JOHN THE RIPPER

GUÍA COMPLETA

ALEJANDRO G VERA

John The Ripper en Kali Linux: Una Guía Completa de Hacking Ético desde Cero

Alejandro G Vera

1. Introducción a John The Ripper

En el ámbito de la ciberseguridad, la robustez de las contraseñas es una línea de defensa fundamental. Una herramienta clave para evaluar y fortalecer esta defensa es John the Ripper (JTR), un software de código abierto ampliamente reconocido por su capacidad para descifrar contraseñas. Su diseño y funcionalidad lo convierten en un activo invaluable tanto para profesionales de seguridad ofensivos como defensivos.

¿Qué es John The Ripper (JTR)?

John the Ripper es una utilidad de software de código abierto y de distribución gratuita, diseñada específicamente para la auditoría de seguridad de contraseñas y la recuperación de credenciales.[1] Su propósito principal es el

password cracking, un proceso mediante el cual se intentan descubrir contraseñas a partir de sus hashes cifrados.[1] Aunque fue concebido inicialmente para sistemas Unix, su evolución, especialmente a través de la "versión jumbo" y otras compilaciones no oficiales, le ha permitido extender su compatibilidad a una vasta gama de plataformas, incluyendo diversas variantes de Unix (como Linux, *BSD, Solaris, AIX, QNX), macOS, DOS, Win32 y OpenVMS.[2] Esta amplia compatibilidad lo posiciona como una de las herramientas más empleadas en pruebas de seguridad, al integrar múltiples capacidades de descifrado de contraseñas en un único paquete.[3]

Propósito y uso en el Hacking Ético y Pruebas de Penetración

Dentro del marco del hacking ético y las pruebas de penetración, JTR juega un rol crítico. Su aplicación principal radica en la identificación de contraseñas débiles dentro de los sistemas de una organización. Al simular ataques de descifrado, los administradores de sistemas y los pentesters pueden evaluar la fortaleza de las políticas de contraseñas existentes y, consecuentemente, reforzar las defensas antes de que actores maliciosos puedan explotar estas vulnerabilidades.[1] Además de su función ofensiva en un contexto ético, JTR es una herramienta valiosa para la

recuperación de contraseñas perdidas, lo que subraya su utilidad tanto para la identificación de riesgos como para la gestión de incidentes.[3]

Ventajas clave de JTR

Las características intrínsecas de John the Ripper le confieren una serie de ventajas significativas:

- **Código Abierto y Gratuito:** Su naturaleza de código abierto no solo lo hace accesible sin costo, sino que también fomenta la transparencia y permite que una comunidad global de expertos contribuya a su desarrollo y mejora continua.[1]
- **Detección Automática de Hash:** Una de sus funcionalidades más destacadas es la capacidad de detectar automáticamente el tipo de hash de contraseña. Esta característica simplifica enormemente el proceso para el usuario, eliminando la necesidad de identificar manualmente el algoritmo de cifrado, lo que resulta en una mayor eficiencia operativa.[1]
- **Altamente Configurable:** JTR ofrece una flexibilidad considerable a través de sus múltiples modos de ataque y opciones de personalización. Esto permite a los usuarios adaptar los ataques a escenarios de seguridad específicos, maximizando las posibilidades de éxito en la identificación de contraseñas vulnerables.[1]
- **Rapidez y Eficiencia:** La herramienta es reconocida por su velocidad, especialmente cuando se combina con listas de palabras (wordlists) optimizadas y se aprovechan las capacidades de procesamiento del hardware, tanto CPU como GPU.[1]

JTR y Kali Linux: ¿Por qué están integrados?

Kali Linux es una distribución de Linux basada en Debian, meticulosamente diseñada para tareas de seguridad de la información, incluyendo pruebas de penetración, investigación de seguridad, análisis forense e ingeniería inversa.[10] La integración de John the Ripper en Kali Linux no es meramente una conveniencia, sino una decisión estratégica que optimiza el flujo de trabajo de los profesionales de la ciberseguridad. JTR viene preinstalado en Kali Linux, lo que lo convierte en una herramienta estándar

y de acceso inmediato para cualquier usuario de esta distribución.[5]

La preinstalación de JTR en Kali Linux elimina la necesidad de instalaciones o configuraciones adicionales, lo que reduce la carga de trabajo inicial para el profesional.[10] Esto permite a los usuarios concentrarse directamente en la tarea de hacking ético o auditoría, sin desviar tiempo y recursos a la configuración de herramientas. En un campo donde la agilidad y la capacidad de respuesta son fundamentales, esta optimización es crucial, especialmente en escenarios de tiempo limitado o durante la respuesta a incidentes.

La capacidad de JTR para ser utilizado por "hackers, tanto éticos como no" [1] revela una tensión inherente en las herramientas de ciberseguridad. La misma potencia que permite a un actor malicioso comprometer sistemas, capacita a un defensor para identificar y remediar vulnerabilidades. Esta dualidad subraya la importancia de la ética en el uso de estas herramientas y la necesidad de una comprensión profunda de su funcionamiento para ambos lados del espectro de seguridad. Para el profesional, aprender JTR no se limita a la capacidad de "atacar", sino que se extiende a la habilidad de "proteger" de manera más efectiva.

2. Preparando el Entorno: Kali Linux y John The Ripper

Antes de iniciar cualquier operación de descifrado de contraseñas con John the Ripper, es fundamental asegurar que la herramienta esté correctamente instalada y actualizada en el entorno de Kali Linux. Esta preparación es un paso esencial para garantizar la eficiencia y la precisión de las auditorías de seguridad.

Verificación de la instalación de JTR en Kali Linux

Dado que John the Ripper está preinstalado en Kali Linux, su verificación es un proceso directo. Los profesionales pueden confirmar su presencia y la versión instalada ejecutando el comando john con la opción de ayuda.[5]

- **Comando:** john -h o john --help
    - La ejecución de este comando mostrará el menú de ayuda y la información de

uso de JTR, confirmando que la herramienta está instalada y lista para ser utilizada.[3]

Actualización de JTR (si es necesario)

Mantener John the Ripper actualizado es un aspecto crítico para los profesionales de la ciberseguridad. Las actualizaciones no solo proporcionan acceso a las últimas características y mejoras de rendimiento, sino que también garantizan la compatibilidad con nuevos tipos de hash y la incorporación de contramedidas frente a las técnicas de cifrado más recientes.[12]

En Kali Linux, las actualizaciones se gestionan a través del gestor de paquetes apt.

