

14.12.2023

Курс:

Практическая работа к уроку № Lesson_3

--

Криптографические хеш-функции



Задание_1:

Необходимо для каждой строки получить хеши с помощью алгоритмов: sha256, bcrypt (с солью \$2b\$15\$NSVH/I.9u1l/WoYUd/sSI.) и md5.

```
08122988399  nampoly2537  lasabre97  as534031  Victor_  16MSTF68AYSL  hhrules
cpt704242  gracemac  rayas123123
```

Для выполнения задания используйте библиотеки вашего языка программирования (например, для Python это будут hashlib и bcrypt).

Пример:

```
(kali@kali)-[~]
└─$ locate wordlists
/etc/theHarvester/wordlists
/etc/theHarvester/wordlists/dns-big.txt
...
/usr/share/wordlists/rockyou.txt.gz

(kali@kali)-[~/tmp]
└─$ echo -n 'test' | md5sum
098f6bcd4621d373cade4e832627b4f6  -
```

```
(kali㉿kali)-[~/tmp]
└─$ echo -n 'test' | shasum -a 256
9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08 -
(kali㉿kali)-[~/tmp]
└─$ python3
Python 3.11.6 (main, Oct 8 2023, 05:06:43) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import hashlib
>>> hashlib.sha256(b'test').hexdigest()
'9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08'
```

```
(kali㉿kali)-[~/tmp]
└─$ hashcat -m 0 098f6bcd4621d373cade4e832627b4f6 rockyou.txt
hashcat (v6.2.6) starting
...
```

```
(kali㉿kali)-[~/tmp]
└─$ export SALT1='salt'
(kali㉿kali)-[~/tmp]
└─$ export SALT2='sugar'
(kali㉿kali)-[~/tmp]
└─$ echo -n 'iloveyou$SALT1' | shasum -a 256
174aaef2f3430ef0d39dba782dd3d71bf20dcb2b1354664c9304e721f76f3b33 -
(kali㉿kali)-[~/tmp]
└─$ echo -n 'iloveyou$SALT2' | shasum -a 256
abccb2ffd5807c8e98f28e499388e9c0ae3bf8e1ffcdd36270f631baafc1961b -
```

```
(kali㉿kali)-[~/tmp]
└─$ python 31.py 16
True
(kali㉿kali)-[~/tmp]
└─$ cat 31.py
import bcrypt
import sys
cost_param = int(sys.argv[1])
password = b'super secret password'
hashed = bcrypt.hashpw(password, bcrypt.gensalt(cost_param))
print(bcrypt.checkpw(password, hashed))
```

```
(kali㉿kali)-[~/tmp]
└─$ time 31.py 16
31.py: command not found
real    0.21s
user    0.06s
sys      0.08s
cpu      71%
```

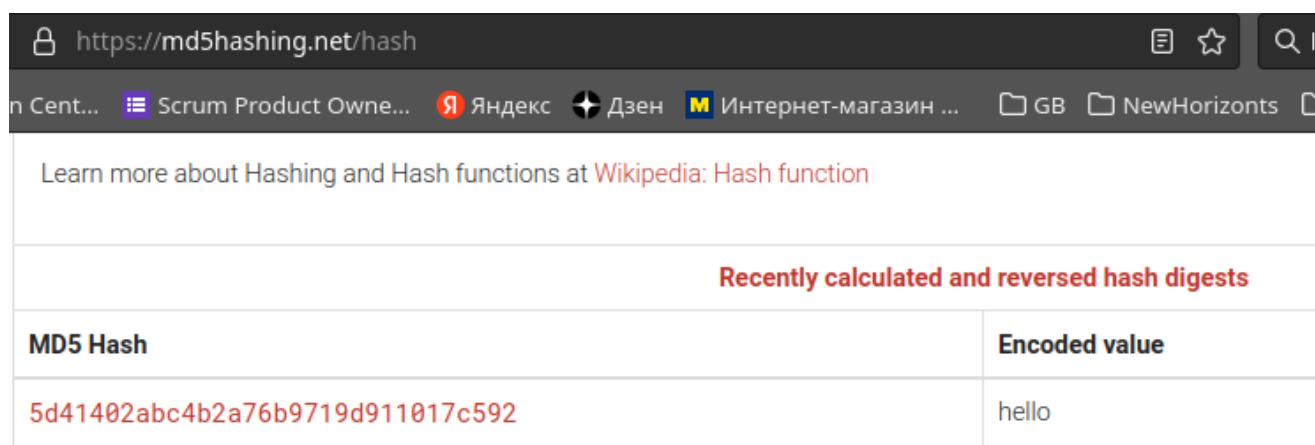
```
(kali㉿kali)-[~/tmp]
└─$ export SALT1='$2b$15$NSVH/I.9u1l/WoYUd/sSI'
(kali㉿kali)-[~/tmp]
└─$ echo -n '08122988399$SALT1' | shasum -a 256
cfc5b15a75ca96735523fc2f2bc6b3ae1f862072a80cdb911898e439be3e508e -
```

