Universidad Carlos III
Ingeniería de la ciberseguridad 2023-24
Práctica 1
Estudio de la Fortaleza de Contraseñas

Curso 2022-23

GRUPO: **84**

Autores:

Alejandro García Berrocal (100451059)
Lucas Gallego Bravo (100429005)

Emails: 100429005@alumnos.uc3m.es 100451059@alumnos.uc3m.es

0. Introducción

Esta primera práctica consiste en el estudio de la fortaleza de contraseñas utilizando la herramienta de rotura de John The Ripper. Para ello se pide que se generen 25 datasets de distintos formatos de contraseñas para el estudio de estrategias para la rotura de estas. Todos estos datasets tendrán contraseñas de longitud de 3 a 7 y un total de 100 contraseñas por cada uno.

Se obtendrán un total de 5 datasets para los 5 formatos que se proponen. El primero serán contraseñas formadas por letras minúsculas, el segundo por letras mayúsculas, el tercero por números, el cuarto por caracteres alfanuméricos y el último por el uso de palabras tomadas de diccionarios siguiendo alguna estrategia que se proponga. Estos datasets contienen las contraseñas que más tarde deberán ser pasadas por un hash para proceder con la práctica.

Por último, el equipo decidió también optar por el punto extra, por lo que se repetirán los pasos anteriores y se obtendrán otros 25 datasets siguiendo los formatos anteriores y se aplicará un distinto hash distinto a los anteriores.

1. Generación de datasets

Para la generación de los datasets se ha optado por crear un script en python que genere los datasets de cada formato con las contraseñas en formato hash para ponerlas a prueba con John The Ripper.

1.1 Diseño

Dentro de este script encontraremos una función por cada formato descrito previamente, además para el formato del diccionario se han usado como estrategias los formatos anteriores a este excepto el de sólo números.

Cada una de estas funciones creará o abrirá un documento de texto con el nombre del dataset correspondiente al formato y número de dataset dentro de este formato. Para cada formato de manera aleatoria se escogerán los caracteres adecuados al formato que corresponda e irán incrementando en longitud de 3 a 7 por cada dataset. Para escoger estos caracteres, según el formato se listan números acorde a la tabla ASCII y más tarde se aplica un mapa para determinar el valor del carácter. Una vez se alcance la longitud se unirán en una string, que dará lugar a la contraseña generada, se añadirá al documento de texto, el dataset.

Para el formato de los diccionarios, se han buscado y se han guardado distintos diccionarios en inglés y otros de contraseñas de interés. Siguiendo las estrategias anteriores excepto la numérica, se generará un nuevo wordlist a partir de los guardados. Con dicho wordlist generado se escogerán aleatoriamente contraseñas o palabras y se escogerá de forma aleatoria la estrategia entre los formatos ya especificados. Para el caso del formato alfanumérico se utiliza una función auxiliar que itera por los caracteres de la

contraseña y cambia de forma aleatoria a minúscula, mayúscula, número o carácter especial.

Una vez se obtengan todos los datasets, todos estos se pasarán por una función que los pase a formato hash. Estos formatos hash son SHA256 y MD5 para así cumplir con el punto extra.

El formato de los datasets generados para las contraseñas son 'datasetX_Y.txt' donde X equivale al formato e Y al número. Para las contraseñas en formato hash será 'datasetX_Y_Z.txt' donde X equivale al formato, Y al número y Z al tipo de hash 'SHA256' o 'MD5'.

Con los datasets generados ya se podrían diseñar las estrategias e intentar romper estas contraseñas.

Funciones	Descripción
password_generator_lower	Formato: caracteres minúsculas
password_generator_upper	Formato: caracteres mayúsculas
password_generator_numbers	Formato: caracteres numéricos
password_generator_alphanum	Formato: caracteres alfanuméricos
password_generator_wordlist	Formato: estrategias diccionario
generator_SHA256	Generador de hash SHA256
generator_MD5	Generador de hash MD5
generate_wordlist	Generador de wordlist
modify_string	Auxiliar formato diccionario

2. Definición de estrategias

En esta sección se explican las estrategias que se han creado para aplicar a John The Ripper y romper las contraseñas en formato hash obtenidas previamente. Se ha de mencionar que algunas contraseñas y reglas pueden llegar a ser muy complejas y por lo tanto requiere de mucho tiempo para romperlas. Para ello se estimará un tiempo límite de 10 minutos como máximo para romper las contraseñas. En las siguientes estrategias se llegará a utilizar un diccionario generado con una función a partir de los diccionarios que se han considerados oportunos para las estrategias. Los conjuntos de reglas que se llegarán a utilizar serán las reglas por defecto de John u otros archivos de reglas obtenidos de Internet que se han considerado oportunos para esta práctica.

