

- Práctica voluntaria Ejercicios de tema 3: Seguridad lógica
- Añadido: crackeando con hashcat

# Práctica voluntaria Ejercicios de tema 3: Seguridad lógica

En esta práctica vamos a desencriptar las contraseñas de una máquina linux. Para ello lo vamos a hacer con la herramienta john the ripper (de aquí en adelante 'john').

Vamos a ver la diferencia entre usar el john que te descargas directamente desde los repositorios con respecto a descargarnoslo de la página oficial y compilarlo cambiando alguna cosa.

Metemos los hashes en un archivo

```
athos@athos-virtual-machine ~ $ cat pass.practica.charo.voluntaria
alumno1:$1$zmDCo$P/Rrln2jTy30eTvjl8Mg0:14544:0:99999:7:::
root:$1$bM36INXG$nlckzvSVJy.z42Atf5p6n.:11585:0:99999:7:::
```

ejecutamos el comando `john pass.practica.charo.voluntaria`

```
athos@athos-virtual-machine ~ $ john pass.practica.charo.voluntaria
Loaded 2 password hashes with 2 different salts (md5crypt [MD5 32/64 X2])
Press 'q' or Ctrl-C to abort, almost any other key for status
alumno (alumno1)
█
```

Si pulsamos cualquier tecla menos la `q` o `ctrl+c` veremos el estado por el que va el crackeo

```
athos@athos-virtual-machine ~ $ cat pass.practica.charo.voluntaria
alumno1:$1$zmDCo$P/Rrln2jTy30eTvjl8Mg0:14544:0:99999:7:::
root:$1$bM36INXG$nlckzvSVJy.z42Atf5p6n.:11585:0:99999:7:::
athos@athos-virtual-machine ~ $ john pass.practica.charo.voluntaria
Loaded 2 password hashes with 2 different salts (md5crypt [MD5 32/64 X2])
Press 'q' or Ctrl-C to abort, almost any other key for status
alumno (alumno1)
lg 0:00:03:49 3/3 0.004348g/s 11527p/s 11527c/s 11527C/s mindor13..mindor11
lg 0:00:03:53 3/3 0.004273g/s 11533p/s 11533c/s 11533C/s lilld12..lilld13
lg 0:00:03:54 3/3 0.004255g/s 11535p/s 11535c/s 11535C/s leorge2..leorgod
█
```

Como vemos que va a tardar mucho, vamos a descargar un diccionario y probaremos con él, aunque como podemos ver, nos ha sacado una de las dos contraseñas

```
athos@athos-virtual-machine ~ $ cat pass.practica.charo.voluntaria
alumno1:$1$zmDCospP/RrlN2jTy30eTvjl8Mg0:14544:0:99999:7:::
root:$1$bM36iNXG$nlckzvSVJy.z42Atf5p6n.:11585:0:99999:7:::
athos@athos-virtual-machine ~ $ john pass.practica.charo.voluntaria
Loaded 2 password hashes with 2 different salts (md5crypt [MD5 32/64 X2])
Press 'q' or Ctrl-C to abort, almost any other key for status
alumno
(alumno)
1g 0:00:03:49 3/3 0.004348g/s 11527p/s 11527c/s 11527C/s mindor13..mindor11
1g 0:00:03:53 3/3 0.004273g/s 11533p/s 11533c/s 11533C/s lilld12..lilld13
1g 0:00:03:54 3/3 0.004255g/s 11535p/s 11535c/s 11535C/s leorge2..leorgod
1g 0:00:06:19 3/3 0.002631g/s 11653p/s 11653c/s 11653C/s 092kje..092kjr
1g 0:00:11:49 3/3 0.001408g/s 11829p/s 11829c/s 11829C/s tr2g2z..tr2ga2
1g 0:00:18:12 3/3 0.000914g/s 11881p/s 11881c/s 11881C/s dcdp0p..dcdp0d
1g 0:00:18:16 3/3 0.000911g/s 11883p/s 11883c/s 11883C/s p03m39..p03m3m
1g 0:00:23:20 3/3 0.000714g/s 11681p/s 11681c/s 11681C/s 111ml..111rl
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
athos@athos-virtual-machine ~ $ john pass.practica.charo.voluntaria --show
alumno1:alumno:14544:0:99999:7:::

1 password hash cracked, 1 left
athos@athos-virtual-machine ~ $
```

Vamos a intentar sacar la otra con ataque de diccionario.

Passwords - SkullSecurity - Mozilla Firefox

https://wiki.skullsecurity.org/Passwords

These are dictionaries that come with tools/worms/etc, designed for cracking passwords. As far as I know, I'm not breaking any licensing agreements by mirroring them with credit; if you don't want me to host one of these files, let me know and I'll remove it.

Name	Compressed	Uncompressed	Notes
John the Ripper	John.txt.bz2 (10,934 bytes)	n/a	Simple, extremely good, designed to be modified
Cain & Abel	cain.txt.bz2 (1,069,968 bytes)	n/a	Fairly comprehensive, not ordered
Conficker worm	conficker.txt.bz2 (1411 bytes)	n/a	Used by conficker worm to spread - low quality
500 worst passwords	500-worst-passwords.txt.bz2 (1868 bytes)	n/a	
370 Banned Twitter passwords	twitter-banned.txt.bz2 (1509 bytes)	n/a	

### Leaked passwords

Passwords that were leaked or stolen from sites. I'm hosting them because it seems like nobody else does (hopefully it isn't because hosting them is illegal :)). Naturally, I'm not the one who stole these; I simply found them online, removed any names/email addresses/etc (I don't see any reason to supply usernames -- If you do have a good reason, email me (ron-at-skullsecurity.net) and I'll see if I have them.

The best use of these is to generate or test password lists.

Note: The dates are approximate.