- **Comando para actualizar los repositorios:** sudo apt update
- **Comando para actualizar JTR:** sudo apt install john
  - Este comando instalará la versión más reciente de JTR disponible en los repositorios de Kali Linux.[5]

Para aquellos que buscan las características más vanguardistas o las optimizaciones más recientes, especialmente en la "versión jumbo" o de desarrollo, se recomienda obtener el código directamente del repositorio de Git y compilarlo desde la fuente.[12] En el ámbito del descifrado de contraseñas, los algoritmos de hashing y las técnicas de ataque evolucionan rápidamente. Una versión desactualizada de JTR podría carecer del soporte para los hashes más recientes o de optimizaciones cruciales, lo que la haría ineficaz contra contraseñas modernas. Esto resalta un principio fundamental de la ciberseguridad: la necesidad de una adaptación y mejora continuas para mantenerse al día con las amenazas emergentes.

Conceptos básicos de archivos de contraseñas en Linux (passwd, shadow)

En los sistemas operativos Linux, la gestión de la información de los usuarios y sus contraseñas se realiza a través de archivos específicos. El archivo /etc/passwd contiene información no sensible de los usuarios, como sus nombres de usuario, identificadores de usuario (UID), identificadores de grupo (GID), directorios de inicio y

shells de inicio de sesión.[5] Por otro lado, el archivo

/etc/shadow es de suma importancia, ya que almacena de forma segura los hashes de las contraseñas de los usuarios, siendo accesible únicamente por el usuario root.[13]

La separación de la información de usuario y los hashes de contraseñas en /etc/passwd y /etc/shadow respectivamente, es un mecanismo de seguridad fundamental en los sistemas Linux. Si un atacante logra obtener acceso únicamente al archivo /etc/passwd, no tendrá los hashes de las contraseñas necesarios para realizar un ataque de descifrado. La necesidad de combinar ambos archivos utilizando la utilidad unshadow (que se detallará en la siguiente sección) para que JTR pueda procesarlos, ilustra que un atacante debe escalar privilegios o explotar otra vulnerabilidad para acceder al archivo shadow. Este diseño eleva la barrera para un compromiso exitoso del sistema, ya que la seguridad de los hashes de contraseñas no depende únicamente de su cifrado, sino también de la restricción de acceso al archivo que los contiene.

3. Extracción de Hashes de Contraseñas para John The Ripper

Para que John the Ripper pueda iniciar el proceso de descifrado de contraseñas, es imprescindible que tenga acceso a los hashes de estas. Esta sección detalla los métodos para obtener estos hashes, centrándose en la utilidad unshadow para sistemas Linux y mencionando otras herramientas para una variedad más amplia de fuentes.

El comando unshadow: Combinando /etc/passwd y /etc/shadow

En entornos Linux, los hashes de contraseñas se encuentran resguardados en el archivo /etc/shadow, el cual, por razones de seguridad, no es directamente legible por usuarios comunes. Para que JTR pueda procesar estos hashes, a menudo es necesario fusionar la información de usuario contenida en /etc/passwd con los hashes de /etc/shadow en un formato unificado que JTR pueda interpretar.[5]

La utilidad unshadow, que forma parte del paquete de John the Ripper, está diseñada

precisamente para esta tarea. Requiere privilegios de root para poder leer el archivo /etc/shadow.[5]

- **Ejemplo de comando:**
  Bash
  sudo unshadow /etc/passwd /etc/shadow > hashes.txt

  Este comando toma el contenido de /etc/passwd y /etc/shadow, los combina y redirige la salida a un nuevo archivo denominado hashes.txt. Este archivo hashes.txt contendrá las entradas de usuario junto con sus respectivos hashes de contraseña, en un formato listo para ser procesado por JTR.[5] Es una práctica recomendada establecer
  umask 077 antes de crear el archivo de hashes (mypasswd en algunos ejemplos de documentación) para asegurar que no sea legible por otros usuarios del sistema.[14]

La necesidad de utilizar sudo para ejecutar unshadow [5], junto con la existencia de diversas herramientas

*2john para extraer hashes de archivos como ZIP o KeePass [2], indica que la extracción de hashes rara vez es el primer paso en una cadena de ataque. Para llegar a la fase de descifrado de contraseñas con JTR, un atacante ya debe haber logrado algún nivel de compromiso inicial, como obtener acceso de bajo privilegio a un sistema o haber exfiltrado archivos protegidos. Esto pone de manifiesto que el descifrado de contraseñas es a menudo una fase posterior en una cadena de ataque, y que la protección de los archivos que contienen hashes es tan crucial como la fortaleza de las contraseñas mismas. La seguridad física y una gestión de accesos robusta son, por tanto, tan importantes como las políticas de contraseñas.

Otros métodos de extracción de hashes (mención breve de *2john utilities)

La versión "jumbo" de JTR y sus herramientas auxiliares, conocidas como utilidades *2john, amplían significativamente la capacidad de la herramienta para extraer hashes de una vasta gama de fuentes, más allá de los archivos de sistema Linux.[2] Estas utilidades están diseñadas para preprocesar formatos específicos, convirtiéndolos a un formato que JTR puede entender y procesar.

Algunos ejemplos de estas herramientas y sus aplicaciones incluyen:

- zip2john: Utilizado para extraer hashes de archivos ZIP protegidos con contraseña.[11]
- keepass2john: Diseñado para bases de datos KeePass (.kdbx).[5]
- ssh2john: Permite extraer hashes de claves SSH privadas.[11]
- rar2john: Para archivos RAR comprimidos.[5]
- bitlocker2john: Para unidades de disco cifradas con BitLocker.[5]
- eapmd5tojohn: Para procesar archivos pcap que contienen hashes EAP-MD5.[5]

La existencia de estas múltiples herramientas *2john [2] que preprocesan diferentes tipos de archivos para JTR demuestra que JTR no es una herramienta aislada, sino una pieza central en un ecosistema más amplio de utilidades de seguridad. Su diseño modular permite la integración con otras herramientas para manejar diversos formatos de datos, maximizando su utilidad y eficiencia en escenarios complejos de hacking ético. Esto también implica que el aprendizaje de JTR es una puerta de entrada para comprender cómo se interconectan las diferentes fases y herramientas en una prueba de penetración.

4. Modos de Ataque de John The Ripper

John the Ripper ofrece una variedad de modos de ataque, cada uno diseñado para abordar diferentes escenarios y tipos de contraseñas. La selección del modo adecuado es fundamental para la eficiencia y el éxito de una auditoría de seguridad.

