Esto es una recreación de un posible escenario real

El G.E.O. de la Policía Nacional ha realizado una intervención contra un piso franco donde ha sido detenido un individuo. el cual se tenían pruebas de un posible ataque terrorista que iba a realizar. Se ha intervenido un ordenador que estaba encendido al cual se le ha realizado un TRIAGE y extraído la memoria volátil. Una vez extraída la memoria volátil, se ha realizado una clonación del disco duro del dispositivo. También han requisado un USB que puede contener información delicada. Tú objetivo es realizar un peritaje informático donde vas a tener que realizar un estadio completo de todas estas evidencias con la posterior redacción de su informe.

Evidencias:

- Captura de RAM
- Clonado del Disco Duro
- Clonado del USB

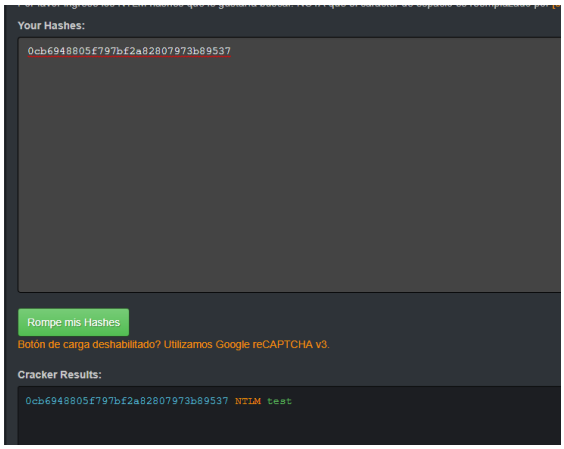
Todas las evidencias se podrán descargar a través del siguiente enlace: https://drive.google.com/open?id=1ccxxRSShqvtjS7sl3xSrV0oLZaDmffD1

RESOLUCIÓN DE LAS PRINCIPALES EVIDENCIAS

Se va a comenzar realizando un análisis a la memoria RAM. Para ello se va a utilizar la herramienta Volatility. Por ejemplo, se puede comprobar si los hashes NTLM de las contraseñas del usuario se encuentran cargadas en memoria. Para ello vamos a utilizar el plugin de hashdump.

```
root@kali:/media/seguridad# volatility -f MemoryDumpWH.raw --profile=Win7SP0x86 hashdump
Volatility Foundation Volatility Framework 2.6
Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
PC-0048:1000:aad3b435b51404eeaad3b435b51404ee:0cb6948805f797bf2a82807973b89537:::
```

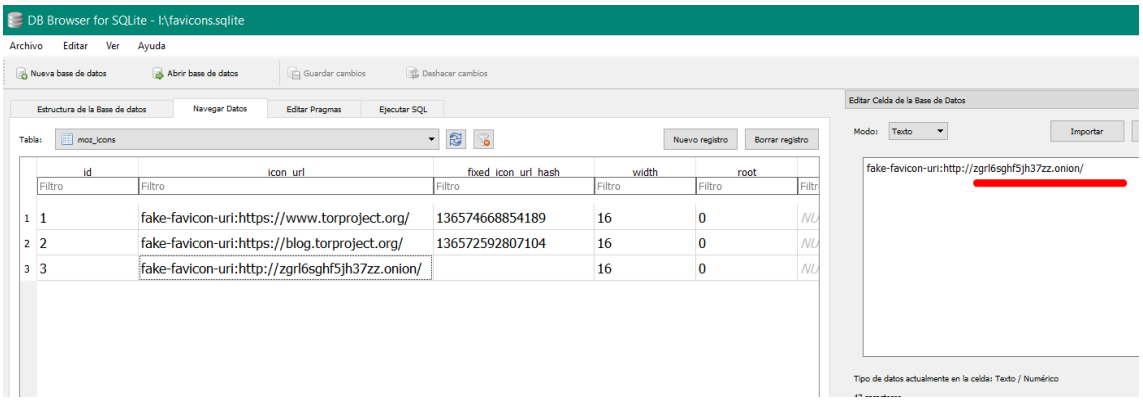
Si el hash se encuentra cargado en memoria podemos probar diferentes servicios en internet como CrackStation o hacer uso de la herramienta John The Ripper o Hashcat para intentar romper el hash.



