En este artículo te explicaremos cómo intentar descifrar un PDF con contraseña usando un ataque de fuerza bruta con JohnTheRipper.

1. Crear binarios de JohnTheRipper

Tendremos que trabajar con la versión Jumbo de JohnTheRipper. Esta es una versión "jumbo" mejorada por la comunidad de John the Ripper. Tiene mucho código, documentación y datos aportados por la comunidad de usuarios. Este no es el código "oficial" de John the Ripper. Es muy fácil agregar código nuevo a jumbo: los requisitos de calidad son bajos. Esto significa que obtiene una gran cantidad de funciones que no son lo suficientemente "maduras" o que son inapropiadas para el JtR oficial, lo que a su vez también significa que se esperan errores en este código.

Procede a obtener el código fuente de JohnTheRipper (La rama "bleeding-jumbo" (por defecto) se basa en 1.8.0-Jumbo-1) del repositorio en Github con el siguiente comando (o descarga el zip con el contenido y extrae en algún directorio):

git clone https://github.com/magnumripper/JohnTheRipper.git

Esto creará un directorio llamado JohnTheRipper en el directorio actual. Puedes leer más sobre la versión "Jumbo" del proyecto JohnTheRipper en el sitio web oficial o visitar el repositorio de código no oficial en Github aquí . Cambia al directorio src de JohnTheRipper con el siguiente comando:

cd ./JohnTheRipper/src

Procede a descargar las listas de paquetes de los repositorios con el siguiente comando:

sudo apt-get update

E instala libssl:

sudo apt-get install libssl-dev

La biblioteca requiere que libssl (openssl) esté instalado en tu sistema, por lo que en caso de que no lo tengas, el comando anterior hará el truco para cumplir con este requisito. Una vez clonado el repositorio, proceda a ingresar al directorio fuente que contiene el código fuente de JohnTheRipper:

cd ./JohnTheRipper/src

Dentro de este directorio procederemos a la construcción con la siguiente instrucción:

./configure && make

Esta versión de Jumbo tiene autoconf que admite la cadena muy común, lo que le permite compilar las fuentes en un sistema similar a Unix. Una vez que finalice el proceso de compilación, cambie al directorio de ejecución dentro del directorio JohnTheRipper:

cd ..

cd ./run

Dentro de este directorio encontrará (después de la compilación) todas las herramientas que la biblioteca tiene para ofrecer (incluido el propio John), puedes enumerar el directorio para comparar:

ls

Verás todas las herramientas de JohnTheRipper dentro de este directorio:

JohnTheRipper CLI Tools

Ahora que tiene las herramientas para continuar, comencemos con el ataque de fuerza bruta.

2. Generar un archivo hash PDF

JohnTheRipper, como se mencionó al principio del artículo, no está relacionado por sí mismo con los PDF, sino con contraseñas y elementos de seguridad. Es por eso que deberá crear el archivo hash del PDF con la herramienta pdf2john.pl (disponible en el directorio de ejecución después de compilar desde la fuente). Esta herramienta le permite obtener el hash (Leer metainformación) del archivo a través de este script de perl, que se puede extraer en un nuevo archivo con el siguiente comando:

pdf2john.pl /root/Desktop/pdf\_protected.pdf > /root/Desktop/pdf.hash

Este comando creará un archivo .hash en el directorio definido. Este es el archivo que usaremos para trabajar con la herramienta JohnTheRipper:

pdf2john.pl Generate PDF Hash

El archivo pdf.hash contiene un texto como:

/root/Desktop/pdf\_protected.pdf:$pdf$4\*4\*128\*-4\*1\*16\*d22933dd530666ced293ccf5f860214f\*32\*ab12d2e30eae3d43d69ac57efc2918c128bf4e5e4e758a4164004e56fffa0108\*32\*07419b7fe590bdc9926bb9750adeedb8c07b5407d9b5314c012d16e25e9e7f20

Ahora que tenemos el archivo hash, podemos continuar con el ataque de fuerza bruta usando la herramienta CLI de John.

3. Ataque de Fuerza bruta con John

Ahora que tenemos el archivo .hash del PDF con contraseña que queremos desbloquear, solo necesitamos pasar el archivo como argumento a la herramienta CLI de JohnTheRipper (en el directorio run):

john protected\_pdf.hash

Esto usará UTF-8 como la codificación de entrada predeterminada y comenzará a adivinar la contraseña del archivo PDF usando la lista de palabras predeterminada de la biblioteca. Si se encuentra, mostrará la contraseña y la ruta al PDF protegido:

Decrypted PDF Password Kali Linux JohnTheRipper

Si intentsa ejecutar el comando en el mismo archivo después de haber adivinado la contraseña, verás los siguientes mensajes: "No se cargaron hashes de contraseña", "No se cargaron hashes de contraseña" o "No quedan hashes de contraseña por descifrar (consulte las preguntas frecuentes). ) ". Las contraseñas descifradas se imprimirán en el terminal y se guardarán en el archivo llamado $ JOHN / john.pot (en la documentación y en el archivo de configuración de John, "$ JOHN" se refiere al "directorio de inicio" de John; de qué directorio es realmente depende sobre cómo instaló John). El archivo $ JOHN / john.pot también se usa para no cargar hashes de contraseña que ya descifraste cuando ejecutas a John la próxima vez.

Si ese es el caso, podrá ver la contraseña nuevamente del mismo archivo usando la bandera --show:

john --show /root/Desktop/pdf.hash

Entonces se mostrará la contraseña (en nuestro caso 54321):

JohnTheRipper Show Password Hash

Usar una lista de palabras personalizada

Si no desea utilizar el archivo password.lst predeterminado de JohnTheRipper, simplemente especifique la ruta al nuevo archivo usando el argumento --wordlist:

john --wordlist=password.lst protected\_pdf.hash

Como recomendación final, la herramienta ofrece crackear una gran cantidad de archivos, por lo que es posible que desee leer la documentación de la biblioteca . El resto de la documentación se encuentra en archivos separados, enumerados aquí en el orden de lectura recomendado:

INSTALAR - instrucciones de instalación

OPCIONES: opciones de línea de comandos y utilidades adicionales

EJEMPLOS - ejemplos de uso - muy recomendado

MODOS - modos de craqueo: que son

FAQ - preguntas frecuentes

ERRORES: lista de errores conocidos

DYNAMIC - cómo usar formato dinámico en JtR

FORMATOS DEL COMPILADOR DINÁMICO: lista de formatos hash conocidos creados con el compilador dinámico

DYNAMIC\_SCRIPTING: cómo crear / optimizar un formato que utiliza dinámica

README.bash -mentation - cómo habilitar la finalización de bash para JtR

CONTACTO (\*) - cómo contactar al autor u obtener soporte

CONFIG (\*) - cómo personalizar

EXTERNAL (\*) - definir un modo externo

REGLAS (\*) - sintaxis de reglas de lista de palabras

CAMBIOS (\*) - historial de cambios

CRÉDITOS (\*) - créditos

LICENCIA: derechos de autor y términos de licencia

COPIA: GNU GPL versión 2, como se indica en la LICENCIA anterior

(\*) la mayoría de los usuarios pueden omitirlos de forma segura.