Name	Compressed	Uncompressed	Date	Notes
Rockyou	rockyou.txt.bz2 (60,498,886 bytes)	n/a		
Rockyou with count	rockyou-withcount.txt.bz2 (59,500,255 bytes)	n/a	2009-12	Best list available; huge, stolen unencrypted
phpb	phpb.txt.bz2 (868,606 bytes)	n/a		Ordered by commonness
phpb with count	phpb-withcount.txt.bz2 (872,867 bytes)	n/a	2009-01	Cracked from md5 by Brandon Enright (97%+ coverage)
phpb with md5	phpb-withmd5.txt.bz2 (4,117,887 bytes)	n/a		
MySpace	myspace.txt.bz2 (175,970 bytes)	n/a		
MySpace - with count	myspace-withcount.txt.bz2 (179,929 bytes)	n/a	2006-10	Captured via phishing
Hotmail	hotmail.txt.bz2 (47,195 bytes)	n/a		
Hotmail with count	hotmail-withcount.txt.bz2 (47,975 bytes)	n/a		Unknown isn't clearly understood how these were stolen

Descargamos el rockyou y podemos comprobar que tiene mas de 14 millones de contraseñas

```
athos@athos-virtual-machine ~  
athos@athos-virtual-machine ~ 161x45  
athos@athos-virtual-machine ~ $ ls  
Desktop Downloads hashcat-4.0.1 pass.practica.charo.voluntaria Postman Public Templates  
Documents github Music Pictures pruebaguid rockyou.txt rockyou.txt Videos  
athos@athos-virtual-machine ~ $ wc -l rockyou.txt  
14344391 rockyou.txt  
athos@athos-virtual-machine ~ $
```

Y ejecutaremos john de la siguiente forma

```
john --wordlist=rockyou.txt pass.practica.charo.voluntaria
```

```
athos@athos-virtual-machine ~ $ john --wordlist=rockyou.txt pass.practica.charo.voluntaria  
Loaded 2 password hashes with 2 different salts (md5crypt [MD5 32/64 X2])  
Remaining 1 password hash  
Press 'q' or Ctrl-C to abort, almost any other key for status  
0g 0:00:00:08 0% 0g/s 10918p/s 10918c/s 10918C/s yomamal23..yidarmy
```

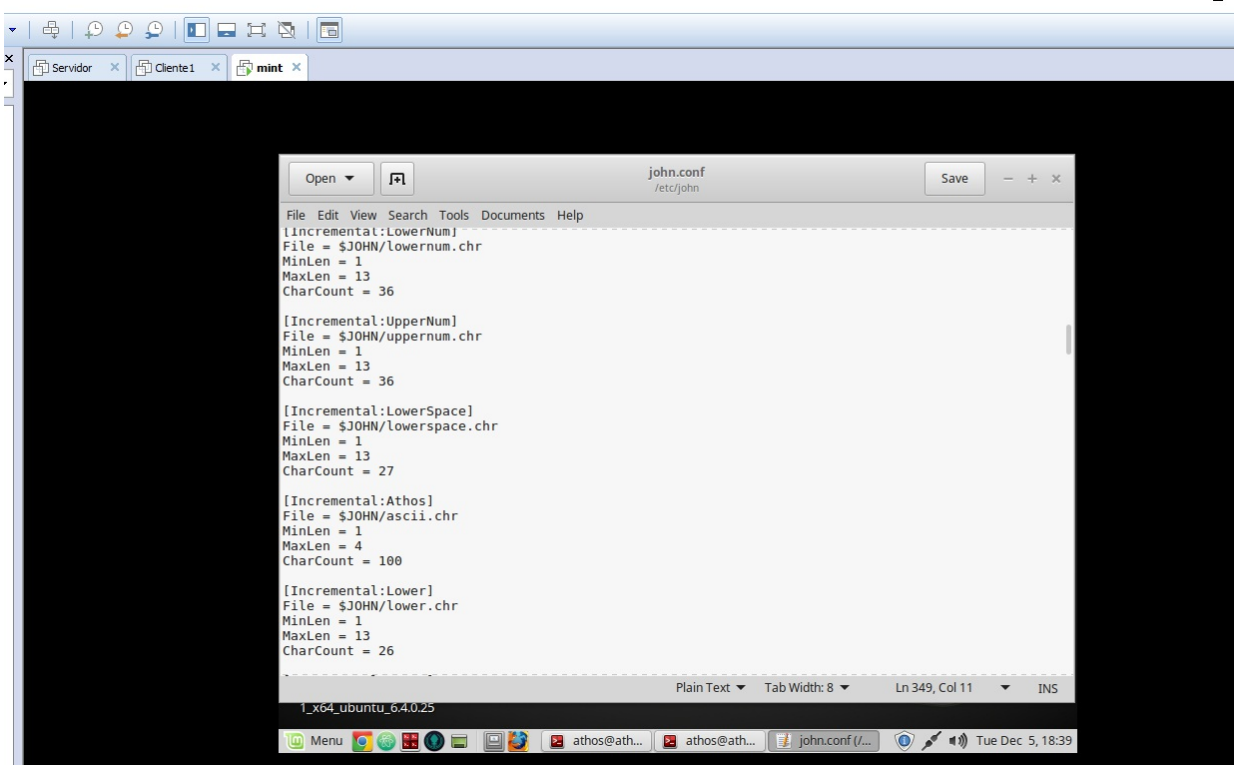
Como vemos en la imagen anterior, john nos guarda la contraseña que sacó anteriormente, si quisieramos verla tendríamos que usar el parametro `--show`

No hemos tenido suerte, no la ha sacado, tendríamos que probar con otros diccionarios o de forma incremental.

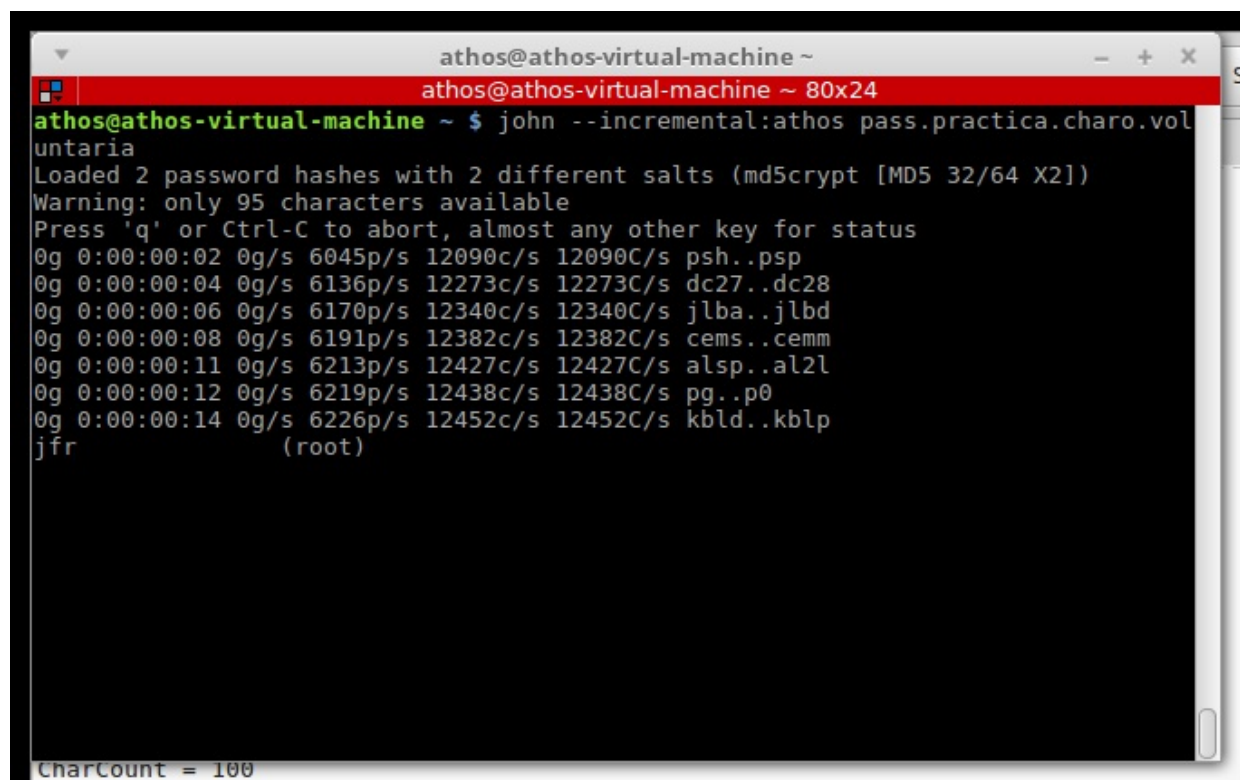
```
athos@athos-virtual-machine ~  
athos@athos-virtual-machine ~ 161x45  
athos@athos-virtual-machine ~ $ ls  
Desktop Downloads hashcat-4.0.1 pass.practica.charo.voluntaria Postman Public Templates  
Documents github Music Pictures pruebaguid rockyou.txt rockyou.txt Videos  
athos@athos-virtual-machine ~ $ wc -l rockyou.txt  
14344391 rockyou.txt  
athos@athos-virtual-machine ~ $ john --wordlist=rockyou.txt pass.practica.charo.voluntaria  
Loaded 2 password hashes with 2 different salts (md5crypt [MD5 32/64 X2])  
Remaining 1 password hash  
Press 'q' or Ctrl-C to abort, almost any other key for status  
0g 0:00:00:08 0% 0g/s 10918p/s 10918c/s 10918C/s yomamal23..yidarmy  
0g 0:00:00:43 2% 0g/s 10620p/s 10620c/s 10620C/s warhammer5..warhammer4  
0g 0:00:00:46 2% 0g/s 10653p/s 10653c/s 10653C/s luvweed..luvumuch  
0g 0:00:01:37 6% 0g/s 10446p/s 10446c/s 10446C/s beatbox1..beatarmy  
0g 0:00:01:58 7% 0g/s 10585p/s 10585c/s 10585C/s superluna..superluisa  
0g 0:00:02:40 10% 0g/s 10744p/s 10744c/s 10744C/s ilovegj..ilovegiz  
0g 0:00:04:10 17% 0g/s 10905p/s 10905c/s 10905C/s wmb2116..wmb202062  
0g 0:00:05:05 21% 0g/s 10967p/s 10967c/s 10967C/s taylakane..taylajill  
0g 0:00:06:30 28% 0g/s 11025p/s 11025c/s 11025C/s renolboo..renolaguna  
0g 0:00:13:37 63% 0g/s 11116p/s 11116c/s 11116C/s chuvariariwariw..chuwap  
0g 0:00:21:16 100% 0g/s 11237p/s 11237c/s 11237C/s Vamos!  
Session completed  
athos@athos-virtual-machine ~ $ john --wordlist=rockyou.txt pass.practica.charo.voluntaria --show  
Invalid options combination or duplicate option: "--show"  
athos@athos-virtual-machine ~ $ john --show  
Password files required, but none specified  
athos@athos-virtual-machine ~ $ john --show pass.practica.charo.voluntaria  
alumno1:alumno:14544:0:99999:7:::  
1 password hash cracked, 1 left  
athos@athos-virtual-machine ~ $
```

Editamos el archivo de configuración de john para añadir un nuevo modo incremental, en el que le

vamos a decir que busque contraseñas solo de 1 a 4 caracteres



Ahora ejecutamos john con nuestro nuevo modo incremental.



Y como vemos en la imagen anterior conseguimos la contraseña.

Si lo quisiéramos hacer con hashcat, sería de la siguiente manera:



```
athos@athos-virtual-machine ~  
athos@athos-virtual-machine ~ 98x35  
athos@athos-virtual-machine ~ $ ./hashcat-4.0.1/hashcat64.bin -i -0 -m 500 -a 3 -1 ?u?l prueba.pas  
s ?1?1?1?1 --outfile=testresult.txt
```

Esto nos guardara la contraseña en el archivo `testresult.txt` . Lo ejecutamos:

En la siguiente imagen, nos podemos fijar, en que `guess-queue` es el 'paso' por el que va. Como hemos puesto que la longitud máxima sea 4 caracteres ( `?1?1?1?1` -> cada `?1` representa un caracter con charset `?l?d` (es decir, letras mayusculas y minusculas)) está ya haciendo el cálculo de 3 caracteres, y vemos que ha sacado 1 de las dos contraseñas como podemos apreciar en `Recovered` . Si vamos al archivo que pusimos de output ( `testresult.txt` ) veremos que tenemos la contraseña.

```
athos@athos-virtual-machine ~  
athos@athos-virtual-machine ~ 98x35  
Hash.Target.....: prueba.pass  
Time.Started.....: Tue Dec 5 19:45:34 2017 (2 secs)  
Time.Estimated....: Tue Dec 5 19:45:40 2017 (4 secs)  
Guess.Mask.....: ?1?1?1 [3]  
Guess.Charset....: -1 ?u?l, -2 Undefined, -3 Undefined, -4 Undefined  
Guess.Queue.....: 3/4 (75.00%)  
Speed.Dev.#1.....: 19600 H/s (6.39ms)  
Recovered.....: 1/2 (50.00%) Digests, 1/2 (50.00%) Salts  
Progress.....: 82560/281216 (29.36%)  
Rejected.....: 0/82560 (0.00%)  
Restore.Point....: 640/2704 (23.67%)  
Candidates.#1....: Bdc -> BLJ  
HWMon.Dev.#1.....: N/A  
  
Approaching final keyspace - workload adjusted.  
  
Session.....: hashcat  
Status.....: Exhausted  
Hash.Type.....: md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)  
Hash.Target.....: prueba.pass  
Time.Started.....: Tue Dec 5 19:45:34 2017 (7 secs)  
Time.Estimated....: Tue Dec 5 19:45:41 2017 (0 secs)  
Guess.Mask.....: ?1?1?1 [3]  
Guess.Charset....: -1 ?u?l, -2 Undefined, -3 Undefined, -4 Undefined  
Guess.Queue.....: 3/4 (75.00%)  
Speed.Dev.#1.....: 19109 H/s (1.56ms)  
Recovered.....: 1/2 (50.00%) Digests, 1/2 (50.00%) Salts  
Progress.....: 281216/281216 (100.00%)  
Rejected.....: 0/281216 (0.00%)  
Restore.Point....: 2704/2704 (100.00%)  
Candidates.#1....: XtV -> XQz  
HWMon.Dev.#1.....: N/A  
  
Linux Mint 18.2 Sonya  
Menu [Icons] [Taskbar] [System Tray] athos@ath... athos@ath... john.conf (/... Tue Dec 5, 19:45
```