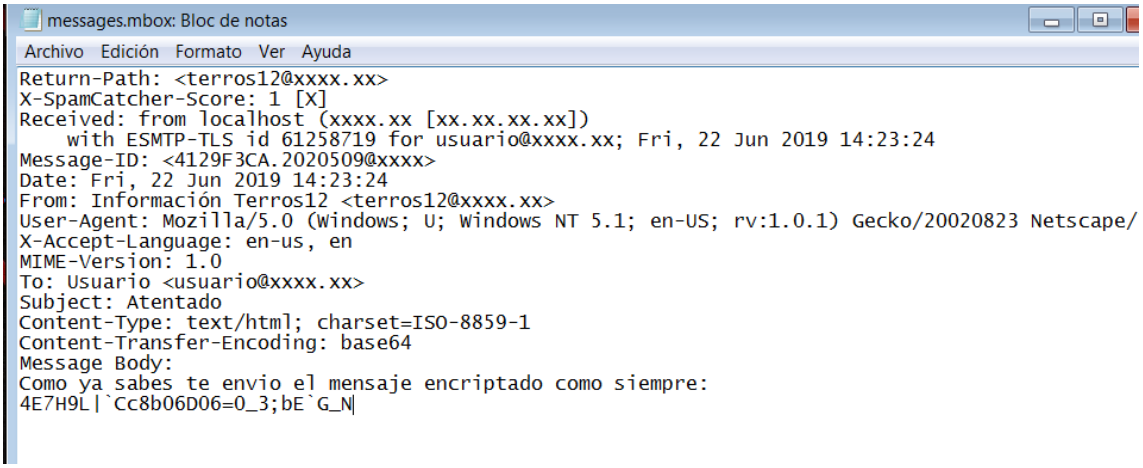
Analizando la memoria RAM podemos ver que en el histórico de comandos del terminal cargado en memoria ha eliminado un fichero con el comando del.

```
root@kali:/media/seguridad# volatility -f MemoryDumpWH.raw --profile=Win7SP0x86 cmdscan
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: conhost.exe Pid: 3904
CommandHistory: 0x3c0470 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 3 LastAdded: 2 LastDisplayed: 2
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x58
Cmd #0 @ 0x3bbc00: cd Desktop
Cmd #1 @ 0x3b4d30: del F1ch3r0_D3l3t3333333.txt [redacted]
Cmd #2 @ 0x3bd890: dir
```

Se puede comprobar que en su disco duro se encuentra TOR Browser instalado. Podemos intentar averiguar cuales son los .onion que ha visitado en la Red TOR haciendo uso de la base de datos de tipo SQLite de los favicons.



Analizando más evidencias del disco duro se puede visualizar que tuvo el cliente de correo electrónico Thunderbird instalado. Aunque el programa ya no se encuentre en el equipo se pueden visualizar aún los correos electrónicos que contenía. Se observa que contiene un correo electrónico donde indica que “M1r4g3_es_el_0bj3t1v0”.



En el disco duro también existe un fichero de base de datos de Keepass pero está cifrado con contraseña. Para ello podemos romper el hash con John The Ripper y Hashcat.

python2 keepass2john passwords.kdbx > pass.hash

hashcat -m 13400 -a 0 -w 1 pass.hash ../rockyou.txt --force

.\hashcat64.exe -m 13400 -a 0 -w 1 .\hashkeepass.txt .\rockyou.txt --force --show

```
root@kali:/media/seguridad# keepass2john passwords.kdbx > db_whctf.hash
root@kali:/media/seguridad# cat db_whctf.hash
passwords:$keepass$*2*60000*222*8211439106de32bcea45bb4ca351afbbe901aaeeb09387c6fe77745b47f570c3*f330c25a675c789da3
1718b400ac329d3ba080504f27e78d2ccb507e7d0aa4c0*c484d089822a7301ec3cfae90a625273*3c98f58a7f3c8660197e24fdded02394d3c
cc382cc3002be39f2b75d24a2fd74*1867e1e5f9beb3b2bb3c4c3e708084cddeeea4731f72748541db80e33d048548
```

```
root@kali:/media/seguridad# hashcat -m 13400 -a 0 -w 1 db_whctf.hash /usr/share/wordlists/rockyou.txt --force
hashcat (pull/1273/head) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 1024/3015 MB allocatable, 2MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.
Watchdog: Temperature retain trigger disabled.
```

```
Dictionary cache building /usr/share/wordlists/rockyou.txt: 33553434 bytes (23.9Dictionary cache building /usr/sha
e/wordlists/rockyou.txt: 67106869 bytes (47.9Dictionary cache building /usr/share/wordlists/rockyou.txt: 134213744
bytes (95.Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14343297
* Runtime...: 3 secs

- Device #1: autotuned kernel-accel to 128
- Device #1: autotuned kernel-loops to 64
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => [s]tatus [p]ause [r]es$keepass$*2*60000*222*8211439106de
2bcea45bb4ca351afbbe901aaeeb09387c6fe77745b47f570c3*f330c25a675c789da31718b400ac329d3ba080504f27e78d2ccb507e7d0aa4
0*c484d089822a7301ec3cfae90a625273*3c98f58a7f3c8660197e24fdded02394d3ccc382cc3002be39f2b75d24a2fd74*1867e1e5f9beb3
2bb3c4c3e708084cddeeea4731f72748541db80e33d048548:iloveyou2

Session..........: hashcat
Status...........: Cracked
```

Si analizamos el dispositivo USB se puede comprobar que está cifrado con Bitlocker. En el keepass que anteriormente hemos crackeado se puede visualizar que existe una contraseña para descifrar este dispositivo. Dentro del USB se puede visualizar un documento PDF con la clave para la bomba que se quería plantar.