INSTRUCCIONES:

Antes de comenzar el ejercicio, crea un directorio llamado *<EjercicioCrypto>*. Dentro del fichero crea un nuevo fichero llamado *<ejercicio_crypto.txt>* y le incorporaras el siguiente texto:

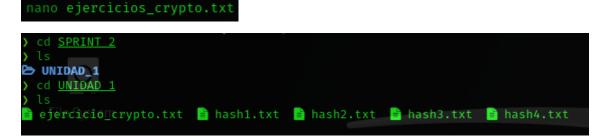
b2pvlGltcG9ydGFudGUgcXVllGNvZGImaWNhciBubyBlcyBsbyBtaXNtbyBxdWUgY2ImcmFy

Para entregar el ejercicio realiza un informe con las capturas de pantalla necesarias para mostrar la ejecución de los ejercicios, mostrando cada enunciado con su imagen correspondiente En esta actividad tendrás que usar tu Kali.

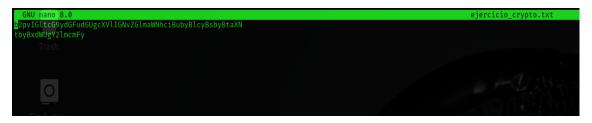
DESCRIPCIÓN Y RESOLUCIÓN:

-- EJERCICIO 1 – FUNCIONES HASH:

- PREVIO:
 - Creo el archivo <ejercicio_crypto> dentro del directorio <EjercicioCrypto>:



Abro el archivo con nano y le copio el texto.



1.- Crea un hash MD5 del fichero ejercicio_crypto.txt y guárdalo como hash1.txt



2.- Crea un hash SHA-1 del fichero *ejercicio_crypto.txt* y guárdalo como *hash2.txt*.



3.- Crea un hash SHA-256 del fichero *ejercicio_crypto.txt* y guárdalo como *hash3.txt*.



4.- Crea un hash SHA-512 del fichero *ejercicio_crypto.txt* y guárdalo como hash4.tx.



- 5.- Comprueba con la función "hash-identifier" el resultado de cada uno de los ficheros hash obtenidos anteriormente. ¿Acierta en la predicción del tipo de hash?
 - Realizo un script para que automatice la solución, pero no me funciona con la función hash – identifier pero si con hashid:

```
#!/bin/bash

# Directorio donde se encuentran los archivos de texto
DIR_HASHES="/home/vicevil/BOOT_CIBER_2024/SPRINT_2/UNIDAD_1"

# Iterar sobre cada archivo de texto en el directorio
for archivo in "$DIR_HASHES"/*.txt; do
echo "Procesando archivo: $archivo"

# Leer cada linea del archivo y pasarla a hashid
while read -r linea; do
# Extraer solo el hash de cada linea y descartar el texto final
hash=$(echo "$linea" | awk '{print $1}')
echo "Identificando hash: $hash"
# Llamar a hashid con el hash
hashid "$hash"
done < "$archivo"
```

