



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος **Ασφάλεια Πληροφοριακών Συστημάτων**

Άσκηση <<αριθμός άσκησης>>	4η Άσκηση - Ασφάλεια συνθηματικών
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Κουσουννής Κωνσταντίνος p14086
Ημερομηνία παράδοσης	03-12-2018



Χρησιμοποιώντας όποια εργαλεία ελέγχου συνθηματικών θέλετε (π.χ. Cain, John the Ripper, OphCrack κτλ) να αποκρυπτογραφήσετε τα συνθηματικά, με την καταλληλότερη μέθοδο, με βάση τα δεδομένα που δίνονται σε κάθε περίπτωση. Να αναφέρεται σε κάθε περίπτωση, το συνθηματικό, ποια μέθοδο χρησιμοποιήσατε, γιατί την επιλέξατε και πόσος χρόνος απαιτήθηκε.

1)9FC63641386C279AB1A62C51F0C8268AF67309218121C804905E3F3BD7D59E51 : Το συνθηματικό μπορεί να περιλαμβάνει οποιοδήποτε χαρακτήρα αλλά έχει μικρό μήκος (<6). Με βάση τα χαρακτηριστικά του hash value να βρείτε ποια συνάρτηση έχει χρησιμοποιηθεί.

2)95DCC3CEAC9865A72B66C84546A3549C : Είναι MD5 συνθηματικό. Το συνθηματικό περιλαμβάνει μόνο μικρά γράμματα και είναι μέχρι 7 χαρακτήρες.

3)B54B9E2BE205AB4D239258FEE9735952D19C9762 : Πρόκειται για SHA-1 συνθηματικό, το οποίο είναι πάνω από 10 χαρακτήρες. Είναι passphrase το οποίο περιλαμβάνει μόνο γράμματα, μικρά και κεφαλαία. (Υπόδειξη: διαλέξτε την κατάλληλη μέθοδο ανάκτησης του συνθηματικού και τις κατάλληλες παραμέτρους)

3)836DCD3F1A0E9A14B0D83E64125B32E0D4440F45 : Το συνθηματικό είναι ελληνική λέξη.

4)2D4027D6DF9C0256B8D4474CE88F8C88 : Πρόκειται για το MD5 hash του συνθηματικού. Όμως επειδή γνωρίζουμε ότι το σύστημα-στόχος χρησιμοποιεί μόνο 4-ψήφια PIN, καλείστε να δημιουργήσετε το rainbow table για όλα τα συνθηματικά-PIN 4 ψηφίων για τη συνάρτηση MD5. Περιλάβετε στην απάντηση αυτού του βήματος πόσος χρόνος απαιτήθηκε στο σύστημά σας για να υπολογιστεί το rainbow table το οποίο θα έχει δημιουργηθεί με τέτοιες παραμέτρους ώστε (α) να παρέχει 100% επιτυχία για την εύρεση οποιουδήποτε PIN 4 ψηφίων και (β) να έχει το ελάχιστο δυνατό μέγεθος. Επίσης περιλάβετε screenshot από την δημιουργία του rainbow table.



1) 9FC63641386C279AB1A62C51F0C8268AF67309218121C804905E3F3BD7D59E51 : Το συνθηματικό μπορεί να περιλαμβάνει οποιοδήποτε χαρακτήρα αλλά έχει μικρό μήκος (<6). Με βάση τα χαρακτηριστικά του hash value να βρείτε ποια συνάρτηση έχει χρησιμοποιηθεί.