Las estrategias siguientes que se proponen son de distinta dificultad así encontrando desde sencillas hasta algunas estrategias con más dificultad que las anteriores.

La única regla que se aplicará de Internet es la denominada 'OneRuleToRuleThemAll'. Por otro lado, se consideró otra regla pero no se llegó a aplicar aún siendo interesante lo que ofrecía, su nombre las reglas de 'NyxGeek'.

2.1 Estrategia 1

Un caso en el que se intenta romper contraseñas de usuarios de una página web para comprar de segunda mano con el que se cree poder robar información útil de los usuarios. Se sabe que la longitud mínima que establece esta página web es de 3. Los usuarios tienen permitido el uso de números y caracteres especiales.

La primera estrategia propuesta es un ataque de fuerza bruta incremental en modo 'alnum'. En esta estrategia se definirá con una longitud mínima de 3 de la contraseña, que prueba todas las combinaciones posibles desde este rango establecido. El comando para la ejecución se realizará desde la carpeta donde se generan los datasets y los hashes.

Para la ejecución de esta estrategia se usa el siguiente comando:

john --format=Raw-SHA256 --incremental=alnum --min-length=3 <dataset>

john --format=Raw-MD5 --incremental=alnum --min-length=3 <dataset>

2.2 Estrategia 2

Un caso en que se intentan romper contraseñas de usuarios de una página de compra de videojuegos en formato digital. Se cree que si se rompen las contraseñas de los usuarios se pueden obtener datos bancarios. Se les pide a los usuarios tener contraseñas con un mínimo de 4 caracteres, por lo general parte de estos usuarios usa una longitud de 7 a 9. La página pide que como mínimo haya números y permite el uso de caracteres especiales.

La segunda estrategia que se propone es un ataque con máscara de alfanuméricos con una longitud de 9 de la contraseña. Para la ejecución de esta estrategia se usa el siguiente comando:

john --format=Raw-SHA256 --mask=?a?a?a?a?a?a?a?a?a --min-length=4 <dataset>

john --format=Raw-MD5 --mask=?a?a?a?a?a?a?a?a?a --min-length=4 <dataset>

2.3 Estrategia 3

Se tiene un caso en el que se intentan romper contraseñas de los usuarios de una consultoría de la que se espera obtener datos de interés. Se estima que el mínimo de caracteres que piden en la aplicación es un mínimo de 4 y un máximo de 20. Se tiene un diccionario con posibles contraseñas que los usuarios podrían estar utilizando. Debido a la diversidad de las contraseñas de los usuarios se podrían tener contraseñas de todo tipo.

La tercera estrategia propuesta es un ataque por diccionario con regla de transformación. Las reglas que se utilizarán son todas las ofrecidas por John. Para la ejecución de esta estrategia se usa el siguiente comando:

john --format=Raw-SHA256 --wordlist=gen_wordlist.lst --rule=all --min-length=4 --max-length=20 <dataset>

john --format=Raw-MD5 --wordlist=gen_wordlist.lst --rule=all --min-length=4 --max-length=20 <dataset>

Nota: El diccionario utilizado es el diccionario generado del script

2.4 Estrategia 4

Se trata de romper las contraseñas de usuarios de la página web de una empresa donde se pueden encontrar clientes, empleados y administradores. La empresa es reciente en el mercado y carece de seguridad. Se tiene un diccionario con posibles contraseñas puesto que nació de una colaboración de 2 empresas pequeñas.