Tabla 1: Modos de Ataque de John The Ripper y sus Características Principales

Modo de Ataque	Concepto Principal	Cuándo Usarlo	Ejemplo de Comando	Ventajas	Desventajas/ Consideraciones
Diccionario (Wordlist)	Toma una lista de palabras, cifra cada una y la	Contraseñas comunes, basadas en palabras, nombres,	john --wordlist=/ruta/a/wordlist.txt hashes.txt	Rápido y efectivo para contraseñas débiles.	Depende de la calidad y exhaustividad de la wordlist.

	compara con el hash objetivo.	etc.			
Fuerza Bruta (Incremental)	Intenta todas las combinaciones posibles de caracteres hasta una longitud definida.	Contraseñas complejas que no están en diccionarios.	john --incremental hashes.txt	Capacidad de crackear cualquier contraseña (dado suficiente tiempo y recursos).	Muy lento y computacionalmente intensivo, especialmente para contraseñas largas.
"Single Crack"	Utiliza información del usuario (nombre de usuario, nombre completo) y aplica reglas de transformación.	Auditorías iniciales, contraseñas basadas en información personal del usuario.	john --single hashes.txt	Muy rápido y eficiente si las contraseñas están relacionadas con el usuario.	Solo efectivo si la contraseña tiene relación con los datos del usuario.
Máscara (Mask Mode)	Define un patrón específico de caracteres (ej. ?l?l?d para 2 letras y 1 dígito).	Cuando se tiene información parcial sobre la estructura de la contraseña.	john --mask='?l?l?l?d' hashes.txt	Permite un ataque más dirigido que la fuerza bruta pura, reduciendo el espacio de búsqueda.	Requiere algún conocimiento previo sobre el patrón de la contraseña.
Externo (External Mode)	Permite definir funciones de cracking personalizadas usando un lenguaje similar a C.	Escenarios muy específicos donde se necesita lógica de mutación avanzada o compleja.	john --external=MiModoCustom hashes.txt	Máxima flexibilidad y personalización para ataques únicos.	Requiere conocimientos de programación y es más complejo de configurar.

Modo Diccionario (Wordlist Attack)

El modo diccionario es uno de los enfoques más comunes y efectivos en el descifrado de contraseñas. Su funcionamiento se basa en tomar una lista predefinida de palabras, conocida como *wordlist*, cifrar cada una de estas palabras utilizando el mismo algoritmo de hash que la contraseña objetivo, y luego comparar el resultado con el hash que se intenta crackear.[1] La eficacia de este método reside en la premisa de que un número significativo de usuarios opta por contraseñas comunes, palabras encontradas en diccionarios, nombres propios o combinaciones sencillas de estos elementos.

Kali Linux, al ser una distribución orientada a la seguridad, incluye varias wordlists útiles. Una de las más conocidas y ampliamente utilizadas es rockyou.txt, aunque John the Ripper también viene con su propia password.lst por defecto.[5] La ruta común para estas wordlists en Kali Linux es

/usr/share/wordlists/.[17]

- **Ejemplo de comando:**
  Bash
  ```
  john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
  ```

  Este comando instruye a JTR para que utilice el archivo rockyou.txt como wordlist en su intento de descifrar los hashes contenidos en hashes.txt.[3]

Para aumentar las posibilidades de éxito, el modo diccionario puede combinarse con **reglas de mangling** (--rules). Estas reglas permiten a JTR modificar las palabras de la wordlist de diversas maneras, como añadir números, cambiar mayúsculas/minúsculas, duplicar palabras o invertirlas, generando así un número mucho mayor de candidatos de contraseña.[5] Las reglas de mangling se definen en el archivo de configuración de JTR (

john.conf o john.ini) y pueden ser personalizadas para adaptarse a patrones específicos.[18]

- **Ejemplo de comando con reglas de mangling:**
  Bash
  ```
  john --wordlist=/usr/share/wordlists/rockyou.txt --rules hashes.txt
  ```

Este comando aplica las reglas de mangling predeterminadas (o las personalizadas si se especifican en john.conf) a cada palabra de la wordlist.[5]

- **Ejemplo de regla personalizada (en john.conf):**
  ```
  $[0-9]
  $[0-9]$[0-9]
  $[0-9]$[0-9]$[0-9]
  ```

  Esta regla, denominada AppendNum, intentaría añadir uno, dos o tres dígitos al final de cada palabra de la wordlist.[19] Para aplicar una regla personalizada específica, el comando sería:
  Bash
  ```
  john --wordlist=wordlist.txt --rules:AppendNum hashes.txt
  ```

Modo de Ataque de Fuerza Bruta (Incremental Mode)

El modo de fuerza bruta, conocido en JTR como "incremental mode", es el enfoque más exhaustivo para el descifrado de contraseñas. En este modo, JTR intenta sistemáticamente todas las combinaciones posibles de caracteres hasta una longitud determinada.[1] Este método es particularmente útil para contraseñas que no se encuentran en diccionarios o wordlists, ya que no depende de una lista preexistente. Sin embargo, es computacionalmente muy intensivo y, por lo tanto, lento, especialmente cuando se trata de contraseñas de mayor longitud. Para optimizar el proceso, JTR utiliza tablas de frecuencia de caracteres, priorizando las combinaciones más probables en sus intentos.[4]

La especificación de conjuntos de caracteres y la longitud son parámetros clave en este modo. Se pueden definir conjuntos específicos (por ejemplo, solo caracteres alfabéticos, solo numéricos, alfanuméricos, o todos los caracteres posibles) y rangos de longitud para el ataque.[15]

- **Ejemplo de comando básico:**
  Bash
  ```
  john --incremental hashes.txt
  ```

  Este comando inicia un ataque incremental utilizando el conjunto de caracteres predeterminado de JTR.[3]

- **Ejemplo con charset específico:**
  Bash
  ```
  john --incremental:alpha hashes.txt
  ```

  Esto restringe el ataque a combinaciones que solo incluyen caracteres alfabéticos (letras). Otros *charsets* comunes incluyen digits (solo números), lanman (letras, números y algunos caracteres especiales), y all (todos los caracteres posibles).[3]

Modo "Single Crack"

El modo "Single Crack" es el más rápido de los modos de ataque de John the Ripper y se recomienda como punto de partida en cualquier auditoría. Su eficiencia radica en que JTR utiliza información directamente relacionada con el usuario, como nombres de inicio de sesión, nombres completos (del campo GECOS) y nombres de directorios de inicio, y aplica un extenso conjunto de reglas de mangling a esta información para generar candidatos de contraseña.[4] Este enfoque es altamente eficiente porque la información generada se prueba únicamente contra los hashes de las cuentas de las que se obtuvo, lo que reduce drásticamente el espacio de búsqueda.