Ανοίγω στο kali linux το εργαλείο που λέγεται hash-identify
Πληκτρολογώ τον κωδικό στο εργαλείο και μου εμφανίζει τα πιθανά hashes

```
Applications ▾ Places ▾ Terminal ▾

File Edit View Search Terminal Help

#####
9FC63641386C279AB1A62C51F0C8268AF67309218121C804905E3F3BD7D59E51 #
Ana# yzin \ \ \ \ \ 641386C279AB1A62C51 \ \ \ \ \ 268AF67309 \ \ \ \ \ 18121 \ \ \ \ \ 80 \ \ \ \ \ 05E3 \ \ \ \ \ BD7D59E51#
[+]#Snefru-256 #
[+]#SHA-256 #
[+]#RIPEMD-256 #
[+]#Haval-256 #
[+]#GOST R 34.11-94 v1.1 #
[+]#GOST CryptoPro S-Box By Zion3R #
[+]#SHA3-256 www.Blackploit.com #
[+]#Skein-256 Root@Blackploit.com #
[+]#####
^Z
[5]-----
HASH: 9FC63641386C279AB1A62C51F0C8268AF67309218121C804905E3F3BD7D59E51
bash: hashid.py: command not found
Possible Hashs: kt0p# 
[+] SHA-256
[+] Haval-256

Least Possible Hashs:
[+] GOST R 34.11-94
[+] RipeMD-256
[+] SNEFRU-256
[+] SHA-256(HMAC)
[+] Haval-256(HMAC)
[+] RipeMD-256(HMAC)
[+] SNEFRU-256(HMAC)
[+] SHA-256(md5($pass))
[+] SHA-256(sha1($pass))

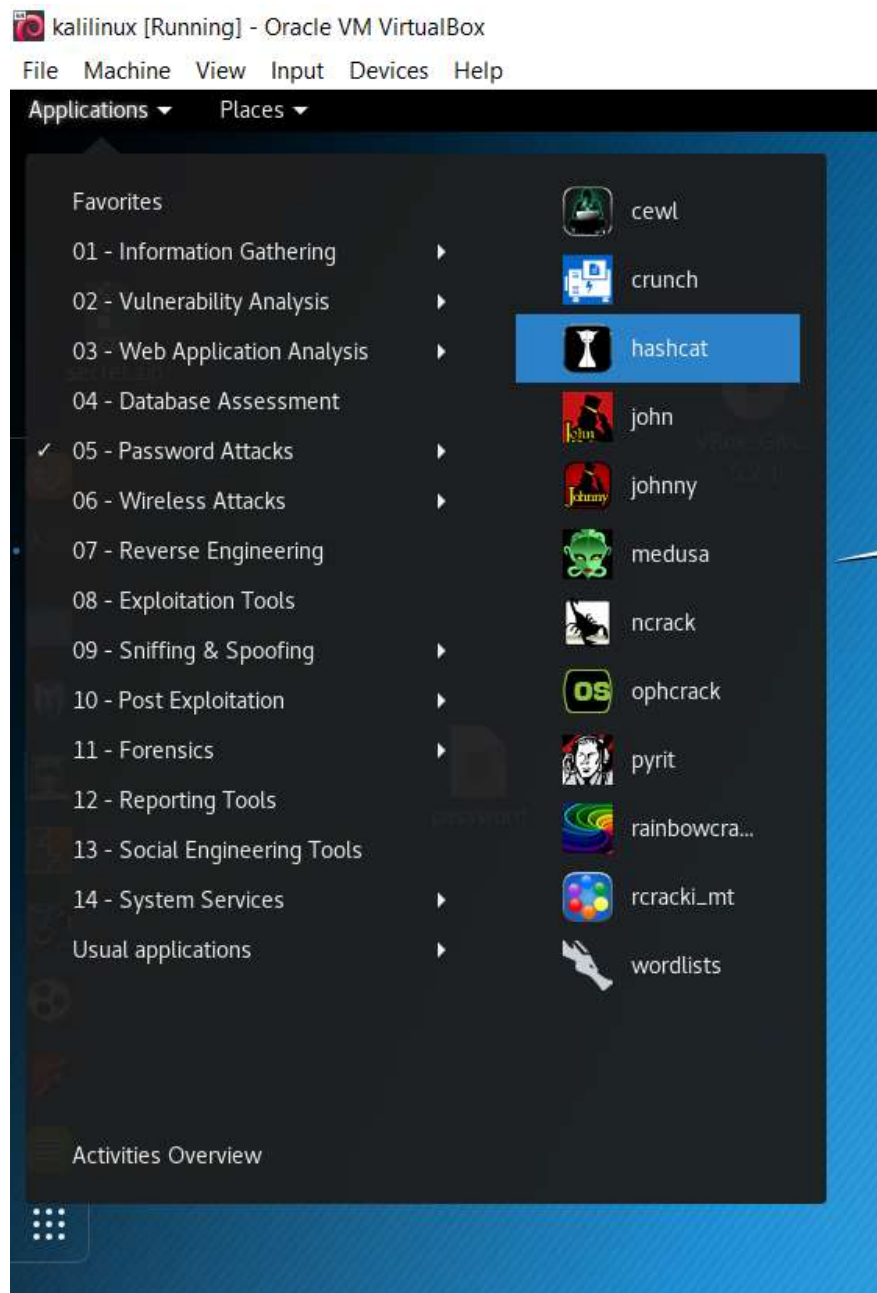
-----
HASH: --help

Not Found.

-----
HASH: █
```



Θα χρησιμοποιήσουμε το εργαλείο hashcat στο Kali Linux.