La cuarta estrategia propuesta es un ataque por diccionario con una regla que se ha considerado oportuna para esta práctica, su nombre es 'OneRuleToRuleThemAl'l. Esta regla indica que es una regla conjunta de las mejores que han existido de hob064, best64, T0XICv1, toggles5, InsidePro-PasswordsPro, rockyou-30000, InsidePro-HashManager, d3ad0ne, dive, unix-ninja-leetspeak, generated2, d3adhob0, KoreLogic's, Rockyou50000, _NSAKEY.v2.dive. Los pasos para poder usar esta regla es descargar el fichero que se encuentra en la bibliografía y usar su ruta donde se guarde como regla. El diccionario que se ha optado por usar es el que hemos generado para elegir de manera aleatoria las contraseñas para los datasets además de el conocido diccionario de 'rockyou'.

Para la ejecución de esta estrategia se usa el siguiente comando:

ls wlst | xargs -t -I file john --format=Raw-SHA256 --wordlist=wlst/file --rule /opt/OneRuleToRuleThemAll.rule --min-length=3 <dataset>

ls wlst | xargs -t -I file john --format=Raw-MD5 --wordlist=wlst/file --rule /opt/OneRuleToRuleThemAll.rule --min-length=3 <dataset>

Nota: Para usar varios diccionarios se necesita crear una carpeta que contenga todos los diccionarios, una vez así se puede se puede probar las reglas con cada uno de los diccionarios y romper algunas contraseñas. En este caso la carpeta que contiene los diccionarios es 'wlst'.

2.5 Estrategia 5

Se intenta romper las contraseñas de una app de videojuegos muy común entre adolescentes que contiene métodos de pago que es de interés para robar. Siendo adolescentes la complejidad de las contraseñas es baja y se estima que las contraseñas estarán formadas por letras y números con un mínimo de longitud de 4.

La quinta estrategia es un ataque con máscara que sólo contenga números y letras, como se especifica con una longitud mínima de 4 y se quieren romper con rapidez con lo que se añaden 3 hilos para romperlas lo antes posible. Para la ejecución de esta estrategia se usa el siguiente comando:

john --format=Raw-SHA256
--mask=[A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0
-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9] --min-length=4 --fork=3 <dataset>

john --format=Raw-MD5
--mask=[A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0
-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9][A-Za-z0-9] --min-length=4 --fork=3 <dataset>

2.6 Estrategia 6

Se tiene un caso en el que se intentan romper contraseñas de los usuarios de una consultoría con el que se espera obtener datos de interés. Se estima que el mínimo de caracteres que piden en la aplicación es un mínimo de 3 y un máximo de 15. Se tiene un diccionario con posibles contraseñas que los usuarios podrían utilizar. Debido a la diversidad de las contraseñas de los usuarios se podrían tener contraseñas de todo tipo.

La sexta estrategia propuesta es un ataque por diccionario con regla de transformación. Las reglas que se utilizarán son todas las ofrecidas por John. Para la ejecución de esta estrategia se usa el siguiente comando:

john --format=Raw-SHA256 --wordlist=gen_wordlist.lst --rule=all --min-length=3 --max-length=15 --fork=4 <dataset>

john --format=Raw-MD5 --wordlist=gen_wordlist.lst --rule=all --min-length=3 --max-length=15 --fork=4 <dataset>

Nota: El diccionario utilizado es el diccionario generado del script

3. Análisis de resultados

A continuación se mostrarán los resultados obtenidos de las estrategias definidas. Para esta sección se pide obtener la media y mediana de los tiempos en segundos y el porcentaje de rotura de contraseñas de los dataset. Se recuerda que al optar por el punto extra se tendrán secciones por cada estrategia.

