# Taller 2.1 - Recuperación de Contraseñas

Ismael Macareno Chouikh

2024-12-13

## Índice

# 1. Entorno

Máquina Virtual Ubuntu 24.04 LTS

- RAM: 4096 MiBi
- CPUs: 2
- Almacenamiento: 65 GiBi

# 2. Ejercicio 1: Rompiendo contraseñas con *john the ripper*

***John the ripper*** es un clásico entre las herramientas de ruptura de contraseñas por métodos de fuerza bruta. Originario de sistemas Linux actualmente está disponible para múltiples sistemas operativos. La herramienta presenta múltiples opciones pero esto no hace que su uso sea demasiado complejo, vamos a ver algunos de los pasos básicos.

## 2.1. Instalación

Para instalar el programa habrá que ejecutar el siguiente comando:

- `sudo apt install john`

Probar el rendimiento es bastante interesante debido a que dependiendo de la potencia de nuestra máquina tendrá más o menos rendimiento

```
maka@makasad:~$ john --test
Created directory: /home/maka/.john
Will run 2 OpenMP threads
Benchmarking: descrypt, traditional crypt(3) [DES 128/128 SSE2]... DONE
Many salts:     10928K c/s real, 5475K c/s virtual
Only one salt:  10249K c/s real, 5124K c/s virtual

Benchmarking: bsdicrypt, BSDI crypt(3) ("_J9..", 725 iterations) [DES 128/128 SSE2]... DONE
Many salts:     338432 c/s real, 169385 c/s virtual
Only one salt:  342425 c/s real, 171212 c/s virtual

Benchmarking: md5crypt [MD5 32/64 X2]... DONE
Raw:    32972 c/s real, 16502 c/s virtual

Benchmarking: bcrypt ("$2a$05", 32 iterations) [Blowfish 32/64 X3]... DONE
Raw:    2527 c/s real, 1263 c/s virtual

Benchmarking: LM [DES 128/128 SSE2]... DONE
Raw:    79355K c/s real, 39757K c/s virtual

Benchmarking: AFS, Kerberos AFS [DES 48/64 4K]... DONE
Short:  576819 c/s real, 575667 c/s virtual
Long:   2598K c/s real, 2598K c/s virtual

Benchmarking: tripcode [DES 128/128 SSE2]... DONE
Raw:    7916K c/s real, 3958K c/s virtual

Benchmarking: dummy [N/A]... DONE
```

```
Raw:    90879K c/s real, 90698K c/s virtual

Benchmarking: crypt, generic crypt(3) [?/64]... DONE
Many salts:     386246 c/s real, 193316 c/s virtual
Only one salt:  398342 c/s real, 199570 c/s virtual
```

## 2.2. Caso práctico 1: *Cracking* de contraseñas por fuerza bruta

En primer lugar, para usar *John* como *crackeador* de sistemas de archivosc, es posible usar la utilidad ***unshadow*** mencionada anteriormente, en un sistema GNU/Linux encontraremos los ficheros

- `/etc/passwd`
- `/etc/shadow`

con estos dos ficheros se puede extraer la información correspondiente a los *hashes* de las cuentas del SSOO y posteriormente usar *John* con la información que nos facilite el comando `unshadow`.

Antes de usar el comando `unshadow`, debemos copiar los dos ficheros a otra carpeta para no comprometer el sistema. El comando para copiar es:

- `cp [origen] [destino]`
- `unshadow /etc/passwd /etcshadow >mypasswd`

Con el comando `unshadow` de arriba los *hashes* se almacenaran en un fichero llamado mypasswd que después puede ser usado por *John*.

- Crea un fichero de texto con el siguiente contenido `user:AZl.zWwxIh15Q`

```
maka@makasad:~$ echo "user:AZl.zWwxIh15Q" > fichero.txt
maka@makasad:~$ cat fichero.txt
user:AZl.zWwxIh15Q
```

- Desde un terminal de usuario acceder al directorio donde está dicho fichero y teclea `john fichero.txt`

```
maka@makasad:~$ john fichero.txt
Loaded 1 password hash (descrypt, traditional crypt(3) [DES 128/128 SSE2])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: MaxLen = 13 is too large for the current hash type, reduced to 8
example          (user)
1g 0:00:03:50 3/3 0.004331g/s 7540Kp/s 7540Kc/s 7540KC/s exi59a3..exam0b'
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Dale tiempo al sistema, mientras trabaja puedes dedicarte a hacer la otra parte de la práctica. Ten en cuenta que *John the Ripper* usa un sistema de fuerza bruta que suele ser lento. Cuando el proceso finalice ejecuta el siguiente comando para ver la contraseña `john --show fichero.txt`

```
maka@makasad:~$ john --show fichero.txt
user:example

1 password hash cracked, 0 left-
```

## 2.3. Caso práctico 2: *Cracking* de contraseñas por diccionario de claves

A continuación usaremos un diccionario de claves, podemos buscar uno en internet pero lo mejor en este caso es usar uno propio asegurándonos que la palabra **example** está en él.

Usa el fichero contraseñas.txt del taller o crea un fichero llamado ***passwords.txt*** e introduce una palabra clave en cada línea, con 10 palabras nos vale para esta prueba.

```
maka@makasad:~$ vi passwords.txt
maka@makasad:~$ cat passwords.txt
bobesponja
calamardo
arenita
crustaceo
crujiente
chico
percebe
tritonman
example
pera
manzana
platano
```

Ahora volveremos a ejecutar *John the ripper*, pero con el parámetro `--wordlist` seguido de la ruta de nuestro archivo:

- `john --wordlist=passwords.txt fichero.txt`

El proceso será instantáneo, ya que el diccionario es pequeño.

```
maka@makasad:~$ john --wordlist=passwords.txt fichero.txt
Loaded 1 password hash (descrypt, traditional crypt(3) [DES 128/128 SSE2])
No password hashes left to crack (see FAQ)
```

**Si te falla**

Deberías asegurarte que los dos estén codificados en la misma opción y que el hash está perfectamente escrito

Ahora con la opción `--show` podemos ver si nuestra *password* ha sido vulnerada

```
maka@makasad:~$ john --show fichero.txt
user:example

1 password hash cracked, 0 left
```

## 2.4. La prueba de fuego

En un sistema GNU/Linux crea diferentes usuarios con claves de diferente longitud y complejidad, podemos ir desde el típico, **admin**, **user**, **123456** hasta claves del estilo de **EulDlMdCNnMQA2021!**. Debes tener en cuenta que la práctica puede tardar mucho tiempo si creas claves muy complejas por lo que puede ser interesante que primero hagas todo el proceso con claves cortas y simples y luego vayas complicando el proceso creando claves más complicadas.

Una vez realizado ese proceso crea una copia del fichero donde Linux almacena las claves (`/etc/shadow`) en un directorio de tu usuario. A partir de ese momento realiza:

### 2.4.1. Ataque de contraseñas por fuerza bruta

### 2.4.2. Ataque de contraseñas por diccionario (lo puedes crear tu o bajar uno de Internet)

- Creamos los usuarios con el comando `adduser`

```
root@makasad:~# ls -la /home | grep -e calamardo -e arenita -e admin
drwxr-x---  2 admin      admin      4096 Dec  9 13:45 admin
drwxr-x---  2 arenita    arenita    4096 Dec  9 13:46 arenita
drwxr-x---  2 calamardo  calamardo  4096 Dec  9 13:46 calamardo
```

- Hacemos una copia del fichero de contraseñas del sistema (/etc/shadow):
  - `sudo cp /etc/shadow/ passwd.txt`
- Le damos a `passwd.txt` permisos de lectura para no tener que estar usado `sudo` todo el rato
  - `sudo chmod a+r passwd.txt`
- Por otro lado vamos a necesitar un diccionario, lo podemos bajar de internet o usar el creado anteriormente modificandolo un poco para que quede de la siguiente manera

```
maka@makasad:~$ vi passwords.txt
maka@makasad:~$ cat passwords.txt
bobesponja
calamardo
arenita
crustaceo
crujiente
chico
percebe
tritonman
example
pera
manzana
platano
123456
EulDlMdCNnMQA2021!
```

- Luego de tener el diccionario tendremos que ejecutar el siguiente comando
  - `john passwd.txt`

**Si te falla**

Bajar una versión distinta del programa desde Github

```
root@makasad:~# john /home/maka/passwd.txt
Loaded 7 password hashes with 7 different salts (crypt, generic crypt(3) [?/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
```

```
arenitaarenita    (arenita)
alumno            (alumnofct1)
fary              (fary)
alumno            (alumnofct2)
123456            (admin)
```

Como se puede ver en el bloque de código de arriba se muestran las contraseñas de los usuarios

# 3. Ejercicio 2: Quitar la contraseña de una cuenta en *Windows*

## 3.1. Requerimientos

Para este ejercicio vamos a necesitar una MV con cualquier versión de Windows (a partir de XP). En la MV, crea una cuenta de usuario que se llame **sad** con una contraseña muy compleja.

En mi caso usaré una MV *Windows* 7 Ultimate

## 3.2. Creación de cuenta en Windows 10

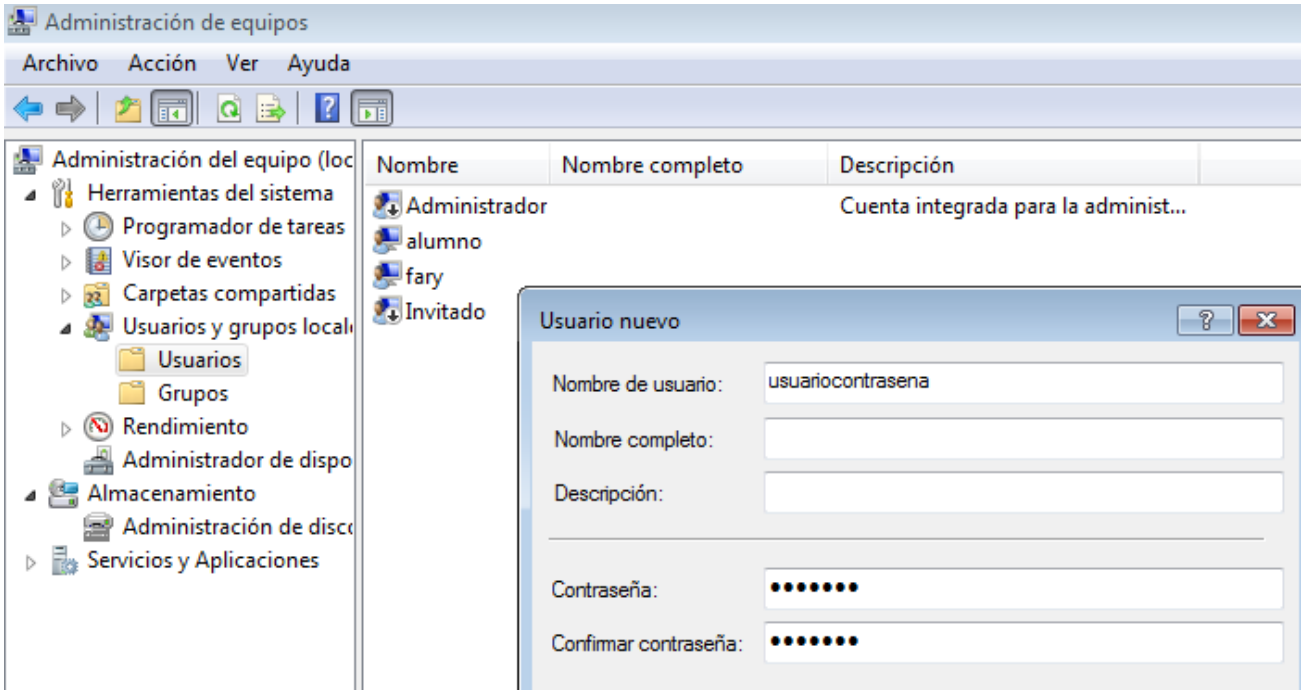


Figura 1: Macareno, Ismael. (2024). Creación de usuario en Windows 7 *Ultimate* [PNG]. Propia

## 3.3. Averiguar contraseña con `mimikatz`

Para averiguar la contraseña con `mimikatz` lo que tendríamos que hacer sería realizar los siguientes pasos:

1. Acceder a https://github.com/ParrotSec/mimikatz
2. Descargar la versión de 64 bits
3. Ejecutar el programa en nuestro *Windows* como **administradores**
4. Ejecutar los siguientes comandos:

- `privilege::debug`
- `sekurlsa::logonpasswords`

Cuando ejecutemos esos dos comandos internos del programa nos aparecerán todos los usuarios con sus respectivas contraseñas

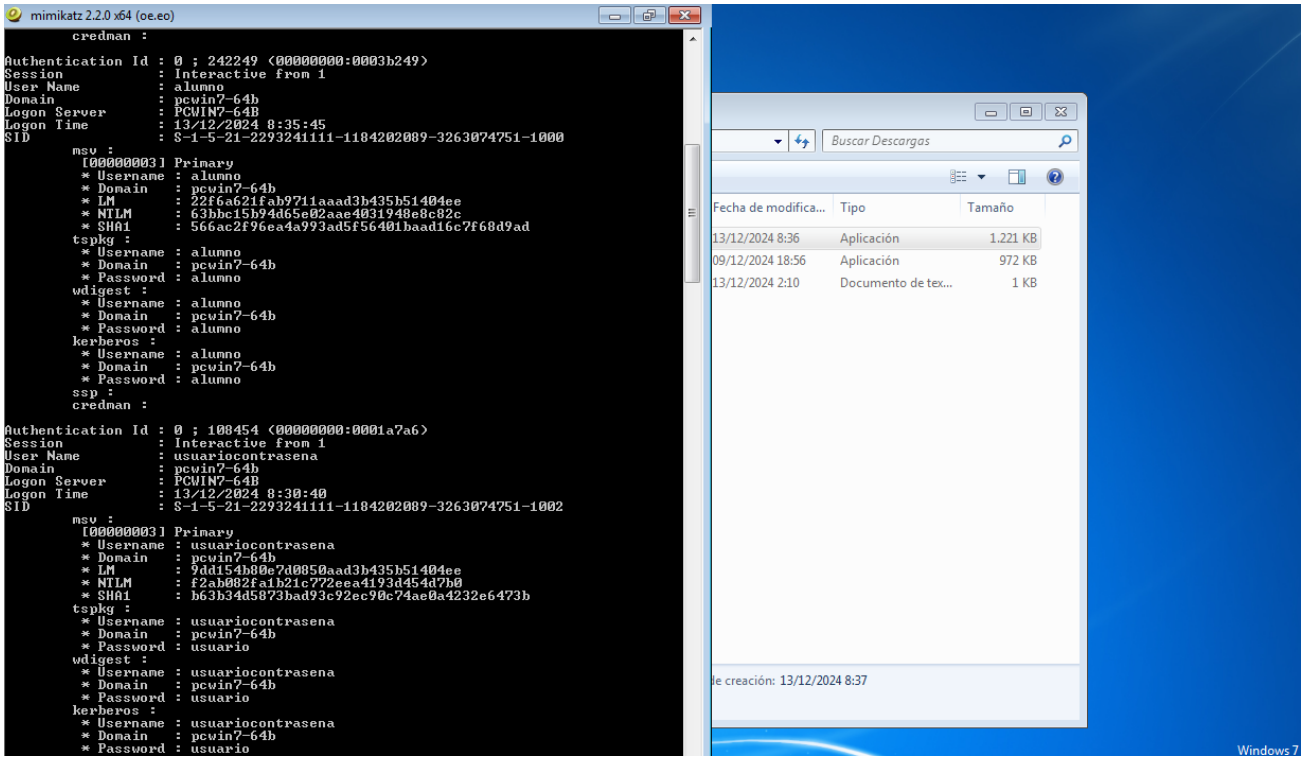


Figura 2: Macareno, Ismael. (2024). Resultado del comando `sekurlsa::logonpasswords` [PNG]. Propia

### 3.4. Cambiar la contraseña con `mimikatz`

Para modificar la contraseña con `mimikatz` los pasos a seguir serían los siguientes:

1. Acceder a https://github.com/ParrotSec/mimikatz
2. Descargar la versión de 64 bits
3. Ejecutar el programa en nuestro *Windows* como **administradores**
4. Ejecutar los siguientes comandos:
   - `privilege::debug`
   - `sekurlsa::logonpasswords`
   - `accounts::setpassword /user:<username>/newpassword:<newpassword>`

Cuando ejecutemos tres comandos internos del programa se supone que se deberían haber modificado las contraseñas del usuarios que hayamos elegido (En mi caso el usuario `usuariocontrasena` con *password* `usuario`)

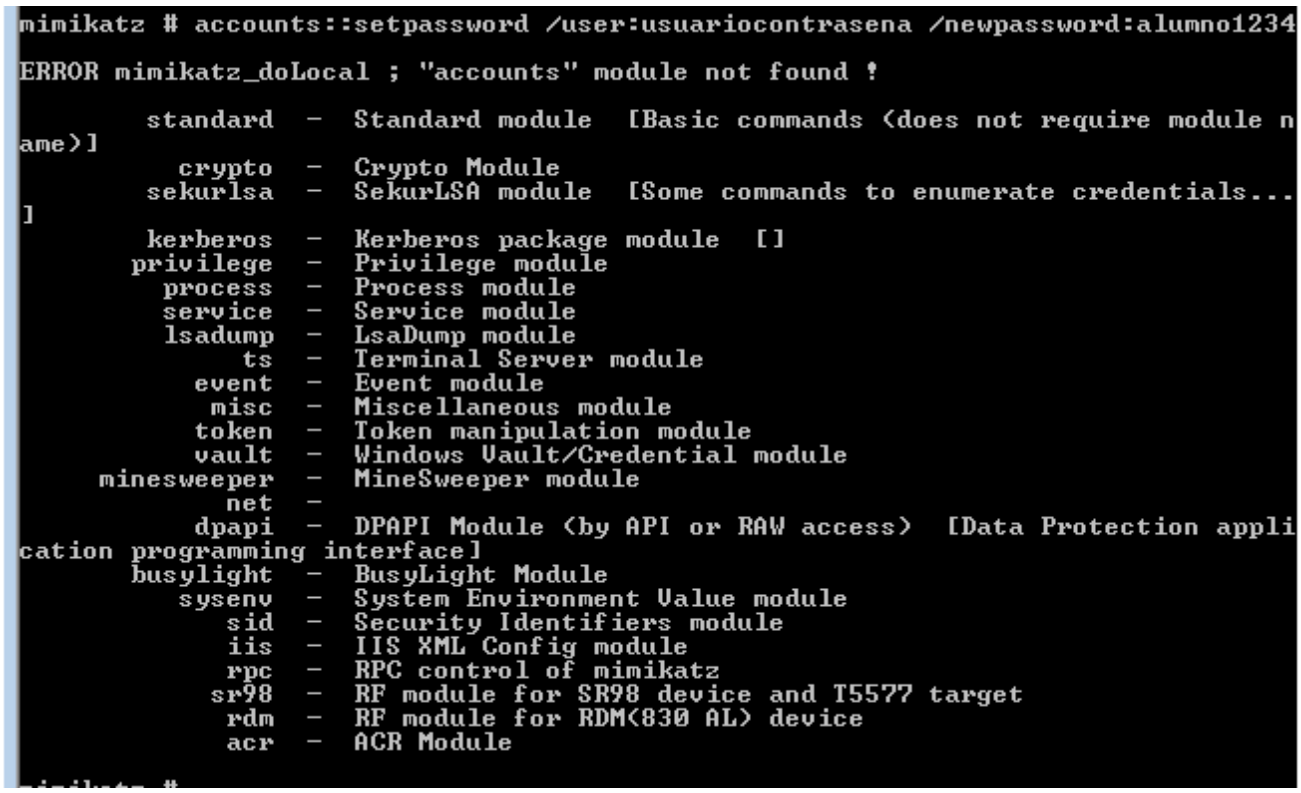


Figura 3: Macareno, Ismael. (2024). Resultado del comando `accounts::setpassword /user:usuariocontrasena /newpassword:alumno1234` [PNG]. Propia

## 4. Ejercicio 3: `Hydra`

Hydra es un conocido Software que intenta romper por fuerza bruta la contraseña de una cantidad impresionante de protocolos: TELNET, FTP, HTTP, HTTPS, HTTP - PROXY, SMB, SMBNT, MS -SQL, MYSQL, REXEC,RSH, RLOGIN, CVS, SNMP, SMTP -AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, LDAP2, LDAP3, Postgres, Teamspeak, Cisco auth, Cisco enable, AFP, LDAP2, Cisco AAA (incorporado en el módulo de Telnet).

Su éxito se debe a un algoritmo que asegura ser el más eficiente y rápido en este tipo de ataques. Hydra viene instalado en Kali Linux aunque también existe versión para Windows. Aunque su descarga e instalación está protegida por la mayoría de antivirus del mercado.

El uso básico de Hydra es:

- `hydra -l user -P diccionario.txt -vV 127.0.0.1 ftp`

### 4.1. Sentencias elementales

- `-l`: Es el nombre de usuario, si se usa en mayúscula (-L) se puede poner un diccionario (*.txt) con usuarios (muy práctico cuando no se sabe el usuario) [-l admin] [-L admin.txt].
- `-p`: Se pone el password, si se usa en mayúscula (-P) se puede poner un diccionario de passwords (es lo más lógico usar esta opción) [-p password] [-P diccionario.txt].
- `-v`: Es el verbose mode que imprime en pantalla los intentos de usuarios-password, si se usa con mayúscula (-V) Hydra nos dará más detalles del proceso de crackeo [-v] [-vV].

### 4.2. Sentencias avanzadas

- `-R`: esto restaura la sesión anterior que se nos haya caído (muy normal cuando nuestro pc no es muy bueno) o que hayamos parado el ataque por algún motivo [-R]
- `-S`: se conecta por SSL [-S]
- `-s`: Se especifica un puerto por si no es el por defecto en el protocolo (ejemplo: telnet (23), ftp (21), smtp (25), etc...) [-s 4450].
- `-C`: Esta sentencia se usa eliminando -l/-L y -p/-P ya que aquí se especifica un diccionario combo, osea que tenga tanto usuario como contraseña con el formato user:pass [-C diccionario-combo.txt].
- `-o`: Esto es muy útil ya que nos ira dejando en un documento que nosotros especifiquemos todas las contraseñas y usuarios que vaya sacando [-o output.txt].
- `-f`: Se cierra Hydra después de encontrar el primer password [-f].
- `-w`: Con este se puede especificar el tiempo máximo (en segundos) que queramos que este crackeando passwords [-w 9999999].
- `-t`: Este es uno de los más útiles si es que cuentas con un computador con buenas características y buena banda ancha, ya que permite cambiar la cantidad de contraseñas/passwords que se prueban en paralelo que son 16 por defecto [-t 32].

## 5. Clonado de *John the ripper* mediante `git`

1. `git clone https://github.com/openwall/john.git`
2. `cd john/src/`
3. `sudo apt install gcc`
4. `./configure --without-openssl && make`