**d)** Verifica que s'ha creat el nou usuari. Executa:

```
cat  /etc/passwd | grep inf1
```

i comprova que el resultat és similar a aquest:

```
inf1:1001:1001:,,,:/home/inf1:/bin/bash
```

## 4- Configuració del servei SSH i de Google Authenticator

### 4.1- Comprova que el servei SSH està instal·lat i funcionant

**a)** Comprova que el servei SSH està instal·lat. Executa:

```
aptitude search openssh-server
```

i comprova que es mostra el següent resultat:

```
i  A  openssh-server                - secure shell (SSH) server, for secure access from remote machines
```

**b)** Comprova que el servei SSH està activat, funcionant i en execució. Executa

```
systemctl statis ssh | grep "Active"
```

i comprova que es mostra un resultat similar a aquest (amb data, hora i durada d'acord al moment de l'execució de la instrucció):

```
Active: active (running) since Fri 2021-12-17 12:55:38 CET; 1h 16min ago
```

4.2- Instal·la i configura el servidor SSH per treballar amb Google Authenticator

**a)** Instal·la el programari per autenticar usuaris via **Google Authenticator** sobre Linux. Executa:

**sudo aptitude install libpam-google-authenticator**

**b)** Per obligar al servidor SSH a treballar amb Google Authenticator afegeix les següents línie al final del fitxer **/etc/pam.d/sshd**:

```
#
# Adding 2FA  via Google authenticator
auth required pam_google_authenticator.so
#
```

**NOTA:** Haurà d'obrir el fitxer amb l'ordre **nano** i treballant amb **sudo**.

**c)** Modifica el paràmetre **ChallengeResponseAuthentication** del fitxer de configuració del servei SSH **/etc/ssh/sshd_config**. Fes que el seu valor sigui **yes**. Hàuràs de:

- Executar **sudo nano /etc/ssh/sshd_config**
- Buscar el paràmetre **ChallengeResponseAuthentication** que està a la línia 63 de fitxer i canviar **no** per un **yes**.
- Desar i sortir.

**d)** Reinicia el servei SSH. Executa:

**sudo systemctl restart sshd**

4.3- Configura Google Authenticator

**a)** Des d'un terminal i des del directori personal del teu usuari de sistema, executa el programa de configuració de **Google Authenticator**:

**google-authenticator**

i respon

**Do you want authenticaction token to be time based (y/n) y**

**b)** Instal·la l'aplicació Google Authenticator des de Play Store o Apple Store.

**c)** Executa l'app, escaneja el codi QR i apunta el codi que es genera. Compte que el codi té una durada màxima.

**d)** Entra el codi generat des de l'app de Google Authenticator al mòbil dins de
Enter code from app (-1 to skip):

**e)** Copia els codis d'emergència que es generen en un lloc segur. Seran utils si no pots accedir al mòbil per generar codis temporals d'accés.

**f)** Respon **y** a la resta de preguntes

**5- Proves d'accés**

**a)** Comprova l'adreça IP de la màquina amb el servidor SSH. Executa:

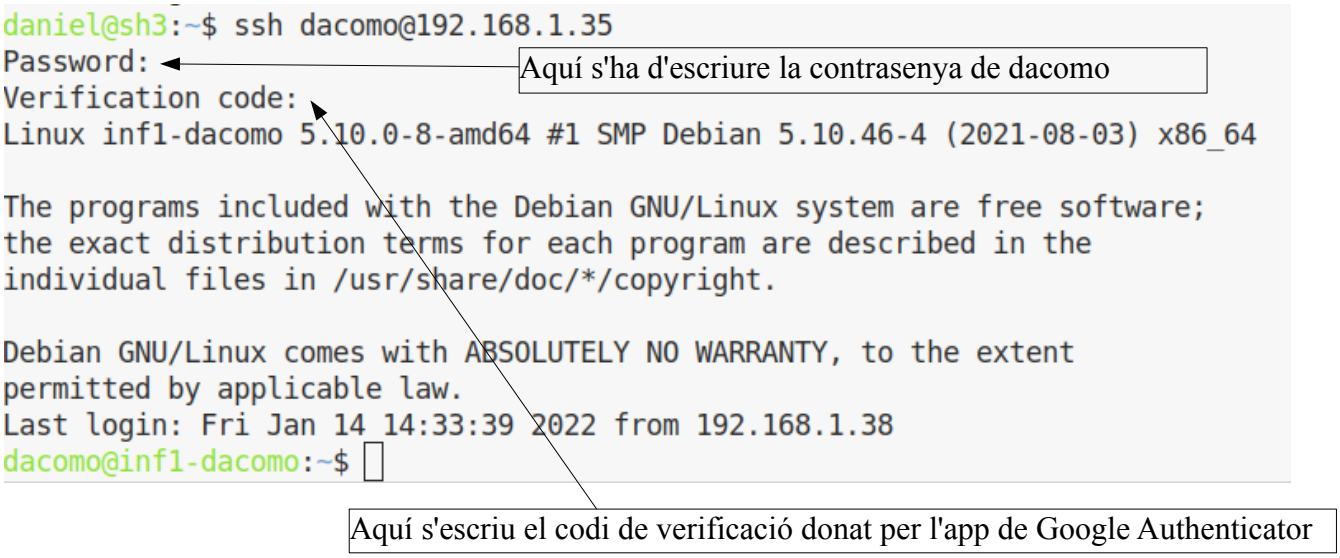
```
ip -4 -br add show dev enp0s3
```

**b)** Comprova des d'una altra màquina pots fer un **ping** a la màquina amb el servei SSH.

**c)** Des d'una altra màquina accedeix a la teva màquina via SSH amb el teu compte d'usuari i comprova que et demana la contrasenya i després el codi generat des l'app Google Authenticator del mòbil.

Per accedir per tant, executa: **ssh nom_usuari@adreça_ip** a on nom_usuari és el nom del teu usuari a la màquina amb el servidor SSH, i adreça_ip és l'adreça IP de a màquina amb el servidor SSH.

**d)** Aquí poso una captura d'exemple amb la meva màquina virtual:




**5- Documentació**

**1-** Instal·lació i configuració d'un sistema de doble autenticació amb Google Authenticator pel servei SSH: https://ubuntu.com/tutorials/configure-ssh-2fa#1-overview

**Forma de lliurament de la pràctica**

**1-** Activitat en grups de 2 persones o individual.

**2-** Demostració:
**a)** Demanaré a cadda membre del grup que es validi dins del servidor SSH del company utilitzant doble autenticació.

**3-** Dates de lliurament:
* Setmana del **17-1-22 al 21-1-22** dins de l'horari escolar per aconseguir el **100%** de la nota.
* Setmana del **24-1-22 al 28-1-22** dins de l'horari escolar per aconseguir el **100%** de la nota.
* Setmana del **31-1-22 al 4-2-22** dins de l'horari escolar per aconseguir el **70%** de la nota.
* Després del **31-01-22** no s'accepta la pràctica