Archivo `testresult.txt` :

```
athos@athos-virtual-machine ~
athos@athos-virtual-machine ~ 80x24
athos@athos-virtual-machine ~ $ cat tes
testresult.txt
athos@athos-virtual-machine ~ $ cat testresult.txt
$1$bM36INXG$nlckzvSVJy.z42Atf5p6n.:jfr
athos@athos-virtual-machine ~ $
```

## Añadido: crackeando con hashcat

Lo primero que vamos a hacer es descargarnos hashcat desde la página oficial. Debemos prestar especial atención a los requisitos (los drivers de la tarjeta gráfica), ya que una mala configuración hará que el rendimiento sea muchísimo menor.

### Descargar hashcat

The screenshot shows the official hashcat website. A red box highlights the 'hashcat\_binaries' download link in the 'Download' table. Another red box highlights the 'GPU Driver requirements' section, with red arrows pointing from it to a box labeled 'IMPORTANTE'.

Name	Version	Date	Download	Signature
hashcat_binaries	v4.0.1	2017.11.07	<a href="#">Download</a>	<a href="#">PGP</a>
hashcat_sources	v4.0.1	2017.11.07	<a href="#">Download</a>	<a href="#">PGP</a>

Signing key on PGP keyservers: RSA, 2048-bit, Key ID: 2048R/8A16544F, Fingerprint: A708 3322 9004 0B41 99CC 0052 3C17 DA8B 8A16 544F

Check out our [GitHub Repository](#) for the latest development version

**GPU Driver requirements:**

- AMD GPUs on Linux require "RadeonOpenCompute (ROCm)" Software Platform (1.6.100 or later)
- AMD GPUs on Windows require "AMD Radeon Software Crimson Edition" (15.12 or later)
- Intel CPUs require "OpenCL Runtime for Intel Core and Intel Xeon Processors" (16.1.1 or later)
- Intel GPUs on Linux require "OpenCL 2.0 GPU Driver Package for Linux" (2.0 or later)
- Intel GPUs on Windows require "OpenCL Driver for Intel Iris and Intel HD Graphics"
- NVIDIA GPUs require "NVIDIA Driver" (367.x or later)

**Features**

- World's fastest password cracker
- World's first and only in-kernel rule engine
- Free
- Open-Source (MIT License)
- Multi-OS (Linux, Windows and macOS)
- Multi-Platform (CPU, GPU, DSP, FPGA, etc., everything that comes with an OpenCL runtime)
- Multi-Hash (Cracking multiple hashes at the same time)
- Multi-Devices (Utilizing multiple devices in same system)
- Multi-Device-Types (Utilizing mixed device types in same system)
- Supports distributed cracking networks (using overlay)
- Supports interactive pause / resume
- Supports sessions
- Supports restore
- Supports reading password candidates from file and stdin
- Supports hex-salt and hex-charset
- Supports automatic performance tuning
- Supports automatic keyspace ordering markov-chains
- Built-in benchmarking system
- Integrated thermal watchdog
- 200+ Hash-Types implemented with performance in mind
- ... and much more

**Screenshot**

Ya podríamos usar hashcat, pero vamos a descargarnos hashcat-gui para no tener que ejecutar los comandos desde consola.

### Descargar hashcat-gui

hashcat - advanced password | Hashcat GUI - HashKiller.co.uk

https://hashkiller.co.uk/hashcat-gui.aspx

# ONLY REAL PASSWORDS OF USERS!

Home Forums Decrypter / Cracker Database Info Hash Min Max WPA Crack Lists and Competition Contest Tools Hashcat GUI

Hashcat GUI **HashcatGUI\_1.00r3.zip - v1.00r3 [6.33MB]**

**Included**

- hashcat-utils-1.0
- cap2hccap

**Requirements:**

- OS: Windows Only
- dotNET Framework: v4
- **hashcat 3.00**

Your folder structure should look like the image to the right...

Extract the hashcat apps into folders within the Hashcat GUI directory.

- cap2hccap
- conf\_app
- conf\_dict
- hashcat-0.42
- hashcat-utils-1.0
- oclHashcat-lite-0.13
- oclHashcat-plus-0.12
- rules
- App.HashcatGUI.exe
- fingerprint\_readme.txt
- help.txt
- merge.txt
- 32\_hex\_split\_into\_16\_hex\_LM.cmd
- fingerprint\_auto.cmd

**HashCat GUI Tutorial**

Hashcat GUI Tutorial

Hash File: F:\hashcat\hashes\green\_md5.lst

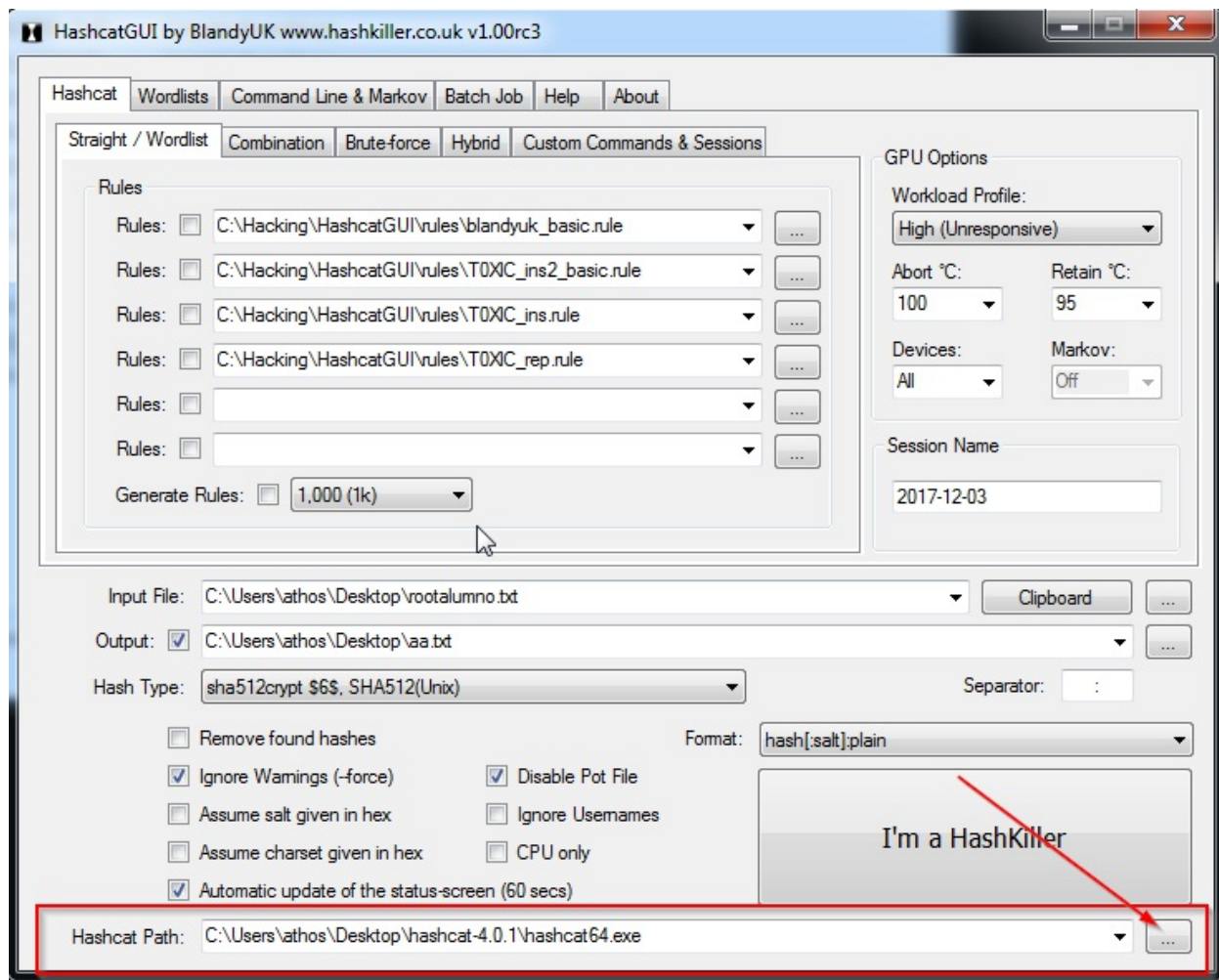
Hashes: 7915526 total, 1 unique salts, 7915526 unique digests  
Bitmaps: 21 bits, 1048576 entries, 0x000fffff mask, 4194304 bytes  
Workload: 250 loops, 32 accel  
Watchdog: Temperature abort trigger disabled  
Watchdog: Temperature retain trigger disabled  
Device #1: Turks, 972MHz, 6400  
Device #2: Cypress, 972MHz, 18000  
Device #1: Kernel ./kernels/4098/m0000\_al.Turks\_1268.1\_1268.1 (VM).kernel not found in cache! Building may take a while...  
Device #1: Kernel ./kernels/4098/m0000\_al.Turks\_1268.1\_1268.1 (VM).kernel (233664 bytes)  
Device #1: Kernel ./kernels/4098/markov\_le\_plus\_v4.Turks\_1268.1\_1268.1 (VM).kernel not found in cache! Building may take a while...  
Device #1: Kernel ./kernels/4098/markov\_le\_plus\_v4.Turks\_1268.1\_1268.1 (VM).kernel (322108 bytes)  
Device #1: Kernel ./kernels/4098/bzero.Turks\_1268.1\_1268.1 (VM).kernel (33864 bytes)  
Device #2: Kernel ./kernels/4098/m0000\_al.Cypress\_1268.1\_1268.1 (VM).kernel not found in cache! Building may take a while...  
Device #2: Kernel ./kernels/4098/m0000\_al.Cypress\_1268.1\_1268.1 (VM).kernel (231632 bytes)  
Device #2: Kernel ./kernels/4098/markov\_le\_plus\_v4.Cypress\_1268.1\_1268.1 (VM).kernel not found in cache! Building may take a while...

Provided by given: [https://www.youtube.com/watch?v=Tj-U5hQSy\\_E](https://www.youtube.com/watch?v=Tj-U5hQSy_E)

**HashCat GUI v0.31 Tutorial and Overview**

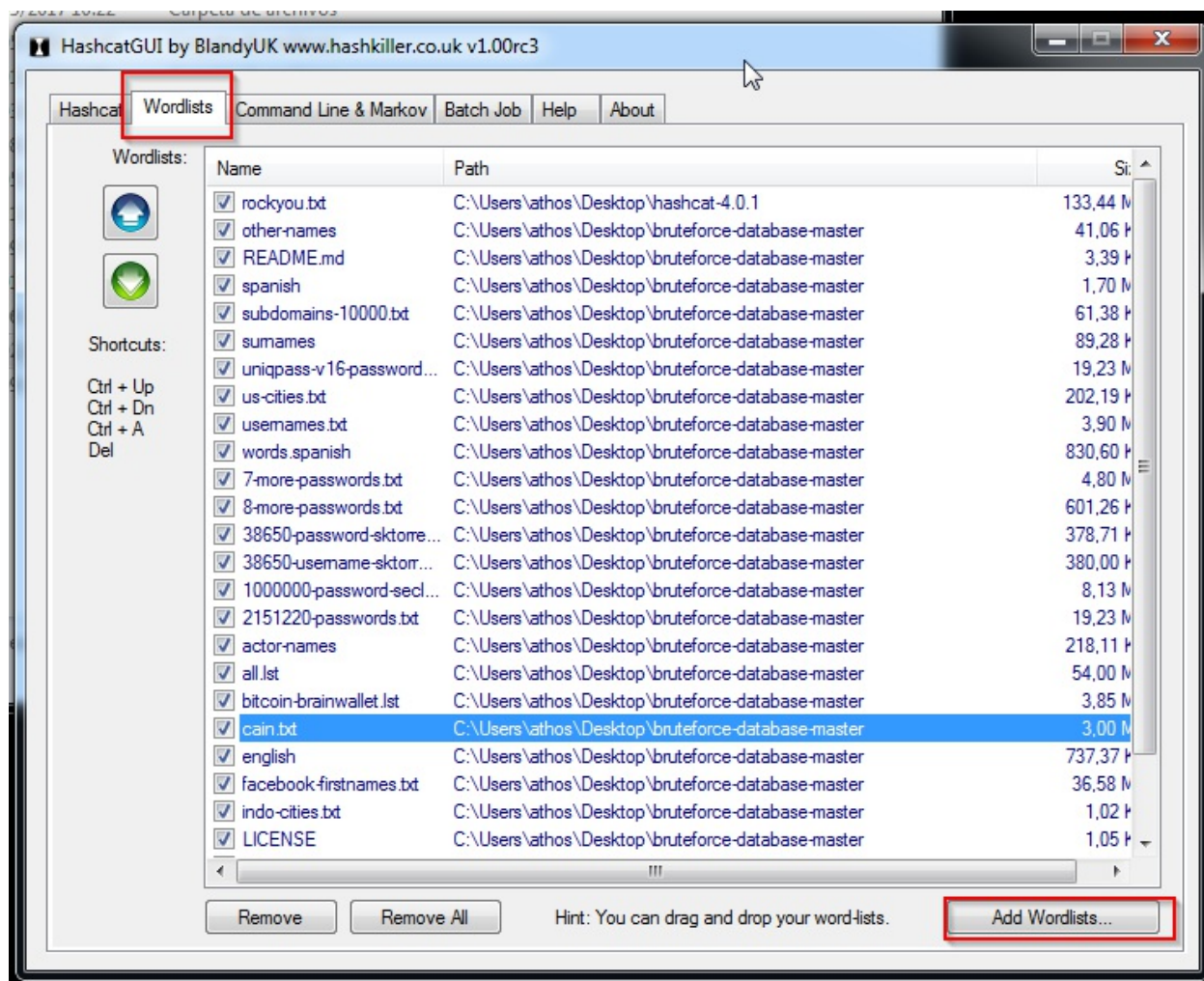
Hashcat GUI v0.31 Tutorial and Overview

Abriremos hashcat-gui y en la parte inferior, seleccionaremos la ruta al hashcat



Despues, para hacer un ataque por diccionario, seleccionaremos la pestaña wordlists y los añadiremos





Despues de esto, volveremos a la pestaña hashcat, seleccionaremos el archivo donde estan los hashes (archivo shadow de linux) y lo añadiremos a input, seleccionaremos el archivo destino y lo añadiremos a output y pincharemos en l'm a hashkiller

```
C:\Windows\System32\cmd.exe - hashcat64.exe -a 0 --session=2017-12-03 -m 1800 -w 3 --force --status --status-timer=60 --potfile-disable -p : --gpu-temp-retain=95 --gpu-t...

* Device #1: Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz, skipped.
OpenCL Platform #2: NVIDIA Corporation
* Device #2: GeForce GTX 960, 512/2048 MB allocatable, 8MCU

Hashes: 2 digests; 2 unique digests, 2 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Uses-64-Bit

Password length minimum: 0
Password length maximum: 256

ATTENTION! Pure (unoptimized) OpenCL kernels selected.
This enables cracking passwords and salts > length 32 but for the price of drastical reduced performance.
If you want to switch to optimized OpenCL kernels, append -O to your commandline.

Watchdog: Temperature abort trigger set to 100c
Watchdog: Temperature retain trigger set to 95c

Dictionary cache hit:
* Filename.: C:\Users\athos\Desktop\hashcat-4.0.1\rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace...: 14344385

[status] [pause] [resume] [h]ypass [c]heckpoint [q]uit =>

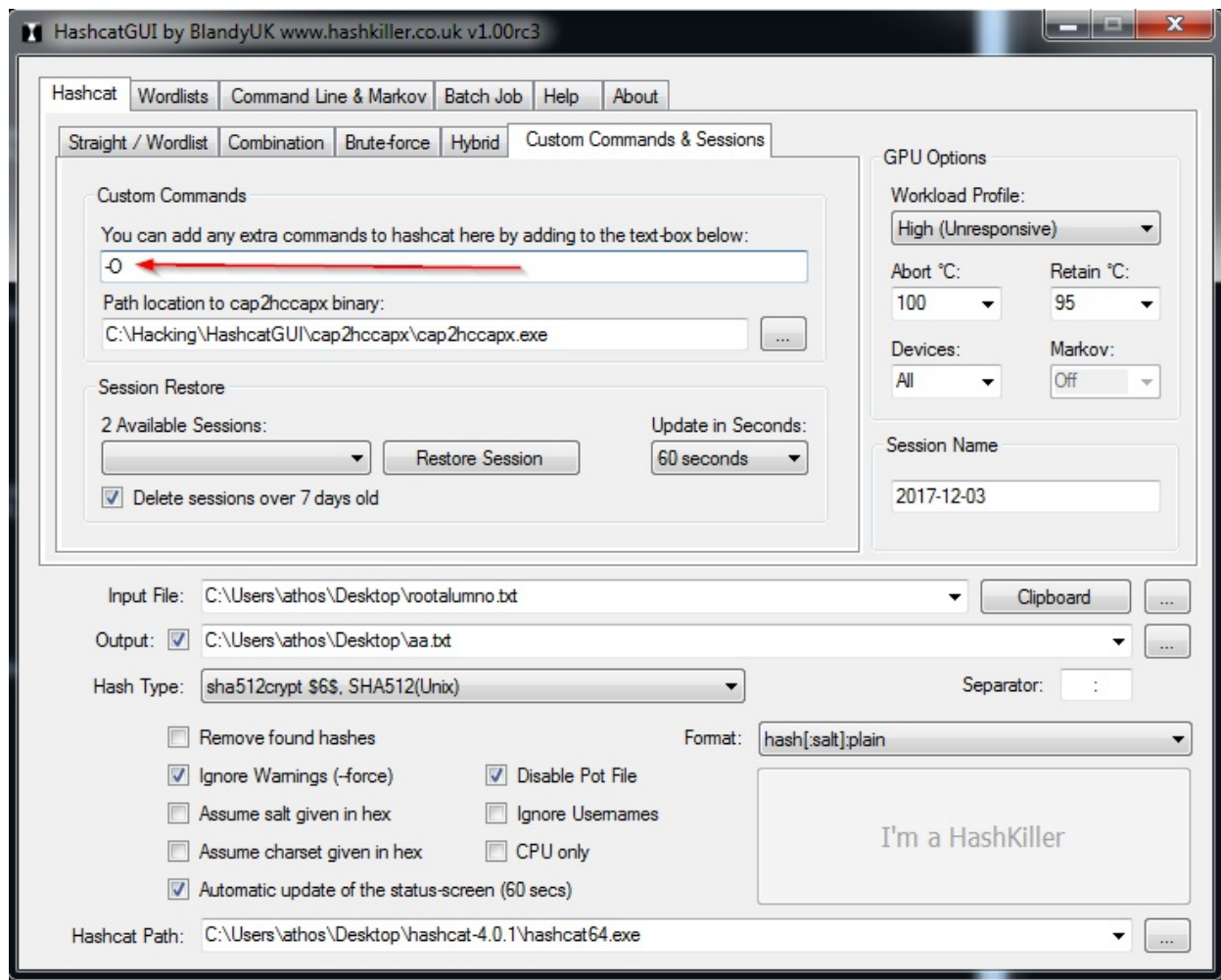
Session.....: 2017-12-03
Status.....: Running
Hash.Type.....: sha512crypt $6$, SHA512 (Unix)
Hash.Target....: C:\Users\athos\Desktop\rootalumno.txt
Time.Started...: Sun Dec 03 14:09:08 2017 (4 secs)
Time.Estimated..: Sun Dec 03 16:17:02 2017 (2 hours, 7 mins)
Guess.Base.....: File (C:\Users\athos\Desktop\hashcat-4.0.1\rockyou.txt)
Guess.Queue....: 1/22 (3.20%)
Speed.Dev.#2....: 3740 H/s (76.34ms)
Recovered.....: 0/2 (0.00%) Digests, 0/2 (0.00%) Salts
Progress.....: 0/28688770 (0.00%)
Rejected.....: 0/0 (0.00%)
Restore.Point...: 0/14344385 (0.00%)
Candidates.#2...: 123456 -> redlips
HWMon.Dev.#2....: Temp: 46c Fan: 33% Util:100% Core:1126MHz Mem:3505MHz Bus:16

[status] [pause] [resume] [h]ypass [c]heckpoint [q]uit => _
```

Como se puede ver en la imagen, nos da una alerta de que el sistema no esta funcionando del modo optimo. Más abajo, podemos comprobar que hashcat esta testeando 3740 hashes por segundo, algo que puede parecer mucho pero que no lo es.

Para solucionar esto, seguimos la recomendacion del mensaje de alerta, y vamos a añadir la opcion -O a hashcat.

Iremos a la pestaña Custom Commands & sessions y lo introduciremos ahi.



Volveremos a la pestaña `straight/wordlist` y volveremos a iniciar el ataque

Como podemos ver en la siguiente imagen, ya no nos aparece el mensaje de alerta, y vemos que la velocidad de calculo de hashes se ha multiplicado por 10

```

C:\Windows\System32\cmd.exe - hashcat64.exe -a 0 --session=2017-12-03 -m 1800 -w 3 --force --status --status-timer=60 --potfile-disable -p : -O --gpu-tem
hashcat (v4.0.1) starting...

OpenCL Platform #1: Intel(R) Corporation
=====
* Device #1: Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz, skipped.

OpenCL Platform #2: NVIDIA Corporation
=====
* Device #2: GeForce GTX 960, 512/2048 MB allocatable, 8MCU

Hashes: 2 digests; 2 unique digests, 2 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Optimized-Kernel
* Zero-Byte
* Uses-64-Bit

Password length minimum: 0
Password length maximum: 16

Watchdog: Temperature abort trigger set to 100c
Watchdog: Temperature retain trigger set to 95c

Dictionary cache hit:
* Filename..: C:\Users\athos\Desktop\hashcat-4.0.1\rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

[sl]tatus [pl]ause [rl]esume [bl]ypass [cl]heckpoint [ql]uit =>

Session.....: 2017-12-03
Status.....: Running
Hash.Type.....: sha512crypt $6$, SHA512 (Unix)
Hash.Target....: C:\Users\athos\Desktop\rootalumno.txt
Time.Started...: Sun Dec 03 14:15:36 2017 (<? secs)
Time.Estimated..: Sun Dec 03 14:29:29 2017 (<13 mins, 46 secs)
Guess.Base.....: File (C:\Users\athos\Desktop\hashcat-4.0.1\rockyou.txt)
Guess.Queue....: 1/207 (<3.73%)
Speed.Dev.#2....: 34430 H/s (<45.39ms)
Recovered.....: 0/2 (<0.00%) Digests: 0/2 (<0.00%) Salts
Progress.....: 245858/28688770 (<0.86%)
Rejected.....: 98/245858 (<0.04%)
Restore.Point...: 122911/14344385 (<0.86%)
Candidates.#2...: misty69 -> 231276
HWMon.Dev.#2....: Temp: 53c Fan: 33% Util: 90% Core:1126MHz Mem:3505MHz Bus:16

Cracking performance lower than expected?

* Update your OpenCL runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

[sl]tatus [pl]ause [rl]esume [bl]ypass [cl]heckpoint [ql]uit =>

```

En la imagen podemos apreciar, como hashcat va recorriendo los diccionarios que le hemos puesto, y probando todas las combinaciones.



```
C:\Windows\System32\cmd.exe
* Filename...: C:\Users\athos\Desktop\bruteforce-database-master\other-names
* Passwords...: 5564
* Bytes.....: 42048
* Keyspace...: 5564

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

Session.....: 2017-12-03
Status.....: Exhausted
Hash.Type.....: sha512crypt $6$, SHA512 (Unix)
Hash.Target....: C:\Users\athos\Desktop\rootalumno.txt
Time.Started...: Sun Dec 03 14:17:27 2017 (0 secs)
Time.Estimated...: Sun Dec 03 14:17:27 2017 (0 secs)
Guess.Base.....: File (C:\Users\athos\Desktop\bruteforce-database-master\other-names)
Guess.Queue.....: 1/26 (3.85%)
Speed.Dev.#2.....: 27709 H/s (7.64ms)
Recovered.....: 0/2 (0.00%) Digests, 0/2 (0.00%) Salts
Progress.....: 11128/11128 (100.00%)
Rejected.....: 8/11128 (0.07%)
Restore.Point...: 5564/5564 (100.00%)
Candidates.#2....: # -> zwi
HWMon.Dev.#2.....: Temp: 49c Fan: 33% Util: 98% Core:1126MHz Mem:3505MHz Bus:16

Dictionary cache hit:
* Filename...: C:\Users\athos\Desktop\bruteforce-database-master\README.md
* Passwords...: 53
* Bytes.....: 3476
* Keyspace...: 53

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

Session.....: 2017-12-03
Status.....: Exhausted
Hash.Type.....: sha512crypt $6$, SHA512 (Unix)
Hash.Target....: C:\Users\athos\Desktop\rootalumno.txt
Time.Started...: Sun Dec 03 14:17:29 2017 (0 secs)
Time.Estimated...: Sun Dec 03 14:17:29 2017 (0 secs)
Guess.Base.....: File (C:\Users\athos\Desktop\bruteforce-database-master\README.md)
Guess.Queue.....: 2/26 (7.69%)
Speed.Dev.#2.....: 0 H/s (2.70ms)
Recovered.....: 0/2 (0.00%) Digests, 0/2 (0.00%) Salts
Progress.....: 106/106 (100.00%)
Rejected.....: 60/106 (56.60%)
Restore.Point...: 53/53 (100.00%)
Candidates.#2....: ->
HWMon.Dev.#2.....: Temp: 46c Fan: 33% Util: 2% Core:1126MHz Mem:3505MHz Bus:16

Dictionary cache hit:
* Filename...: C:\Users\athos\Desktop\bruteforce-database-master\spanish
* Passwords...: 172122
* Bytes.....: 1787125
* Keyspace...: 172122

Session.....: 2017-12-03
Status.....: Cracked
Hash.Type.....: sha512crypt $6$, SHA512 (Unix)
Hash.Target....: C:\Users\athos\Desktop\rootalumno.txt
Time.Started...: Sun Dec 03 14:17:31 2017 (5 secs)
Time.Estimated...: Sun Dec 03 14:17:36 2017 (0 secs)
Guess.Base.....: File (C:\Users\athos\Desktop\bruteforce-database-master\spanish)
Guess.Queue.....: 3/26 (11.54%)
Speed.Dev.#2.....: 30010 H/s (86.30ms)
Recovered.....: 2/2 (100.00%) Digests, 2/2 (100.00%) Salts
Progress.....: 164836/344244 (47.83%)
Rejected.....: 796/164836 (0.48%)
Restore.Point...: 0/172122 (0.00%)
Candidates.#2....: \a -> fatalmente
HWMon.Dev.#2.....: Temp: 54c Fan: 33% Util:100% Core:1126MHz Mem:3505MHz Bus:16

Started: Sun Dec 03 14:17:23 2017
Stopped: Sun Dec 03 14:17:38 2017

C:\Users\athos\Desktop\hashcat-4.0.1_
```