Una característica adicional de este modo es que las contraseñas que se descifran con éxito también se prueban automáticamente contra todos los demás hashes cargados. Esto es útil en escenarios donde varios usuarios podrían compartir la misma contraseña, maximizando el impacto del ataque inicial.[16]

- **Ejemplo de comando:**
  Bash
  ```
  john --single hashes.txt
  ```

  Este comando inicia el ataque en modo "single crack" contra el archivo de hashes proporcionado.[3]

Modo Máscara (Mask Mode)

El modo máscara de John the Ripper permite a los profesionales definir patrones

específicos para las contraseñas que se intentarán descifrar. Este enfoque es particularmente útil cuando se dispone de alguna información previa sobre la estructura probable de la contraseña, por ejemplo, si se sabe que "comienza con una letra mayúscula, seguida de tres letras minúsculas y dos dígitos".[4]

Para construir la máscara, se utilizan caracteres especiales que representan clases de caracteres: ?l para letras minúsculas, ?u para letras mayúsculas, ?d para dígitos, ?s para símbolos, entre otros.[4]

- **Ejemplo de comando:**
  Bash
  `john --mask='?l?l?l?d' hashes.txt`

  Este comando intentará descifrar contraseñas de cuatro caracteres, compuestas por tres letras minúsculas seguidas de un dígito.[3]

El modo máscara también puede combinarse con el modo incremental para ataques más sofisticados.

- **Ejemplo combinado con incremental:**
  Bash
  `john --incremental=alpha --mask='?u?w' --min-length=7 --max-length=9 hashes.txt`

  Este comando intenta contraseñas con una longitud de 7 a 9 caracteres, donde el primer carácter es una letra mayúscula y el resto son caracteres alfabéticos (tanto mayúsculas como minúsculas). Esta combinación aprovecha la flexibilidad de la máscara para definir un patrón inicial y la optimización del modo incremental para explorar el espacio de búsqueda restante de manera eficiente.[21]

Modo Externo (External Mode)

El modo externo de John the Ripper ofrece la máxima flexibilidad al permitir a los usuarios definir sus propias funciones de cracking utilizando un lenguaje de *scripting* similar a C.[3] Este modo es ideal para implementar lógicas de ataque altamente personalizadas que no están cubiertas por los modos predefinidos de JTR.

Dentro del archivo de configuración de JTR, se pueden definir funciones como init(), filter(), generate() y restore(). Estas funciones permiten un control granular sobre el

proceso de generación y filtrado de palabras, adaptándose a escenarios complejos.[22] Este modo es particularmente valioso para situaciones donde se requiere una lógica de mutación de contraseñas muy específica, posiblemente basada en conocimientos de ingeniería social, patrones únicos observados en un entorno particular o vulnerabilidades de diseño.[9]

La relación inversa entre exhaustividad y eficiencia es una consideración fundamental al seleccionar un modo de ataque. El modo diccionario es rápido y eficiente para contraseñas comunes [1], mientras que el modo de fuerza bruta (incremental) es computacionalmente muy intensivo y lento.[4] El modo "single crack" es el más rápido porque su enfoque es altamente dirigido.[16] Esta observación revela una ley no escrita en el descifrado de contraseñas: cuanto más exhaustivo es un ataque (como la fuerza bruta total), más tiempo y recursos consume. Los modos más eficientes (diccionario, single crack) se basan en la probabilidad y la información disponible sobre el objetivo. Para un profesional, esto significa que la elección del modo de ataque no es arbitraria, sino una decisión estratégica basada en la información disponible y los recursos computacionales. La combinación inteligente de modos es clave para el éxito en una auditoría de seguridad.

Además, la capacidad de JTR para adaptarse a escenarios específicos mediante la personalización de wordlists [6], reglas de mangling [18] y lógicas de ataque a través del modo externo [22] es una ventaja competitiva significativa. No se trata simplemente de una herramienta que "hace fuerza bruta", sino de una plataforma que permite a los expertos aplicar su inteligencia y conocimientos sobre el objetivo para optimizar el proceso de descifrado. Esto transforma el

password cracking de una tarea puramente computacional a una que integra la creatividad y el análisis del *pentester*, haciendo que la herramienta sea más potente y efectiva en manos de un profesional experimentado.

5. Gestión de Sesiones y Visualización de Resultados

Los ataques de descifrado de contraseñas pueden ser procesos de larga duración, especialmente cuando se emplean modos de fuerza bruta o wordlists extensas. John the Ripper ofrece funcionalidades robustas para gestionar estas operaciones, permitiendo a los usuarios guardar el progreso y analizar los resultados de manera

eficiente.

Tabla 2: Comandos Esenciales de John The Ripper

Comando/Opción	Descripción	Ejemplo de Uso
john <archivo_hashes>	Inicia el proceso de cracking con el modo predeterminado.	john hashes.txt
--wordlist=<ruta_wordlist>	Especifica una wordlist para el ataque de diccionario.	john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
--rules	Aplica reglas de mangling a las palabras de la wordlist.	john --wordlist=wordlist.txt --rules hashes.txt
--rules:<nombre_regla>	Aplica una regla de mangling específica.	john --wordlist=wordlist.txt --rules:AppendNum hashes.txt
--incremental	Inicia un ataque de fuerza bruta con el charset predeterminado.	john --incremental hashes.txt
--incremental:<charset>	Inicia un ataque de fuerza bruta con un charset específico (ej. alpha, digits).	john --incremental:alpha hashes.txt
--single	Inicia el modo "single crack" utilizando información del usuario.	john --single hashes.txt
--mask='<patrón>'	Inicia el modo máscara con un patrón definido (ej. ?l?l?l?d).	john --mask='?l?l?l?d' hashes.txt
--session=<nombre_sesion>	Inicia una sesión de cracking con un nombre específico para guardarla.	john --session=mi_auditoria hashes.txt

--restore	Reanuda la última sesión de cracking interrumpida.	john --restore
--restore=<nombre_sesion>	Reanuda una sesión de cracking específica por su nombre.	john --restore=mi_auditoria
--show <archivo_hashes>	Muestra las contraseñas crackeadas del archivo especificado.	john --show hashes.txt
--users=<usuario>	Filtra los resultados para mostrar contraseñas crackeadas de un usuario específico.	john --show --users=root hashes.txt
--groups=<grupo>	Filtra los resultados para mostrar contraseñas crackeadas de usuarios en un grupo específico.	john --show --groups=0 hashes.txt
--fork=<num_procesos>	Utiliza múltiples procesos para el cracking (útil en CPUs multi-núcleo).	john --fork=4 hashes.txt
--format=<formato_hash>	Fuerza a JTR a usar un formato de hash específico.	john --format=Raw-MD5 hash.txt

Guardar y restaurar sesiones (--session, --restore)

Los ataques de descifrado de contraseñas, en particular aquellos que emplean fuerza bruta o wordlists muy extensas, pueden prolongarse por horas, días o incluso semanas. John the Ripper proporciona una funcionalidad esencial para gestionar estos procesos de larga duración: la capacidad de guardar el progreso de una sesión y reanudarla en un momento posterior.[3]

- **Guardar una sesión con nombre:**
  Bash
  `john --session=mi_auditoria hashes.txt`

Este comando inicia una sesión de descifrado y la guarda con el nombre mi_auditoria. JTR creará automáticamente un archivo con extensión .rec (por ejemplo, mi_auditoria.rec) en el directorio de trabajo para almacenar el estado actual de la sesión, permitiendo su recuperación futura.[3]

- **Restaurar una sesión:**
  Bash
  ```
  john --restore
  ```

  Este comando reanuda la última sesión de descifrado que fue interrumpida, permitiendo continuar el trabajo desde el punto exacto donde se detuvo.[3]
- **Restaurar una sesión específica:**
  Bash
  ```
  john --restore=mi_auditoria
  ```

  En situaciones donde se gestionan múltiples sesiones guardadas, es posible especificar el nombre de la sesión a restaurar, lo que facilita la organización y el seguimiento de diferentes tareas de auditoría.[3]

La capacidad de JTR para guardar y restaurar sesiones [3] es crucial para ataques que pueden extenderse durante largos periodos. Esto refleja la realidad de las pruebas de penetración en entornos reales, donde los procesos pueden ser interrumpidos por fallos del sistema, necesidades de recursos o simplemente por la duración inherente de la tarea. La gestión de sesiones permite a los pentesters optimizar su tiempo y recursos, reanudando el trabajo sin perder el progreso. Es un testimonio de la robustez de la herramienta para escenarios del mundo real, donde la persistencia es tan importante como la potencia computacional.

Mostrar contraseñas crackeadas (--show)

Una vez que John the Ripper ha logrado descifrar contraseñas, estas se almacenan en un archivo especial denominado john.pot (conocido como *potfile*). Este archivo, por diseño, no está destinado a ser directamente legible por un ser humano.[14] Para visualizar las contraseñas crackeadas de manera legible y organizada, se utiliza la opción

--show.

- **Ejemplo de comando:**
  Bash
  ```
  john --show hashes.txt
  ```

  Este comando mostrará todas las contraseñas que JTR ha logrado descifrar del archivo hashes.txt en un formato comprensible.[3]

Filtrado de resultados (por usuario, grupo, etc.)

Cuando se trabaja con grandes conjuntos de hashes, el volumen de resultados puede ser abrumador. JTR ofrece opciones para filtrar los resultados, permitiendo a los profesionales centrarse en usuarios o grupos específicos, lo que facilita el análisis y la priorización de hallazgos.

- **Filtrar por usuario:**
  Bash
  ```
  john --show --users=nombre_usuario hashes.txt
  ```

  Este comando mostrará únicamente las contraseñas que han sido crackeadas para el usuario especificado por nombre_usuario.[6]
- **Filtrar por UID (Root):**
  Bash
  ```
  john --show --users=0 hashes.txt
  ```

  Para identificar rápidamente las contraseñas crackeadas correspondientes al usuario root (que generalmente tiene el UID 0 en sistemas Linux), se utiliza este comando.[6]
- **Filtrar por grupo:**
  Bash
  ```
  john --show --groups=0,1 hashes.txt
  ```

  Este comando muestra las contraseñas crackeadas para los usuarios que pertenecen a los grupos con GID 0 y 1.[6]

La opción --show y las capacidades de filtrado por usuario o grupo [3] no son simplemente para "ver" contraseñas, sino para "analizar" los resultados de manera efectiva. En una auditoría de seguridad, no basta con descifrar contraseñas; es fundamental identificar qué cuentas son vulnerables, qué patrones de contraseñas

son débiles y qué usuarios representan un mayor riesgo. Las funciones de visualización y filtrado de JTR permiten a los profesionales de seguridad priorizar y presentar hallazgos de manera clara, facilitando la toma de decisiones para implementar contramedidas efectivas y mejorar la postura de seguridad de una organización.

6. Optimización del Rendimiento de John The Ripper

El rendimiento es un factor crítico en el descifrado de contraseñas, ya que la velocidad de procesamiento impacta directamente en la viabilidad de un ataque. John the Ripper está diseñado para maximizar la utilización de los recursos del sistema, tanto CPU como GPU, para acelerar el proceso.

Uso de CPU (OpenMP, SIMD)

John the Ripper es una herramienta altamente optimizada para el procesamiento en CPU. Puede aprovechar las instrucciones SIMD (Single Instruction, Multiple Data) y el procesamiento multi-núcleo a través de OpenMP para acelerar significativamente el descifrado.[9]

- **OpenMP:** Esta tecnología permite a JTR dividir la carga de trabajo entre múltiples núcleos de CPU. Para habilitar OpenMP y aprovechar el paralelismo, a menudo es necesario compilar JTR con la bandera -fopenmp.[7]
- **SIMD (ej. AVX-512, SSE2):** JTR soporta directamente conjuntos de instrucciones SIMD en arquitecturas x86-64, así como AltiVec en POWER y NEON/ASIMD en ARM/Aarch64.[9] Estas instrucciones permiten a la CPU procesar múltiples datos con una sola instrucción, lo que mejora drásticamente la velocidad de descifrado.
- **Consejo de *tuning*:** El número de hilos de OpenMP puede ser controlado explícitamente mediante la variable de entorno OMP_NUM_THREADS antes de ejecutar JTR.[12]
    - **Ejemplo de comando:**
      Bash
      `OMP_NUM_THREADS=4 john --incremental hashes.txt`

Esto forzará a JTR a utilizar 4 hilos de CPU para el ataque, distribuyendo la carga de trabajo y acelerando el proceso.