Dando como resultado con el script:

```
percicio_crypto.txt hash1.txt hash2.txt hash3.txt hash4.txt⊡identifier_hashes.sh
)./identifier_hashes.sh
Procesandgn_archivo: /home/vicevil/BOOT_CIBER_2024/SPRINT_2/UNIDAD_1/ejercicio_crypto.txt
Identificando hash: b2pvIGltcG9ydGFudGUgcXVIIGNvZGlmaWNhciBubyBlcyBsbyBtaXNtbyBxdWUgY2lmcmFy
Analyzing 'b2pvIGltcG9ydGFudGUgcXVIIGNvZGlmaWNhciBubyBlcyBsbyBtaXNtbyBxdWUgY2lmcmFy'
[+] Unknown hash
Procesando archivo: /home/vicevil/BOOT_CIBER_2024/SPRINT_2/UNIDAD_1/hash1.txt
Identificando hash: 3244441dda3489d1fa5cdb1cf72342dd
Analyzing "3244441dda3489d1fa5cdb1cf72342dd'
[+] MD2_System
[+] MD2 System
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger_128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] NTLM
[+] Domain Cached Credentials 2
[+] RAdmin v2.x
```

```
Procesando archivo: /home/vicevil/BOOT_CIBER_2024/SPRINT_2/UNIDAD_1/hash2.txt
Identificando hash: ddcf5b509a5a1677fc214ae2bb1795983a55fd4d
Analyzing 'ddcf5b509a5a1677fc214ae2bb1795983a55fd4d'
[#] SHA-1
[#] Duble SHA-1
[#] RIPEMD-160
[#] Haval-160
[#] Haval-160
[#] Haval-160
[#] Jieer-160
[#] Kiper-160
[#] Skein-512(160)
Procesando archivo: /home/vicevil/BOOT_CIBER_2024/SPRINT_2/UNIDAD_1/hash3.txt
Identificando hash: 9036a529c6880e1095e412a870426e51423abf670a36ae72e4550dd6791c4cca
Analyzing '0936a529c6880e1095e412a870426e51423abf670a36ae72e4550dd6791c4cca'
[#] Snefru-256
[#] SHA-256
[#] SHA-256
[#] GOST R 34.11-94
[#] GOST CryptoPro S-Box
[#] Haval-256
[#] GOST R 34.11-94
[#] GOST CryptoPro S-Box
[#] Skein-256
[#] Skein-250
[#] Skein-256
[#] Skein-2
```

No obstante, como el ejercicio solicita hacerlo con hash – identifier, procedo a realizarlo de forma manual:

1.-



2.-

```
HASH: 3244441dda3489d1fa5cdb1cf72342dd

Possible Hashs:

+] MD5

+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashs:

+] RAdmin v2.x

+] NTLM

+] MD4

+] MD2

+] MD5(HMAC)

+] MD5(HMAC)

+] MD4(HMAC)

+] MD2(HMAC)

+] MD5(HMAC)
```

3.-

```
HASH: ddcf5b509a5a1677fc214ae2bb1795983a55fd4d

Possible Hashs:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))

Least Possible Hashs:
[+] Tiger-160
[+] Haval-160
[+] RipeMD-160
[+] SHA-1(HMAC)
[+] Tiger-160(HMAC)
[+] RipeMD-160(HMAC)
[+] Haval-160(HMAC)
[+] SHA-1(MaNGOS)
[+] SHA-1(MaNGOS2)
```

4.-

```
HASH: 9036a529c6880e1095e412a870426e51423abf670a36ae72e4550dd6791c4cca

Possible Hashs:
[+] SHA-256
[+] Haval-256

Least Possible Hashs:
[+] GOST R 34.11-94
[+] RipeMD-256
[+] SNEFRU-256
[+] SHA-256(HMAC)
[+] Haval-256(HMAC)
[+] Haval-256(HMAC)
[+] SNEFRU-256(HMAC)
[+] SNEFRU-256(HMAC)
[+] SHA-256(md5($pass))
[+] SHA-256(sha1($pass))
```

5.-

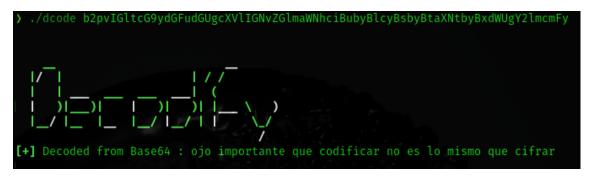
```
HASH: 6a7267d9ead4f41d8bf13e4ec2fbf86efd2418409526523d221c3dcc701c236c75370a724afd70d900febd8e
3e4ff0e1af8be9fe9f5

Possible Hashs:
[+] SHA-512
[+] Whirlpool

Least Possible Hashs:
[+] SHA-512(HMAC)
[+] Whirlpool(HMAC)
```

-- EJERCICIO 2 - CODIFICACIÓN:

1.- Decodifica la cadena de texto del fichero *ejercicio_crypto.txt.* ¿En qué codificación se encontraba? En base64.



2. Codifica en el mismo formato la cadena de texto "No metemos gente en criptas".

Esto se puede directamente en Bash con el comando base64(codificar y decodificar):

o hacerlo a través de la web: https://www.base64encode.org/:

