John the Ripper: El arte de romper contraseñas

"La seguridad es un proceso, no un producto."

Roberto Luzanilla

Estudiante de Ingeniería en Sistemas

13 de mayo de 2025

Índice

1. Introducción a John the Ripper

En el complejo universo de la ciberseguridad, pocas herramientas han permanecido tan relevantes a lo largo del tiempo como John the Ripper. Desarrollada originalmente por Solar Designer, esta herramienta de código abierto se ha convertido en un estándar de facto para la auditoría de contraseñas y la evaluación de su fortaleza.

John the Ripper (comúnmente llamado simplemente "John") destaca por su versatilidad y eficacia. Más allá de ser una simple utilidad para romper contraseñas, representa una filosofía fundamental en seguridad: para proteger adecuadamente un sistema, primero debes entender cómo puede ser vulnerado.

Esta herramienta opera bajo diferentes modos de ataque que simulan las estrategias utilizadas por actores maliciosos: desde ataques de fuerza bruta hasta sofisticados ataques basados en reglas y diccionarios personalizados. Su capacidad para detectar y explotar debilidades en los mecanismos de autenticación la convierte en un aliado invaluable para administradores de sistemas y auditores de seguridad.

Lo que distingue a John the Ripper de otras herramientas similares es su equilibrio entre poder, flexibilidad y accesibilidad. Su arquitectura modular permite adaptarse a diferentes escenarios y requisitos, mientras que su continuo desarrollo garantiza compatibilidad con los más recientes algoritmos de cifrado y hash.

En un mundo donde las violaciones de datos son cada vez más frecuentes y sofisticadas, dominar herramientas como John the Ripper no es simplemente una habilidad técnica: es una necesidad para quienes buscan fortalecer activamente la seguridad de sus sistemas y proteger información sensible.

2. Instalación y configuración

La instalación de John the Ripper varía según el sistema operativo. A continuación, se detallan los métodos más comunes.

2.1. Debian/Ubuntu

```
sudo apt update
sudo apt install john
```

Para la versión Jumbo (recomendada por sus características adicionales):

```
sudo apt update
sudo apt install john-data
```

```
git clone https://github.com/openwall/john -b bleeding-jumbo john
cd john/src
./configure && make -s clean && make -sj4
```

2.2. Kali Linux

Kali Linux ya incluye John the Ripper preinstalado. Para verificar la versión:

```
john --version
```

2.3. Windows

Para Windows, se recomienda descargar la versión compilada desde el sitio oficial de Openwall o utilizar la versión de Cygwin:

1. Visita https://www.openwall.com/john/
2. Descarga la versión adecuada para tu arquitectura
3. Extrae el archivo comprimido
4. Accede a la herramienta desde la línea de comandos

2.4. Verificar instalación

Una vez instalado, puedes comprobar que todo funciona correctamente con:

```
john --test
```

2.5. Configuración básica

La configuración principal de John the Ripper se encuentra en el archivo john.conf. Este archivo contiene opciones para personalizar el comportamiento de la herramienta:

```
# Ubicaci n t pica en sistemas Linux
cat /etc/john/john.conf
# o
cat ~/.john/john.conf
```

Los parámetros más importantes a considerar en la configuración incluyen:

- Rutas a diccionarios y reglas
- Limitaciones de recursos (memoria, CPU)
- Configuración de formatos específicos
- Reglas de mutación personalizadas

3. Estructura y componentes

John the Ripper se basa en una arquitectura modular que le permite manejar diferentes tipos de ataques y formatos de hash. Comprender estos componentes es esencial para aprovechar al máximo la herramienta.

3.1. Modos de operación principales

- **Modo de ataque por diccionario**: Prueba palabras de un archivo de texto.
- **Modo de ataque de fuerza bruta**: Prueba sistemáticamente todas las combinaciones posibles.
- **Modo de ataque con reglas**: Aplica transformaciones a palabras de diccionario según reglas predefinidas.
- **Modo incremental**: Un sofisticado ataque de fuerza bruta que prioriza combinaciones más probables.
- **Modo externo**: Permite programar formas personalizadas de generar contraseñas.

3.2. Formatos de hash soportados

John the Ripper soporta una amplia variedad de formatos de hash y cifrado, incluyendo:

- Unix tradicional (DES, MD5, Blowfish, SHA256, SHA512)
- Windows (LM, NTLM)
- Hashes de base de datos (MySQL, PostgreSQL, Oracle)
- Hashes de aplicaciones web (WordPress, Drupal, Joomla)
- Cifrado de archivos (ZIP, RAR, PDF, Office)
- Y muchos más, especialmente en la versión Jumbo

Para ver todos los formatos disponibles en tu instalación:

```
1 john --list=formats
```

3.3. Archivos importantes

- **john.pot**: Almacena las contraseñas ya crackeadas.
- **john.log**: Registro de la actividad reciente.
- **john.conf**: Archivo de configuración principal.

- **password.lst**: Diccionario por defecto (ubicación según instalación).
- **all.sp**: Tablas de estadísticas para modo incremental.

4. Preparando archivos de hash

Antes de iniciar cualquier ataque con John the Ripper, es necesario obtener y preparar los hashes que se intentarán crackear.

4.1. Extracción de hashes

John incluye varias utilidades para extraer hashes de diferentes fuentes:

```
# Extraer hashes de contrase as de Linux
sudo unshadow /etc/passwd /etc/shadow > hashes.txt

# Extraer hashes de archivos zip
zip2john archivo.zip > zip_hash.txt

# Extraer hashes de archivos RAR
rar2john archivo.rar > rar_hash.txt

# Extraer hashes de documentos Office
office2john documento.docx > office_hash.txt

# Extraer hashes de PDFs
pdf2john documento.pdf > pdf_hash.txt

# Extraer hashes de un sistema Windows (requiere acceso)
pwdump > windows_hashes.txt
# o usar herramientas como mimikatz
```

4.2. Formato de archivos de hash

Un archivo de hash típico para John the Ripper tiene este formato:

```
usuario:$id$salt$hash
```

Donde:

- **usuario**: Nombre de usuario asociado al hash
- **id**: Identificador del tipo de hash
- **$salt$**: Valor de salt utilizado (si aplica)

- **$hash$**: El hash de la contraseña propiamente dicho

5. Ataques básicos

Una vez instalado John the Ripper y preparados los archivos de hash, podemos comenzar con los ataques básicos.

5.1. Ataque automático

El modo automático intenta diferentes métodos de ataque en secuencia:

```
john hashes.txt
```

Este modo primero intenta un ataque de "single crack", luego un ataque basado en diccionario con reglas y finalmente un ataque incremental.

5.2. Ataque por diccionario

Este ataque prueba palabras de un archivo de texto:

```
john --wordlist=/ruta/al/diccionario.txt hashes.txt
```

Algunos diccionarios populares incluyen:

- RockYou (filtrado de más de 32 millones de contraseñas)
- Palabras comunes del idioma (ej. /usr/share/dict/words)
- SecLists (colección de múltiples diccionarios para diferentes propósitos)

5.3. Ataque por diccionario con reglas

Las reglas permiten transformar palabras base para crear variantes:

```
john --wordlist=/ruta/al/diccionario.txt --rules hashes.txt
```

John incluye varios conjuntos de reglas predefinidos:

```
john --list=rules
```

Para especificar un conjunto de reglas en particular:

```
john --wordlist=/ruta/al/diccionario.txt --rules=Jumbo hashes.txt
```

5.4. Ataque de fuerza bruta (modo incremental)

El modo incremental es un ataque de fuerza bruta optimizado:

```
john --incremental hashes.txt
```

Para especificar un modo incremental particular:

```
john --incremental=Digits hashes.txt
```

Los modos predefinidos incluyen:

- **All**: Todos los caracteres ASCII imprimibles
- **Alpha**: Solo letras (a-z, A-Z)
- **Digits**: Solo números (0-9)
- **Alnum**: Alfanumérico (a-z, A-Z, 0-9)

5.5. Ver resultados

Para mostrar las contraseñas ya crackeadas:

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

6. Ataques avanzados

John the Ripper ofrece opciones avanzadas para casos donde los métodos básicos no son suficientes.

6.1. Ataque de máscara

Los ataques de máscara permiten especificar patrones para las contraseñas:

```
john --mask="?d?d?d?d-?l?l?l?l" hashes.txt
```

Donde:

- **?d**: Dígito (0-9)
- **?l**: Letra minúscula (a-z)
- **?u**: Letra mayúscula (A-Z)
- **?s**: Símbolo especial
- **?a**: ASCII imprimible completo

6.2. Ataques híbridos

Combinan diccionarios con ataques de fuerza bruta:

```
john --wordlist=/ruta/diccionario.txt --mask="?d?d?d" hashes.txt
```

Este ejemplo añade tres dígitos a cada palabra del diccionario.