Uso de GPU (OpenCL/CUDA)

Aunque JTR es notablemente eficiente en CPU, también puede aprovechar el poder de procesamiento paralelo de las GPU a través de OpenCL para ciertos tipos de hashes, especialmente aquellos que son computacionalmente intensivos.[7] Sin embargo, es importante señalar que otras herramientas como Hashcat a menudo demuestran mayor eficiencia en GPU para hashes rápidos, debido a su soporte multi-GPU superior y su capacidad de distribución dinámica de trabajo.[9] JTR puede delegar el "mask mode" a las GPUs cuando trabaja con hashes rápidos como NTLM, aprovechando la potencia de cálculo de la tarjeta gráfica para acelerar la búsqueda de patrones.[9]

Consideraciones para hardware específico

La elección entre el uso predominante de CPU o GPU, o una combinación de ambas, depende en gran medida del tipo de hash que se intenta descifrar y del hardware disponible. JTR es generalmente más rápido en CPU para hashes "lentos" como bcrypt, mientras que Hashcat se destaca para hashes "rápidos" como NTLM cuando se utilizan GPUs.[9] En sistemas que carecen de una GPU dedicada, JTR puede operar eficazmente utilizando solo la CPU, sin dependencia de OpenCL.[9] Para plataformas con recursos específicos, como una Raspberry Pi (que es principalmente CPU-centrada), JTR es la herramienta recomendada. En cambio, para dispositivos con capacidades tanto de CPU como de GPU, como una NVIDIA Jetson, se puede emplear JTR para las tareas de CPU y Hashcat para aprovechar la GPU.[9]

Consejos de *tuning* (--fork, OMP_NUM_THREADS)

Para optimizar aún más el rendimiento de JTR, se pueden aplicar varias técnicas de

tuning:

- **--fork:** Esta opción permite a JTR utilizar múltiples procesos para el descifrado, lo que es particularmente beneficioso en sistemas con múltiples núcleos de CPU, ya que cada proceso puede trabajar en una parte diferente del espacio de búsqueda.[3]
    - **Ejemplo de comando:**
      Bash
      ```
      john --fork=4 hashes.txt
      ```
      Esto ejecutará 4 procesos de JTR en paralelo, cada uno aprovechando un núcleo de CPU disponible.
- **Compilación desde la fuente:** Obtener la última versión "bleeding-edge jumbo" y compilarla desde la fuente permite habilitar optimizaciones específicas del procesador (como SSE4x, AVXx) que pueden aumentar el rendimiento de 2 a 4 veces.[7] Esta compilación personalizada asegura que JTR aproveche al máximo las capacidades de hardware de la máquina.
- **Afinidad del procesador:** En entornos de clúster o máquinas con un gran número de núcleos, se puede optimizar la afinidad de los hilos del procesador (por ejemplo, configurando OMP_PROC_BIND=TRUE para OpenMP) para evitar la migración de hilos entre núcleos. Esta práctica reduce la sobrecarga asociada con el cambio de contexto y mejora la eficiencia del procesamiento.[7]

La complementariedad de las herramientas en ciberseguridad es evidente en el ámbito del descifrado de contraseñas. Los datos indican que JTR es potente en CPU para hashes lentos, mientras que Hashcat sobresale en GPU para hashes rápidos.[8] Se observa que "un rig profesional de password cracking contiene múltiples GPUs, y hashcat es la herramienta para usarlas de forma más eficiente", pero JTR "probablemente también se usará allí".[9] Esto pone de manifiesto que en el mundo real del descifrado de contraseñas, no existe una "bala de plata". Los profesionales no eligen una herramienta sobre otra de forma exclusiva, sino que las combinan estratégicamente para aprovechar sus fortalezas complementarias. La optimización del rendimiento se convierte en un arte que implica comprender las capacidades del hardware, las características del hash y las fortalezas de cada herramienta, lo que permite crear un "arsenal" de descifrado más potente y adaptable.

La inversión en hardware como factor limitante y habilitador es otro aspecto crucial. La discusión sobre CPU, GPU, SIMD y OpenMP [7] subraya que el rendimiento del descifrado está directamente ligado a la capacidad computacional del hardware. Esto destaca que, si bien JTR es una herramienta gratuita y de código abierto, la barrera

real para descifrar contraseñas complejas a gran escala es la inversión en hardware potente. Las organizaciones que buscan auditar sus sistemas de manera efectiva deben considerar la adquisición de equipos con CPUs y GPUs de alto rendimiento. Esto también implica que los actores maliciosos con mayores recursos de hardware poseen una ventaja significativa en su capacidad para romper contraseñas, lo que refuerza la necesidad de implementar medidas de seguridad robustas en las organizaciones.

7. Tipos de Hash Soportados por John The Ripper (Jumbo)

La versatilidad de John the Ripper se manifiesta en su capacidad para procesar una amplia y diversa gama de formatos de hash y cifrado. Esta característica lo convierte en una herramienta integral para la auditoría de seguridad en múltiples capas de una infraestructura.

Tabla 3: Ejemplos de Tipos de Hash Soportados por John The Ripper (Jumbo)

Categoría	Ejemplo de Hash/Cifrado	Herramienta *2john (si aplica)	Notas
Sistemas Operativos	Unix DES-based, MD5-based Linux, OpenBSD Blowfish, Windows LM/NTLM, macOS	unshadow (para Unix/Linux)	Cubre la mayoría de los sistemas operativos modernos y legados.
Aplicaciones Web	WordPress, Raw MD5/SHA-1/SHA-256/ SHA-512	N/A	Hashes comunes utilizados en bases de datos de aplicaciones web.
Redes	WiFi WPA-PSK, Windows Network Authentication	N/A	Permite el análisis de credenciales de red capturadas.

Archivos Cifrados	ZIP, RAR, 7z, PDF, Microsoft Office	zip2john, rar2john, 7z2john, pdf2john, office2john	Permite descifrar contraseñas de archivos protegidos.
Claves Privadas Cifradas	SSH, GnuPG, Carteras de Criptomonedas	ssh2john, gpg2john, bitcoin2john	Fundamental para la recuperación de claves de seguridad.
Sistemas de Archivos/Discos Cifrados	macOS .dmg, BitLocker de Windows	dmg2john, bitlocker2john	Permite acceder a datos en volúmenes cifrados.
Servidores de Bases de Datos	SQL, LDAP	N/A	Auditoría de credenciales de acceso a bases de datos.

Discusión sobre la amplia gama de hashes y cifrados

La versión "Jumbo" de John the Ripper es particularmente potente, ya que soporta "cientos de tipos de hash y cifrado".[1] Esta vasta compatibilidad permite a los profesionales de la seguridad abordar una superficie de ataque muy amplia, que va más allá de las contraseñas de los sistemas operativos tradicionales.