En los dos primeros diccionarios no ha conseguido recuperar las contraseñas, pero en el tercero si. Lo podemos ver en la linea Recovered



























El nombre del diccionario se llama spanish, vamos a ver lo que contiene.

```
16408 aprieto
16409 apriimar
16410 apriimar
16411 apriori'stica
16412 apriori'stica
16413 apriori'stico
16414 apriori'stico
16415 apriorismo
16416 apriorismo
16417 aprisa
16418 aprisa
16419 aprisoadero
16420 aprisoadero
16421 aprisoar
16422 aprisoar
16423 aprisoar
16424 aprisoar
16425 aprisionadamente
16426 aprisionadamente
16427 aprisionar
16428 aprisionar
16429 aprisoquero
16430 aprisoquero
16431 aproar
16432 aproar
16433 aprobacio'n
16434 aprobacio'n
16435 aprobada
16436 aprobada
16437 aprobado
16438 aprobado
16439 aprobador
16440 aprobador
16441 aprobadora
16442 aprobadora
16443 aprobante
16444 aprobante
16445 aprobanza
16446 aprobanza
16447 aprobar
16448 aprobar
16449 aprobativa
16450 aprobativa
16451 aprobativo
16452 aprobativo
16453 aprobatoria
16454 aprobatoria
16455 aprobatoriamente
16456 aprobatoriamente
16457 aprobatorio
16458 aprobatorio
16459 aproches
16460 aproches
16461 aprodar
16462 aprodar
16463 aprometer
16464 aprometer
16465 aprontamiento
```

La contraseña estaba ahí.

Nota: En un principio intente hacer el ataque por fuerza bruta, pero no era viable ya que se iba a demorar mucho mas del tiempo que tenia. Estuve como 15 minutos descargando diferentes diccionarios, de grupos de rock, palabras en español, nombres, apellidos etc etc. De esta manera, la contraseña la saque en menos de 2 minutos. Hay que considerar que era una contraseña extremadamente debil, y que uno de los primeros diccionarios tenia la clave.

Estos son los diccionarios que me descargue.

Nombre	Fecha de modifica...	lipo	lamaño
 7-more-passwords	12/11/2017 18:41	Documento de tex...	4.920 KB
 8-more-passwords	12/11/2017 18:41	Documento de tex...	602 KB
 38650-password-sktorrent	12/11/2017 18:41	Documento de tex...	379 KB
 38650-username-sktorrent	12/11/2017 18:41	Documento de tex...	381 KB
 1000000-password-seclists	12/11/2017 18:41	Documento de tex...	8.330 KB
 2151220-passwords	12/11/2017 18:41	Documento de tex...	19.691 KB
 actor-names	22/10/2003 8:07	Archivo	219 KB
 all.lst	24/02/2015 17:19	Archivo LST	55.298 KB
 bitcoin-brainwallet.lst	12/11/2017 18:41	Archivo LST	3.938 KB
 cain	12/11/2017 18:41	Documento de tex...	3.076 KB
 english	21/10/2003 9:35	Archivo	738 KB
 facebook-firstnames	12/11/2017 18:41	Documento de tex...	37.454 KB
 indo-cities	12/11/2017 18:41	Documento de tex...	2 KB
 LICENSE	12/11/2017 18:41	Archivo	2 KB
 lower.lst	08/10/2003 1:58	Archivo LST	831 KB
 movie-characters	22/10/2003 8:07	Archivo	195 KB
 names.hp	22/10/2003 8:07	Archivo HP	884 KB
 other-names	22/10/2003 8:07	Archivo	42 KB
 README.md	12/11/2017 18:41	Archivo MD	4 KB
 spanish	21/10/2003 9:43	Archivo	1.746 KB
 subdomains-10000	12/11/2017 18:41	Documento de tex...	62 KB
 surnames	22/10/2003 8:07	Archivo	90 KB
 uniqpass-v16-passwords	12/11/2017 18:41	Documento de tex...	19.691 KB
 us-cities	12/11/2017 18:41	Documento de tex...	203 KB
 usernames	12/11/2017 18:41	Documento de tex...	3.997 KB
 words.spanish	10/11/1993 7:55	Archivo SPANISH	831 KB

El resultado en el archivo de salida es el siguiente

```
$6$uJu2EH0C$QGzu0z0gLIUr6t7TbyrM6Co1SFysCZexLZDZ.J10UyACN455zSVnn0BIQm44M8LjggYDUYsl
$6$TGpVknGA$Mz.W.BPS0Jv1thFsS/vnYMi4pvNh.ePL0pH8EZ/JPmsNrKcejrap8dieTJ24HzW0h0EOIWI
```