6.3. Reglas personalizadas

Para crear reglas personalizadas, edita el archivo john.conf:

```
[List.Rules:MisReglas]
# A adir un n mero del 0 al 9 al final
$[0-9]
# Capitalizar primera letra
c
# Reemplazar 'a' por '@'
sa@
```

Y luego:

```
john --wordlist=/ruta/diccionario.txt --rules=MisReglas hashes.txt
```

6.4. Distribución de trabajo

Para tareas grandes, John permite distribuir el trabajo:

```
# En la primera m quina (nodos 1/4)
john --incremental --node=1/4 hashes.txt

# En la segunda m quina (nodos 2/4)
john --incremental --node=2/4 hashes.txt

# Y as  sucesivamente...
```

6.5. Utilizando GPU

La versión Jumbo de John the Ripper puede utilizar GPU para acelerar el proceso:

```
# Listar dispositivos OpenCL disponibles
john --list=opencl-devices

# Usar un dispositivo espec fico
john --format=sha512crypt-opencl --opencl-device=1 hashes.txt
```

7. Optimización de rendimiento

El rendimiento de John the Ripper puede optimizarse significativamente con los ajustes adecuados.

7.1. Benchmarking

Antes de iniciar un ataque a gran escala, es útil realizar pruebas de rendimiento:

```
john --test
# O para un formato espec fico
john --test --format=md5crypt
```

7.2. Ajustes de memoria

Puedes controlar el uso de memoria:

```
john --fork=4 --mem-file-size=500MB hashes.txt
```

7.3. Multithreading

John puede utilizar múltiples núcleos:

```
john --fork=4 hashes.txt
# o
john --fork=0 hashes.txt # Utiliza todos los n cleos disponibles
```

7.4. Formatos específicos

Especificar el formato correcto puede acelerar enormemente el proceso:

```
john --format=raw-md5 hashes.txt
```

8. Casos de uso prácticos

John the Ripper se utiliza en diversos escenarios en el ámbito de la ciberseguridad.

8.1. Auditoría de políticas de contraseñas

Verificar el cumplimiento de políticas de seguridad:

```
# Extraer hashes
sudo unshadow /etc/passwd /etc/shadow > hashes.txt

```

```
# Realizar auditor a
john --wordlist=top_10000.txt hashes.txt
john --show --users=usuarios_admin.txt hashes.txt
```

8.2. Recuperación de archivos cifrados

Recuperar contraseñas de archivos protegidos:

```
# Para un archivo ZIP
zip2john archivo.zip > zip_hash.txt
john zip_hash.txt

# Para un documento de Office
office2john documento.docx > office_hash.txt
john office_hash.txt
```

8.3. Análisis forense

En investigaciones forenses para recuperar credenciales:

```
# Recuperar contrase as de un volcado de SAM de Windows
john --format=NT hashes_windows.txt

# An lisis de credenciales encontradas
john --show --format=NT hashes_windows.txt > credenciales_recuperadas.
   txt
```

8.4. Evaluaciones de seguridad

Como parte de pruebas de penetración:

```
# Intentar crackear hashes capturados
john --wordlist=diccionarios/enterprise_common.txt --rules=best64
   hashes_capturados.txt

# Generar informe de resultados
john --show --format=md5crypt hashes_capturados.txt >
   informe_contrase as_debiles.txt
```

9. Limitaciones y consideraciones

A pesar de su potencia, John the Ripper tiene limitaciones que deben tenerse en cuenta.

9.1. Factores limitantes

- **Complejidad de contraseñas**: Contraseñas largas y verdaderamente aleatorias pueden ser prácticamente imposibles de crackear.
- **Algoritmos modernos**: Algunos algoritmos están diseñados para ser computacionalmente intensivos (bcrypt, Argon2, etc.).
- **Limitaciones de hardware**: La velocidad depende directamente de los recursos disponibles.
- **Limitaciones de tiempo**: Algunos ataques pueden tomar días, semanas o incluso años.

9.2. Consideraciones éticas y legales

- Solo utilizar en sistemas propios o con autorización explícita.
- Documentar todas las actividades y mantener registros.
- Reportar vulnerabilidades siguiendo procedimientos adecuados.
- Estar al tanto de las regulaciones locales sobre pruebas de seguridad.
- No utilizar contraseñas recuperadas para acceso no autorizado.

10. Mejores prácticas defensivas

Conocer las herramientas de cracking permite implementar mejores defensas.

10.1. Recomendaciones para administradores

- Implementar políticas de contraseñas robustas: longitud mínima, complejidad, caducidad.
- Utilizar algoritmos de hash modernos con factor de trabajo ajustable (bcrypt, Argon2, etc.).
- Implementar bloqueo de cuentas tras intentos fallidos.
- Utilizar autenticación de múltiples factores donde sea posible.
- Realizar auditorías periódicas de contraseñas con John the Ripper.
- Capacitar a los usuarios sobre creación de contraseñas seguras.

10.2. Patrones a evitar en contraseñas

- Palabras de diccionario comunes.
- Sustituciones obvias (a por @, e por 3, etc.).
- Información personal (fechas de nacimiento, nombres, etc.).
- Secuencias de teclado (qwerty, 123456, etc.).
- Contraseñas cortas, independientemente de su complejidad.
- Contraseñas reutilizadas en múltiples servicios.

11. Futuro de la auditoría de contraseñas

El campo de la auditoría de contraseñas continúa evolucionando, impulsado por la constante carrera entre atacantes más sofisticados y defensores mejor preparados. Lo que antes era una simple verificación de fuerza bruta, hoy involucra inteligencia artificial, computación distribuida y un análisis profundo del comportamiento humano.

Herramientas como John the Ripper no sólo seguirán siendo relevantes, sino que también se adaptarán a nuevas formas de autenticación, desde contraseñas gráficas hasta llaves biométricas híbridas. A medida que los sistemas se endurecen, la auditoría no se queda atrás: se vuelve más estratégica, más quirúrgica, más elegante.

En este panorama cambiante, el pentester del futuro no será sólo un conocedor de herramientas, sino un analista del contexto, un lector de patrones, casi un detective digital. Y la auditoría de contraseñas, lejos de desaparecer, se transformará en una disciplina aún más esencial dentro del universo de la ciberseguridad.

Porque al final, donde hay autenticación, siempre habrá alguien tratando de romperla.

11.1. Tendencias emergentes

- Uso de inteligencia artificial para generar patrones de ataque más eficientes.
- Computación cuántica y sus implicaciones para algoritmos criptográficos.
- Movimiento hacia autenticación sin contraseñas (biometría, tokens, etc.).
- Mayor uso de autenticación de múltiples factores.
- Evolución de algoritmos de derivación de claves resistentes a la paralelización.

11.2. El papel de John the Ripper en el futuro

- Adaptación a nuevos algoritmos y métodos de autenticación.
- Integración con otras herramientas de seguridad en flujos automatizados.
- Mayor énfasis en ataques dirigidos vs. fuerza bruta pura.
- Evolución para manejar volúmenes más grandes de datos.

12. Conclusión

En este juego de gato y ratón que es la ciberseguridad, *John the Ripper* sigue firme como uno de los clásicos que no pasan de moda. No porque sea bonito ni nuevo, sino porque funciona, y lo hace bien. Cuando se trata de romper contraseñas, esta herramienta es brutalmente efectiva.

Pero más allá del código, *John* representa una filosofía: entender la seguridad desde la mente del atacante. No se trata de romper por romper, se trata de pensar como el enemigo para anticiparse, cerrar puertas y reforzar muros antes de que alguien más intente colarse.

Cada contraseña rota no es solo una victoria técnica: es una lección. Un recordatorio de que ningún sistema es perfecto y de que la única defensa real es la mejora constante. Si algo nos enseña *John the Ripper*, es que la seguridad no es un destino, es un camino.

Y claro, con gran poder viene... bueno, ya sabes. Esta herramienta no es un juguete. Usarla implica tener claro el propósito, el permiso y la ética. Porque la línea entre el profesional y el atacante no la traza la herramienta, la traza la intención.

Así que sí, *John* rompe contraseñas, pero también rompe la comodidad de pensar que todo está bajo control. Y por eso, sigue siendo una herramienta digna de respeto en el arsenal de cualquier hacker ético.

13. Referencias y recursos

- Sitio oficial de John the Ripper: https://www.openwall.com/john/
- John the Ripper GitHub: https://github.com/openwall/john
- Documentación oficial: https://www.openwall.com/john/doc/
- "The Password Cracking Bible Kevin Mitnick
- "Hacking: The Art of Exploitation Jon Erickson
- .ºWASP Testing Guide Sección de pruebas de autenticación
- "Password Attacks: Gaining Access to Networks, Computers, and Websites Heath Adams
- Foros de Openwall: https://www.openwall.com/lists/john-users/