Entre los tipos de hashes de contraseñas de usuario que JTR Jumbo puede procesar se incluyen:

- **Sistemas Unix:** Diversas variantes de Unix (como Linux, *BSD, Solaris, AIX, QNX) y macOS.[1]
- **Windows:** Hashes LM y NTLM, que son formatos de contraseña utilizados en sistemas Windows.[1]
- **Aplicaciones Web:** Hashes de plataformas populares como WordPress.[1]
- **Groupware:** Sistemas de colaboración como Notes/Domino.[2]
- **Servidores de Bases de Datos:** Credenciales de acceso a bases de datos SQL y LDAP.[2]

Además de las contraseñas de usuario, JTR Jumbo es capaz de descifrar otros tipos de datos cifrados, lo que amplía su utilidad en auditorías forenses y de penetración:

- **Capturas de Tráfico de Red:** Incluye la autenticación de red de Windows y hashes de contraseñas WiFi WPA-PSK.[1]
- **Claves Privadas Cifradas:** Como las utilizadas en SSH, GnuPG y carteras de criptomonedas.[2]
- **Sistemas de Archivos y Discos Cifrados:** Por ejemplo, archivos .dmg y "sparse bundles" de macOS, así como volúmenes cifrados con BitLocker de Windows.[1]
- **Formatos de Archivo Comprimido:** Soporte para archivos ZIP, RAR y 7z protegidos con contraseña.[1]
- **Archivos de Documentos:** Capacidad para descifrar contraseñas de documentos PDF y archivos de Microsoft Office.[2]

Mención de herramientas *2john para formatos específicos (e.g., zip2john, keepass2john)

Para muchos de estos formatos, JTR no interactúa directamente con el archivo cifrado. En su lugar, utiliza herramientas auxiliares, conocidas como utilidades *2john, que son programas especializados en extraer el hash del archivo original y convertirlo a un formato compatible con JTR.[2]

Ejemplos de estas herramientas incluyen:

- zip2john: Para archivos ZIP.[11]
- keepass2john: Para bases de datos KeePass.[5]
- ssh2john: Para claves SSH privadas.[11]
- rar2john: Para archivos RAR.[5]
- bitlocker2john: Para unidades BitLocker.[5]
- eapmd5tojohn: Para archivos pcap con hashes EAP-MD5.[5]

La evolución de JTR más allá de las contraseñas de sistemas operativos es un desarrollo significativo. Originalmente concebido para Unix [4], la versión "jumbo" de JTR ahora soporta "cientos" de tipos de hash, incluyendo aplicaciones web, bases de datos, archivos cifrados, etc..[1] Esto demuestra la capacidad de adaptación de la herramienta a un panorama de seguridad en constante cambio. Ya no es solo un "cracker de contraseñas de sistema operativo", sino una suite forense y de auditoría multifuncional. Para los profesionales, esto significa que JTR es una herramienta de valor creciente a medida que la superficie de ataque se expande más allá de los sistemas operativos tradicionales.

La interconexión de la seguridad de la información es un concepto fundamental que JTR ayuda a ilustrar. La herramienta puede descifrar hashes de sistemas operativos, redes, aplicaciones, documentos y archivos comprimidos.[1] Esto subraya que la seguridad de la información es un campo interconectado. Una contraseña débil en una aplicación web o un archivo ZIP puede convertirse en el punto de entrada para comprometer todo un sistema o una red. JTR, al abordar esta diversidad de formatos, ayuda a los auditores a comprender y mitigar los riesgos en toda la cadena de seguridad de una organización, desde el nivel del sistema operativo hasta las aplicaciones y los datos almacenados.

8. Implicaciones de Seguridad y Mitigación

La existencia y la eficacia de herramientas como John the Ripper tienen profundas implicaciones para la seguridad de la información. La capacidad de JTR para descifrar contraseñas subraya la necesidad crítica de implementar y mantener políticas de seguridad robustas.

Importancia de contraseñas fuertes y políticas de seguridad

La facilidad con la que JTR puede revelar contraseñas débiles [11] enfatiza la necesidad imperante de implementar políticas de contraseñas robustas. Las características de una contraseña fuerte son fundamentales para mitigar los riesgos de descifrado:

- **Longitud:** Las contraseñas deben ser lo más largas posible. La complejidad del descifrado aumenta exponencialmente con la longitud de la contraseña, lo que hace que las contraseñas cortas sean triviales de romper en comparación con las más largas.[1]
- **Complejidad:** Una contraseña debe ser una mezcla aleatoria de letras mayúsculas y minúsculas, números y caracteres especiales. Esta diversidad de caracteres aumenta el espacio de búsqueda para un atacante, dificultando los ataques de fuerza bruta.[1]
- **Cambio frecuente:** Las contraseñas deben cambiarse regularmente. Aunque una contraseña sea fuerte, su exposición prolongada aumenta el riesgo de ser comprometida a lo largo del tiempo, por lo que los cambios periódicos son una

medida de mitigación importante.[1]

- **Evitar información personal:** Es crucial evitar el uso de nombres de empresas, nombres de usuario, fechas de nacimiento o cualquier información fácilmente adivinable, ya que estas son las primeras combinaciones que un atacante intentará, a menudo con éxito, utilizando técnicas de ingeniería social o ataques de diccionario personalizados.[15]

John the Ripper, en un contexto ético, puede ser utilizado precisamente para auditar estas políticas, identificando contraseñas que no cumplen con los estándares de seguridad y, por lo tanto, representan un riesgo para la organización.[3]

Mecanismos de bloqueo de cuentas

La implementación de mecanismos de bloqueo de cuentas es una contramedida efectiva contra los ataques de fuerza bruta y diccionario. Estos sistemas están diseñados para bloquear una cuenta después de un número predeterminado de intentos de inicio de sesión fallidos.[1] Esto dificulta significativamente que un atacante pruebe un gran número de contraseñas en un corto período de tiempo, ya que cada intento fallido acerca la cuenta al bloqueo, obligando al atacante a esperar o a cambiar de objetivo.

Protección de archivos de hashes

La protección de los archivos que contienen hashes de contraseñas es tan crítica como la fortaleza de las contraseñas mismas. Si un atacante obtiene acceso físico a un sistema Windows unido a un dominio, puede descargar una copia del Security Account Manager (SAM), el cual contiene los hashes de contraseñas de las cuentas locales.[1] De manera similar, en sistemas Linux, el archivo

/etc/shadow, que almacena los hashes, debe estar protegido con permisos estrictos y ser accesible únicamente por el usuario root.[13] La protección física y lógica de estos archivos es fundamental, ya que una vez que los hashes son obtenidos, el atacante puede intentar descifrarlos sin preocuparse por los mecanismos de bloqueo de cuenta, ya que el ataque se realiza fuera del sistema objetivo.[1]

La capacidad de JTR para "demostrar lo fácil que es revelar contraseñas débiles" [11] va más allá de la auditoría técnica. JTR es una herramienta poderosa para la concientización sobre seguridad. Al mostrar a los usuarios y a la gerencia la rapidez con la que se pueden descifrar contraseñas comunes, se puede ilustrar de manera tangible la importancia de las políticas de contraseñas fuertes y la necesidad de invertir en seguridad. Es una "prueba de concepto" práctica para educar y justificar la implementación de medidas de seguridad más robustas.