La media y mediana se obtienen de la siguiente manera:

$$\bar{X} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\text{Med}(X) = \begin{cases} X[\frac{n+1}{2}] & \text{if n is odd} \\ \frac{X[\frac{n}{2}]+X[\frac{n}{2}+1]}{2} & \text{if n is even} \end{cases}$$

3.1 Estrategia 1

3.1.1 SHA256

Dataset	Tiempo	Roturas
dataset1_1_SHA256	126	100
dataset1_2_SHA256	600	89
dataset1_3_SHA256	600	97
dataset1_4_SHA256	600	79
dataset1_5_SHA256	600	7
dataset2_1_SHA256	600	86
dataset2_2_SHA256	600	10
dataset2_3_SHA256	600	44
dataset2_4_SHA256	600	12
dataset2_5_SHA256	600	0
dataset3_1_SHA256	10	100
dataset3_2_SHA256	6	100
dataset3_3_SHA256	49	100

dataset3_4_SHA256	50	100
dataset3_5_SHA256	600	44
dataset4_1_SHA256	600	0
dataset4_2_SHA256	600	0
dataset4_3_SHA256	600	0
dataset4_4_SHA256	600	0
dataset4_5_SHA256	600	0
dataset5_1_SHA256	600	45
dataset5_2_SHA256	600	53
dataset5_3_SHA256	600	56
dataset5_4_SHA256	600	51
dataset5_5_SHA256	600	22

Media	Mediana	% Rotura
489.64	600	47.8%

3.1.2 MD5

Dataset	Tiempo	Roturas
dataset1_1_MD5	86	100
dataset1_2_MD5	615	100
dataset1_3_MD5	605	100
dataset1_4_MD5	619	100
dataset1_5_MD5	600	15
dataset2_1_MD5	600	91
dataset2_2_MD5	600	14
dataset2_3_MD5	600	49
dataset2_4_MD5	600	17
dataset2_5_MD5	600	0

dataset3_1_MD5	5	100
dataset3_2_MD5	603	100
dataset3_3_MD5	30	100
dataset3_4_MD5	30	100
dataset3_5_MD5	600	55
dataset4_1_MD5	600	0
dataset4_2_MD5	600	0
dataset4_3_MD5	600	0
dataset4_4_MD5	600	0
dataset4_5_MD5	600	0
dataset5_1_MD5	600	46
dataset5_2_MD5	600	57
dataset5_3_MD5	600	59
dataset5_4_MD5	600	53
dataset5_5_MD5	600	24

Media	Mediana	% Rotura
511.72	600	51.2%

Conclusiones

El uso de una estrategia de fuerza bruta incremental con alfanuméricos supone una gran cantidad de tiempo para poder romper las contraseñas como se puede comprobar con la media y mediana. Además de eso ha demostrado un 'alto' porcentaje de rotura pero no es viable a partir de cierta longitud. Por otro lado, es tanto el trabajo que realiza esta estrategia que ninguna contraseña de los datasets del formato 4 fueron rotas por lo que da a pensar que esta estrategia es buena se puede utilizar si el nivel de complejidad de la contraseña es baja y de baja longitud. También se ha comprobado que SHA256 es un hash mucho más robusto que MD5 puesto que tarda más tiempo en romper las contraseñas.

3.2 Estrategia 2

3.2.1 SHA256

Dataset	Tiempo	Roturas
dataset1_1_SHA256	600	0
dataset1_2_SHA256	2	100
dataset1_3_SHA256	184	100
dataset1_4_SHA256	600	3
dataset1_5_SHA256	600	0
dataset2_1_SHA256	600	0
dataset2_2_SHA256	3	100
dataset2_3_SHA256	311	100
dataset2_4_SHA256	600	0
dataset2_5_SHA256	600	0
dataset3_1_SHA256	10	0
dataset3_2_SHA256	1	100
dataset3_3_SHA256	100	100
dataset3_4_SHA256	50	0
dataset3_5_SHA256	600	0
dataset4_1_SHA256	600	0
dataset4_2_SHA256	3	100
dataset4_3_SHA256	369	100
dataset4_4_SHA256	600	0
dataset4_5_SHA256	600	0
dataset5_1_SHA256	600	0
dataset5_2_SHA256	4	90
dataset5_3_SHA256	344	95
dataset5_4_SHA256	600	0

dataset5_5_SHA256	600	0

Media	Mediana	% Rotura
367.24	600	39.52%

3.2.2 MD5

Dataset	Tiempo	Roturas
dataset1_1_MD5	600	0
dataset1_2_MD5	1	100
dataset1_3_MD5	103	100
dataset1_4_MD5	600	9
dataset1_5_MD5	600	0
dataset2_1_MD5	600	0
dataset2_2_MD5	2	100
dataset2_3_MD5	174	100
dataset2_4_MD5	600	0
dataset2_5_MD5	600	0
dataset3_1_MD5	5	0
dataset3_2_MD5	1	100
dataset3_3_MD5	59	100
dataset3_4_MD5	30	0
dataset3_5_MD5	600	0
dataset4_1_MD5	600	0
dataset4_2_MD5	2	100
dataset4_3_MD5	216	100
dataset4_4_MD5	600	2
dataset4_5_MD5	600	0

dataset5_1_MD5	600	0
dataset5_2_MD5	2	90
dataset5_3_MD5	199	95
dataset5_4_MD5	600	8
dataset5_5_MD5	600	0

Media	Mediana	% Rotura
343.76	600	39.8%

Conclusiones

Lo primero que se observa son los datasets para longitud 3 que como era de esperar ninguna contraseña se ha podido romper puesto que esta estrategia está dirigida a contraseñas con una longitud superior a 3. El uso de esta estrategia con máscara para alfanuméricos supone un alto nivel computacional para poder romper contraseñas y con el tiempo límite establecido no es capaz de romper muchas contraseñas más allá de una longitud 5 por lo que las siguientes requerirá de una gran cantidad de tiempo de para poder romper más o todas las contraseñas puesto que las máscaras suponen probar todas las combinaciones por longitudes. Se mantiene que SHA256 es más robusto que MD5 como se puede comprobar en tiempo de roturas en ciertos datasets.

3.3 Estrategia 3

3.3.1 SHA256

Dataset	Tiempo	Roturas
dataset1_1_SHA256	600	0
dataset1_2_SHA256	538	60
dataset1_3_SHA256	580	12
dataset1_4_SHA256	509	1
dataset1_5_SHA256	541	1
dataset2_1_SHA256	600	0
dataset2_2_SHA256	485	7

dataset2_3_SHA256	507	1
dataset2_4_SHA256	531	0
dataset2_5_SHA256	540	0
dataset3_1_SHA256	600	0
dataset3_2_SHA256	537	79
dataset3_3_SHA256	529	26
dataset3_4_SHA256	533	28
dataset3_5_SHA256	534	2
dataset4_1_SHA256	600	0
dataset4_2_SHA256	504	5
dataset4_3_SHA256	513	0
dataset4_4_SHA256	504	0
dataset4_5_SHA256	505	0
dataset5_1_SHA256	600	0
dataset5_2_SHA256	498	67
dataset5_3_SHA256	495	73
dataset5_4_SHA256	495	67
dataset5_5_SHA256	387	57

Media	Mediana	% Rotura
530.6	531	19.44%

3.3.2 MD5

Dataset	Tiempo	Roturas
dataset1_1_MD5	600	0
dataset1_2_MD5	467	60
dataset1_3_MD5	475	12

dataset1_4_MD5	470	1
dataset1_5_MD5	474	1
dataset2_1_MD5	600	0
dataset2_2_MD5	464	7
dataset2_3_MD5	468	1
dataset2_4_MD5	469	0
dataset2_5_MD5	462	0
dataset3_1_MD5	600	0
dataset3_2_MD5	483	79
dataset3_3_MD5	475	26
dataset3_4_MD5	467	28
dataset3_5_MD5	470	2
dataset4_1_MD5	600	0
dataset4_2_MD5	482	5
dataset4_3_MD5	483	0
dataset4_4_MD5	486	0
dataset4_5_MD5	490	0
dataset5_1_MD5	600	0
dataset5_2_MD5	482	67
dataset5_3_MD5	471	73
dataset5_4_MD5	492	67
dataset5_5_MD5	465	57

Media	Mediana	% Rotura
499.8	475	19.44%

Conclusiones

El uso de diccionario no parece una mala idea cuando se puede saber que son contraseñas potenciales de usuarios en algún lugar y como se puede apreciar más de la mitad de las contraseñas del formato 5 han sido rotas. Por

el contrario, con los demás datasets que se han generado de forma aleatoria no han sido muy efectivos como se puede apreciar en los datasets de alfanuméricos. Por otro lado, cabe resaltar el tiempo requerido para romper datasets de sólo 100 contraseñas con el uso de todas las reglas de transformación que proporciona John The Ripper, lo mejor podría haber sido acotar a las reglas que hubieran resultado ser más relevantes y como resultado se reduciría el tiempo con el inconveniente de que también se redujera la cantidad de contraseñas rotas. Sorprende obtener el mismo porcentaje de rotura para ambos hashes pero es de esperar que SHA256 requiera de mayor tiempo debido a su robustez. Por supuesto, los datasets de contraseñas con longitud 3 no fueron rotas por el límite de longitud.

3.4 Estrategia 4

3.4.1 SHA256

Dataset	Tiempo	Roturas
dataset1_1_SHA256	37	10
dataset1_2_SHA256	38	33
dataset1_3_SHA256	38	2
dataset1_4_SHA256	38	0
dataset1_5_SHA256	38	0
dataset2_1_SHA256	36	12
dataset2_2_SHA256	37	1
dataset2_3_SHA256	37	1
dataset2_4_SHA256	37	0
dataset2_5_SHA256	36	0
dataset3_1_SHA256	37	30
dataset3_2_SHA256	37	67
dataset3_3_SHA256	37	41
dataset3_4_SHA256	37	41
dataset3_5_SHA256	37	6
dataset4_1_SHA256	38	0

dataset4_2_SHA256	38	0
dataset4_3_SHA256	38	0
dataset4_4_SHA256	38	0
dataset4_5_SHA256	38	0
dataset5_1_SHA256	38	27
dataset5_2_SHA256	37	58
dataset5_3_SHA256	37	66
dataset5_4_SHA256	37	63
dataset5_5_SHA256	37	55

Media	Mediana	% Rotura
37.32	37	20.52%

3.4.2 MD5

Dataset	Tiempo	Roturas
dataset1_1_SHA256	35	10
dataset1_2_SHA256	33	33
dataset1_3_SHA256	32	2
dataset1_4_SHA256	33	0
dataset1_5_SHA256	32	0
dataset2_1_SHA256	33	12
dataset2_2_SHA256	33	1
dataset2_3_SHA256	33	1
dataset2_4_SHA256	33	0
dataset2_5_SHA256	33	0
dataset3_1_SHA256	32	30
dataset3_2_SHA256	33	67

dataset3_3_SHA256	33	41
dataset3_4_SHA256	33	41
dataset3_5_SHA256	33	6
dataset4_1_SHA256	33	0
dataset4_2_SHA256	33	0
dataset4_3_SHA256	33	0
dataset4_4_SHA256	33	0
dataset4_5_SHA256	33	0
dataset5_1_SHA256	33	27
dataset5_2_SHA256	33	58
dataset5_3_SHA256	33	66
dataset5_4_SHA256	33	63
dataset5_5_SHA256	33	55

Media	Mediana	% Rotura
32.96	33	20.52%

Conclusiones

Sorprende como en una pequeña fracción de tiempo esta regla es capaz de romper contraseñas en todos los formatos, excepto los alfanuméricos que no ha sido capaz, a partir de sólo dos diccionarios, el generado y 'rockyou', tanto para MD5 y SHA256. Se puede observar con certeza que tiene una mayor efectividad en el quinto formato como se esperaba. Es impresionante el poder de esta regla y que tiene un impresionante funcionamiento como se muestra en la documentación que se buscó y por lo cual fue de nuestro interés. Si se quiere ver dicha documentación acceda al último enlace proporcionado en la bibliografía.

3.5 Estrategia 5

3.5.1 SHA256

Dataset	Tiempo	Roturas
dataset1_1_SHA256	600	0

dataset1_2_SHA256	1	100
dataset1_3_SHA256	35	100
dataset1_4_SHA256	600	17
dataset1_5_SHA256	600	0
dataset2_1_SHA256	600	0
dataset2_2_SHA256	1	100
dataset2_3_SHA256	35	100
dataset2_4_SHA256	600	37
dataset2_5_SHA256	600	0
dataset3_1_SHA256	600	0
dataset3_2_SHA256	1	100
dataset3_3_SHA256	35	100
dataset3_4_SHA256	600	0
dataset3_5_SHA256	600	0
dataset4_1_SHA256	600	0
dataset4_2_SHA256	600	0
dataset4_3_SHA256	600	0
dataset4_4_SHA256	600	0
dataset4_5_SHA256	600	0
dataset5_1_SHA256	600	0
dataset5_2_SHA256	1	75
dataset5_3_SHA256	35	78
dataset5_4_SHA256	600	10
dataset5_5_SHA256	600	0