Задание 2

Найдите исходные значения каждого из хешей:

```
`23f6121371ef7563ec479de345c9e479
7351a8f9d143de67724772ea1a7e7e82d068cb77c3f495d5d234083d686c1226
cf9baae799c1d9164d2b6c37acf8562d
ae6a4840fe4e29991483c8a4115451d105e2e057e6870279c15c211d3375546f    004a602d9e890b00
9407cad0b776fd5f614649eddf47eefc71aa82d    e0728635632956dd3d28e0b9c28bf97735de32c3
3bb0d8c10c4ce5561daa0d82ee65d692a9d70276f7e53006aa5c4ad34aee1c13
9f00e35c7729cdac242e86051a57020d
6650ee0bf5af2a915618beb0361e67c04cedd4ea0c00d68ea72b16866c62dd16
eead40c59c83c69bdd2505ac081e09c0354345e2    7d63b6cbbbed434353836dc011701321ac4827b6d
f1df20f71e680534    8a99ad26f053411a73104bfe0f1e19193efd5587eefd84bda435ed92397850d1
78dd2b61c96e895035c87528b5c3e2a9    b3dab40f3d51065754e4f0ef0d20039d
c9203499e3a2ff2b1244913777a3614954c1cd17240695fc160fe96914094912
8e82dfe15d36d6ec75f38de62128cf386964bf80b423ccda4fd239fc942d2cd9    43ca9303ecf07627
3ad12414555b1e01870647cb4dc99df88dbdd7eaf1494900fa9dd89bb2758cb3`
```

Для выполнения задания разрешается использовать любые методы.



The screenshot shows a web browser at the URL `https://md5hashing.net/hash`. Below the address bar, there are search results for "Hashing and Hash functions" with a link to Wikipedia. A table titled "Recently calculated and reversed hash digests" contains the following data:

MD5 Hash	Encoded value
5d41402abc4b2a76b9719d911017c592	hello

Задание 3

В файле `leak_db.txt` находятся хеши паролей из скомпрометированной базы данных. Вам необходимо восстановить все пароли из базы данных. Известно, что пароли словарные.

Выводы:

SHA2 — международный стандарт криптографических функций.

Стрибог — отечественный стандарт криптографических функций.

SHA3 — относительно новая криптографическая функция, использующая механизм Sponge (губка), по сути, не нарезает входные данные на блоки и смешивает их, а идёт дальше, после нарезки и сбора она может сделать хеш произвольной длины. То есть запрашивается дайджест не фиксированной длины, как у SHA2 и Стрибог, а произвольной.

Space Time trade-off — подготовительные вычисления до того, как провести атаку, используется пространство, в данном случае на жёстком диске или в оперативной памяти (хеши вычисляются заранее).

Hashcat — программа для перебора хешей.

Crackstation.net — сайт, на котором можно ввести хеш, и, если в базе есть открытый текст, из

которого он был взят, — получить результат.

Соль — некоторая строка, которую добавляют к паролю, чтобы его хеш не совпадал с хешем точно такого же пароля на другом сайте.

Bcrypt — специализированное решение для хранения паролей.

Argon2, Scrypt — функции, предназначенные для замедления вычисления хешей.

Вся информация в данной работе представлена исключительно в ознакомительных целях!

Любое использование на практике без согласования тестирования подпадает под действие УК РФ.

- <https://gb.ru>

*Выполнил: ==AndreiM