La seguridad de las contraseñas no es una solución única, sino un enfoque de defensa en profundidad. La discusión sobre la longitud de las contraseñas, la complejidad, el cambio frecuente, los bloqueos de cuenta y la protección de archivos de hashes [1] demuestra que ninguna medida por sí sola es suficiente. La combinación de contraseñas robustas, políticas de bloqueo, protección de datos sensibles y auditorías continuas (utilizando herramientas como JTR) crea un sistema de defensa resiliente. Esto refuerza la idea de que la ciberseguridad es un proceso iterativo y no un destino, requiriendo una vigilancia constante y una adaptación a las nuevas amenazas.

9. Conclusión

John the Ripper se erige como una herramienta indispensable en el arsenal de cualquier profesional de la ciberseguridad. Su versatilidad, eficiencia y capacidad de personalización lo destacan como una solución robusta para realizar auditorías exhaustivas de la seguridad de las contraseñas en una amplia gama de sistemas y formatos. Desde la detección automática de hashes hasta la implementación de modos de ataque sofisticados como el diccionario, la fuerza bruta, el "single crack" y el modo máscara, JTR permite a los auditores identificar y comprender las vulnerabilidades relacionadas con las credenciales.

Es fundamental recordar que el poder inherente de John the Ripper conlleva una gran responsabilidad. Su utilización debe regirse siempre por principios éticos y legales, y debe contar con el consentimiento explícito de los propietarios del sistema auditado. La herramienta está diseñada para ser empleada en el marco de pruebas de penetración y auditorías de seguridad, con el objetivo final de fortalecer las defensas y mitigar riesgos, no para fines maliciosos.

El panorama de la ciberseguridad se encuentra en constante evolución, con nuevas

técnicas de ataque y contramedidas emergiendo continuamente. Se alienta a los profesionales y entusiastas de la seguridad a continuar explorando las capacidades avanzadas de JTR, a experimentar con diferentes wordlists y reglas de mangling, y a mantenerse actualizados sobre las últimas tendencias en el descifrado de contraseñas y las estrategias de mitigación. La inversión en conocimientos y en hardware adecuado, junto con una comprensión profunda de la complementariedad entre diversas herramientas de seguridad, son pilares para construir una postura de defensa resiliente en el entorno digital actual.

Obras citadas

1. John The Ripper | Bugcrowd, fecha de acceso: julio 25, 2025, https://www.bugcrowd.com/glossary/john-the-ripper/
2. John the Ripper password cracker - Openwall, fecha de acceso: julio 25, 2025, https://www.openwall.com/john/
3. John the Ripper | Hackviser, fecha de acceso: julio 25, 2025, https://hackviser.com/tactics/tools/john-the-ripper
4. John the Ripper - Wikipedia, fecha de acceso: julio 25, 2025, https://en.wikipedia.org/wiki/John_the_Ripper
5. john | Kali Linux Tools, fecha de acceso: julio 25, 2025, https://www.kali.org/tools/john/
6. How to Use John the Ripper: Tips and Tutorials, fecha de acceso: julio 25, 2025, https://www.varonis.com/blog/john-the-ripper
7. Wordlist password cracking using John the Ripper - Publications du gouvernement du Canada, fecha de acceso: julio 25, 2025, https://publications.gc.ca/collections/collection_2018/rddc-drdc/D68-10-25-2018-eng.pdf
8. Password Cracking with John the Ripper | Edureka - YouTube, fecha de acceso: julio 25, 2025, https://www.youtube.com/watch?v=tJRz9j2REb4
9. john-users - hashcat vs. JtR - Openwall, fecha de acceso: julio 25, 2025, https://www.openwall.com/lists/john-users/2020/05/26/1
10. Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution, fecha de acceso: julio 25, 2025, https://www.kali.org/
11. John the Ripper: Password Cracking Tutorial and Review - eSecurity Planet, fecha de acceso: julio 25, 2025, https://www.esecurityplanet.com/products/john-the-ripper/
12. Password Cracking Exercises Introduction - Wellesley College, fecha de acceso: julio 25, 2025, https://cs.wellesley.edu/~cs342/spring16/handouts/passwordCracking.pdf
13. LAB MANUAL ON PASSWORD CRACKING OF KALI LINUX ..., fecha de acceso: julio 25, 2025, https://www.nitttrchd.ac.in/imee/Labmanuals/Password%20Cracking%20of%20Linux%20Operating%20System.pdf
14. John the Ripper - usage examples - Openwall, fecha de acceso: julio 25, 2025,

https://www.openwall.com/john/doc/EXAMPLES.shtml
15. Checking Password Complexity with John the Ripper » ADMIN ..., fecha de acceso: julio 25, 2025, https://www.admin-magazine.com/Articles/John-the-Ripper
16. John the Ripper - cracking modes - Openwall, fecha de acceso: julio 25, 2025, https://www.openwall.com/john/doc/MODES.shtml
17. wordlists | Kali Linux Tools, fecha de acceso: julio 25, 2025, https://www.kali.org/tools/wordlists/
18. John the Ripper - wordlist rules syntax - Openwall, fecha de acceso: julio 25, 2025, https://www.openwall.com/john/doc/RULES.shtml
19. Custom Credential Mutations - Docs | © Rapid7, fecha de acceso: julio 25, 2025, https://help.rapid7.com/metasploit/Content/bruteforce-credentials/credential-mutations.html
20. Distributed Password Cracking with John the Ripper - Department of Computer Science, fecha de acceso: julio 25, 2025, https://www.cs.tufts.edu/comp/116/archive/fall2013/tlubeck.pdf
21. john-users - Re: John The Ripper Incremental Mode - Openwall, fecha de acceso: julio 25, 2025, https://www.openwall.com/lists/john-users/2019/01/31/2
22. John the Ripper - defining an external mode - Openwall, fecha de acceso: julio 25, 2025, https://www.openwall.com/john/doc/EXTERNAL.shtml
23. John the Ripper - command line options - Openwall, fecha de acceso: julio 25, 2025, https://www.openwall.com/john/doc/OPTIONS.shtml
24. More on tuning John the Ripper | Pen Test Partners, fecha de acceso: julio 25, 2025, https://www.pentestpartners.com/security-blog/more-on-tuning-john-the-ripper/