Media	Mediana	% Rotura
317.76	600	32.68%

3.5.2 MD5

Dataset	Tiempo	Roturas
dataset1_1_MD5	600	0
dataset1_2_MD5	1	100
dataset1_3_MD5	9	100
dataset1_4_MD5	543	100
dataset1_5_MD5	600	0
dataset2_1_MD5	600	0
dataset2_2_MD5	1	100
dataset2_3_MD5	9	100
dataset2_4_MD5	543	100
dataset2_5_MD5	600	0
dataset3_1_MD5	600	0
dataset3_2_MD5	1	100
dataset3_3_MD5	8	100
dataset3_4_MD5	556	100
dataset3_5_MD5	600	0
dataset4_1_MD5	600	0
dataset4_2_MD5	600	0
dataset4_3_MD5	600	0
dataset4_4_MD5	600	0
dataset4_5_MD5	600	0
dataset5_1_MD5	600	0
dataset5_2_MD5	1	75
dataset5_3_MD5	8	78
dataset5_4_MD5	555	71
dataset5_5_MD5	600	0

Media	Mediana	% Rotura
401.4	600	44.96%

Conclusiones

Esta estrategia muestra ser mucho más eficiente que la estrategia 2 al utilizar 3 hilos con el uso de 'fork'. Si se comparan las tablas de ambas estrategias se observa que en ciertos datasets se reduce el tiempo, en aquellos cuya longitud es menor a 6, además de que el número de contraseñas rotas es mucho mayor por lo que es muy recomendable utilizar hilos para facilitar el trabajo. En el caso de los datasets de longitud 3 es completamente imposible puesto que el mínimo es 4 y en los de longitud 7 requiere de mucho más tiempo del establecido del límite para obtener alguna que otra contraseña más. Al ser máscara las combinaciones son inmensas a pesar de esta estrategia estar acotada a letras y números y si fuese el caso de la estrategia 2 el número de roturas sería algo menor y el tiempo aumentaría en poca medida. Al estar acotado a números y letras era de esperar que los datasets del formato 4 no fuesen rotos en ninguna medida al ser alfanuméricos. Se sigue demostrando que romper contraseñas con SHA256 en más complejo que con MD5.

3.6 Estrategia 6

3.6.1 SHA256

Dataset	Tiempo	Roturas
dataset1_1_SHA256	60	100
dataset1_2_SHA256	218	60
dataset1_3_SHA256	218	12
dataset1_4_SHA256	215	1
dataset1_5_SHA256	217	1
dataset2_1_SHA256	211	41
dataset2_2_SHA256	214	7
dataset2_3_SHA256	215	1
dataset2_4_SHA256	216	0
dataset2_5_SHA256	217	0
dataset3_1_SHA256	60	100

dataset3_2_SHA256	222	79
dataset3_3_SHA256	218	26
dataset3_4_SHA256	214	28
dataset3_5_SHA256	219	2
dataset4_1_SHA256	218	25
dataset4_2_SHA256	219	5
dataset4_3_SHA256	217	0
dataset4_4_SHA256	219	0
dataset4_5_SHA256	221	0
dataset5_1_SHA256	215	90
dataset5_2_SHA256	218	77
dataset5_3_SHA256	219	78
dataset5_4_SHA256	219	69
dataset5_5_SHA256	214	58

