Comparativa

David Pazán

18 de junio del 2022

1 Tabla comparativa

1.1 Ingreso de texto

Para la realización de esta tabla comparativa se hace uso de la palabra **mequieropegaruntiromeudeus**, y a continuación se hace despliegue de los resultados obtenidos:

Hash	Valor Hash
HashLab	¡H8YHfWFm5Qs" TGAke6¡TUFswzyzCHOX?9oFY7uT8B$_r p.v' Ji - Il$
MD5	03cdd31b5b237b641716ba6219a94508
SHA1	2a237a432a9d5d05cf39c12a9c315ac926e6bc09
SHA256	a1b86bc009d6c5d33593b82ae0c99a39c161cd19171480aad75147921078d03e

Hash	Tiempo	Largo	Base	Entropia
HashLab	0.001092672348022461	55	74	341.5199351095923
MD5	2.193450927734375e-05	32	16	128.0
SHA1	1.0251998901367188e-05	40	16	160.0
SHA256	5.7220458984375e-06	64	16	256.0

Como se puede observar gracias a la tabla anterior obtenida, la entropía que se presenta para los distintos *Hash* estudiados, el realizado en esta experiencia presenta una gran diferencia con respecto a los otros, dando a entender que ofrecemos mayor seguridad. Mientras que en los usados por la comunidad, el SHA256 presenta mayor entropía con respecto a sus pares, debido al largo del hash. Por lo cual, se saca la conclusión de que se debe alargar los hash, para obtener una mejor entropia.
De esto también se puede indicar que no porque la entropía sea alta el rendimiento igual va a ser bueno, por como se ha de poder observar en la tabla de a continuación para el rendimiento de exigencia.

1.2 Selección de archivo

Para la realización de esta tabla comparativa se hace uso de los distintos archivos *1.txt, 10.txt, 20.txt, 50.txt* para las entradas de archivos

Archivo	HashLab	MD5	SHA1	SHA256
1.txt	0.004302978515625	0.00033783912658691	0.00018835067749023	0.00018024444580078
10.txt	0.0059909820556	0.000484466552	0.0004630088806	0.0004506111145
20.txt	0.008331060409545	0.0005633831024169	0.0005288124084472	0.0005640983581542
50.txt	0.02671957015991	0.003726482391357	0.003785848617553	0.03624081611633

Como se aprecia respecto a los tiempos, el hash realizado para la experiencia, siempre tuvo mayor tiempo con respecto a la competencia. Una de las ideas del porque ocurre esto, puede ser debido al gran diccionario que hace uso, recordando que en la tabla anterior se presenta este dato, haciendo operaciones de validación sobre los carácteres definidos y que son posibles de usar.

2 Conclusión final

Claramente los hash son mas conocidos y mas utilizados debido a su entropía y velocidad debido a la conversión que tienen y trabajan con bytes con funciones insertas en Python, en cambio el hash creado trabaja con operaciones mas lentas que los cambios de bytes como multiplicaciones y ciclos anidados además que el hash propio trabaja con los valores de conversión entre números y caracteres que vienen de un diccionario ya creado comparando cifras, lo que se traduce en más tiempo para la ejecución, por lo que es recomendable utilizar SHA256 que es la de mejor rendimiento