Media	Mediana	% Rotura
204.52	217	34.40%

3.6.2 MD5

Dataset	Tiempo	Roturas
dataset1_1_MD5	59	100
dataset1_2_MD5	130	60
dataset1_3_MD5	133	12
dataset1_4_MD5	133	1
dataset1_5_MD5	131	1
dataset2_1_MD5	133	41
dataset2_2_MD5	132	7

dataset2_3_MD5	132	1
dataset2_4_MD5	134	0
dataset2_5_MD5	133	0
dataset3_1_MD5	60	100
dataset3_2_MD5	134	79
dataset3_3_MD5	135	26
dataset3_4_MD5	133	28
dataset3_5_MD5	135	2
dataset4_1_MD5	141	25
dataset4_2_MD5	133	5
dataset4_3_MD5	132	0
dataset4_4_MD5	131	0
dataset4_5_MD5	134	0
dataset5_1_MD5	132	90
dataset5_2_MD5	133	77
dataset5_3_MD5	134	78
dataset5_4_MD5	134	69
dataset5_5_MD5	133	58

Media	Mediana	% Rotura
127.36	133	34.40%

Conclusiones

Esta estrategia muestra ser mucho más eficiente que la estrategia 3 al utilizar 4 hilos con el uso del 'fork' y a la reducción de la longitud mínima y máxima. Si se comparan las tablas de ambas estrategias se puede observar que se reduce el tiempo en todos los datasets, al mismo tiempo el índice de rotura de esta estrategía es mayor. Debido a la reducción de la longitud mínima esta estrategía es capaz de romper los datasets de 3 palabras. Tal y como se puede apreciar en las tablas superiores, en los datasets que introducen caracteres alfanuméricos de forma aleatoria la estrategía reduce drásticamente su efectividad rompiendo en muchos casos una o ninguna contraseña independientemente de la longitud del dataset. Sorprende obtener el mismo

porcentaje de rotura para ambos hashes pero es de esperar que SHA256 requiera de mayor tiempo debido a su robustez.

4. Problemas

A la hora de la realización de la práctica se han encontrado numerosos problemas. El primer problema que se encontró en el momento de comenzar este laboratorio fue la instalación de John The Ripper, ya que al utilizar el comando 'sudo apt-get install john' no se instalaba la última versión del mismo por más que se realizarán updates desde la terminal. Debido a eso se tuvo que realizar la instalación manualmente descargando los archivos en la página oficial.

Otros problemas que se han tenido a la hora de realizar la práctica han sido a la hora de realizar las estrategias de la misma, en un principio se quiso realizar estrategias con algunas reglas externas a John que se habían encontrado navegando por la web en repositorios de Github y que tras analizarlas parecieron ser interesantes, pero tras probar de todas las formas que se comentaban para hacer uso de ellas no se consiguió que funcionasen en absoluto, lo cual dió lugar a la falta de información y guías en la web. La cantidad de información que se puede encontrar navegando por internet parece ser un poco limitada, y la mayoría de páginas únicamente enseñan a instalarlo o a realizar los comandos básicos para aprender a utilizar esta herramienta. A pesar de la guía que se ha proporcionado ha sido un poco complicado el uso de algunos de estos comandos.

5. Conclusiones

Desde un punto de vista de un atacante es más sencillo atacar contraseñas con poco longitud utilizando máscaras o diccionarios con reglas de transformación con el uso de hilos. Los usuarios más débiles son aquellos que tienen contraseñas cortas y que sólo están compuestos por letras o números a las que se denominan como contraseñas débiles. En cambio aquellas contraseñas que son alfanuméricas son las denominadas como fuertes. Por supuesto la complejidad de estas aumentará independientemente de cómo sean si se aumenta en gran medida la longitud de estas, además del tipo hash que se utilice sobre estas supone mucho mayor o menor esfuerzo para romper las contraseñas. Las estrategias que resultan más interesantes a un atacante son como se ha mencionado, el uso de máscaras y reglas de transformación en diccionarios.

Desde un punto de vista de un usuario, que es un objetivo potencial, se ha concluido que se deben tener contraseñas que supongan una gran dificultad para romper incluso si es difícil de recordar para la persona misma. Es importante proteger aquello que se posee, para ello una contraseña robusta y grande supone un alto grado de seguridad. Dicho esto es muy recomendable utilizar un gestor de contraseñas que genere contraseñas de alta complejidad para los usuarios y las guarde de manera efectiva.

6. Bibliografía

https://github.com/nyxgeek/nyxgeek-rules/tree/master/john-rules
https://github.com/NotSoSecure/password_cracking_rules
https://notsosecure.com/one-rule-to-rule-them-all