Μας λέει ότι τα πιο
πιθανά αποτελέσματα
είναι το SHA-256 ή
Haval-256

Παιρνουμε το dictionary που έχει το hashcat

Πληκτρολογούμε την εντολή **locate rockyou.txt** και μας βρίσκει η εντολή το αντίστοιχο path
δηλαδή:



```
root@kali:~# locate rockyou.txt
/root/Desktop/rockyou.txt
/usr/share/wordlists/rockyou.txt.gz
root@kali:~#
```

Βλέπουμε ότι το Path: **/usr/share/wordlist/rockyou.txt.gz**

Στην συνέχεια κανουμε αντιγραφή στην επιφάνει εργασίας με την εντολή
cp /usr/share/wordlist/rockyou.txt.gz root/Desktop/rockyou.txt.gz

Στην συνέχεια εξαγουμε το ζιπαρισμένο αρχείο σε txt αρχείο
gunzip rockyou.txt.gz rockyou.txt

2)95DCC3CEAC9865A72B66C84546A3549C : Είναι MD5 συνθηματικό. Το συνθηματικό περιλαμβάνει μόνο μικρά γράμματα και είναι μέχρι 7 χαρακτήρες.



kalilinux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places Terminal

File Edit View Search Terminal Help

```
6 | Hybrid Wordlist + Mask
7 | Hybrid Mask + Wordlist
```

- [Built-in Charsets] -

? | Charset

====+=====

```
l | abcdefghijklmnopqrstuvwxyz
u | ABCDEFGHIJKLMNOPQRSTUVWXYZ
d | 0123456789
h | 0123456789abcdef
H | 0123456789ABCDEF
s | !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
a | ?l?u?d?s
b | 0x00 - 0xff
```

- [OpenCL Device Types] -

| Device Type

====+=====

```
1 | CPU
2 | GPU
3 | FPGA, DSP, Co-Processor
```

- [Workload Profiles] -

#	Performance	Runtime	Power Consumption	Desktop Impact
1	Low	2 ms	Low	Minimal
2	Default	12 ms	Economic	Noticeable
3	High	96 ms	High	Unresponsive
4	Nightmare	480 ms	Insane	Headless

- [Basic Examples] -

Attack-Mode	Hash-Type	Example command
Wordlist	\$P\$	hashcat -a 0 -m 400 example400.hash example.dict
Wordlist + Rules	MD5	hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
Brute-Force	MD5	hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a
Combinator	MD5	hashcat -a 1 -m 0 example0.hash example.dict example.dict

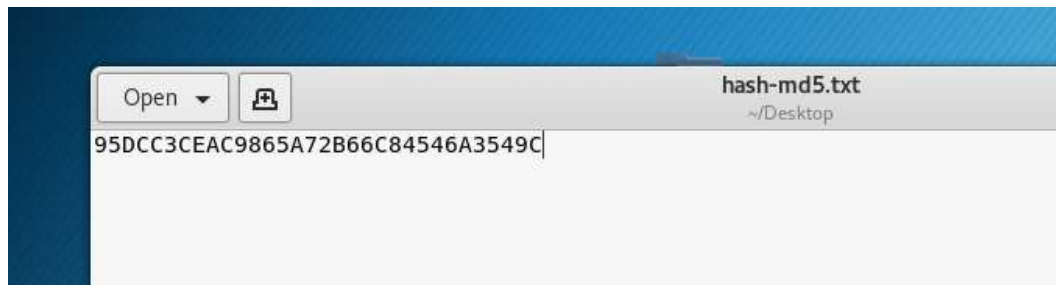
If you still have no idea what just happened, try the following pages:

* https://hashcat.net/wiki/#howtos_videos_papers_articles_etc_in_the_wild

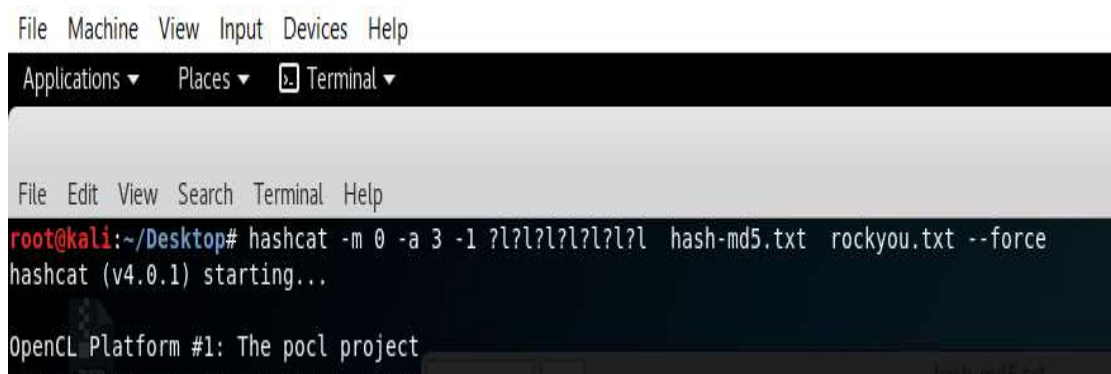
* <https://hashcat.net/faq/>

```
root@kali:~# hashcat -m 0 -a 3 -1 ?l hash-md5.txt rockyou.txt --force
```

Παίρνουμε το password απο την εκφώνηση 95DCC3CEAC9865A72B66C84546A3549C



Αποθηκεύουμε το κωδικό σε ένα αρχείο που ονομάζεται hash-md5.txt



Πληκρολογώ την εντολή :

hashcat -m 0 -a 3 -1 ?l?l?l?l?l?l?l hash-md5.txt rockyou.txt --force

Το **-m** είναι το hashtype που χρησιμοποιείτε.



Options Short / Long	Type	Description	Example
-m, --hash-type 65536 entries	Num	Hash-type, see references below	-m 1000
-a, --attack-mode	Num	Attack-mode, see references below	-a 3
-V, --version		Print version	
-h, --help		Print help	
-E, --quiet		Suppress output	
-N, --hex-charset		Assume charset is given in hex	
-S, --hex-salt		Assume salt is given in hex	
-W, --hex-wordlist		Assume words in wordlist are given in hex	
-f, --force		Ignore warnings	
-u, --status		Enable automatic update of the status screen	
-t, --status-timer	Num	Sets seconds between status screen updates to X	--status-timer=1
-M, --machine-readable		Display the status view in a machine-readable format	

Το **o** είναι ότι θέλουμε να είναι md5

Not Found	#	Name	Category
Single Hash			
900		MD4	Raw Hash
0		MD5	Raw Hash
5100		Half MD5	Raw Hash
100		SHA1	Raw Hash
1300		SHA-224 (minimum: 0)	Raw Hash
1400		SHA-256 (maximum: 256)	Raw Hash
10800		SHA-384	Raw Hash
1700		SHA-512 (unoptimized) OpenCL kernels selected.	Raw Hash
5000		SHA-3 (Keccak) (passwords and salts > length 32 but < 1024 will result in a drastic reduction of performance)	Raw Hash
600		BLAKE2b-512 (to optimized OpenCL kernels, append --blake2b-optimized)	Raw Hash
10100		Single Hash	Raw Hash

Το **-a** είναι το είδος της επίθεσης

Options Short / Long	Type	Description	Example
-m, --hash-type 65536 entries	Num	Hash-type, see references below	-m 1000
-a, --attack-mode	Num	Attack-mode, see references below	-a 3
-V, --version		Print version	
-h, --help		Print help	
-E, --quiet		Suppress output	
-N, --hex-charset		Assume charset is given in hex	
-S, --hex-salt		Assume salt is given in hex	
-W, --hex-wordlist		Assume words in wordlist are given in hex	
-f, --force		Ignore warnings	
-u, --status		Enable automatic update of the status screen	
-t, --status-timer	Num	Sets seconds between status screen updates to X	--status-timer=1
-M, --machine-readable		Display the status view in a machine-readable format	



Το 3 είναι brute-force

```
# Mode length minimum: 0
====+==== length maximum: 256
0 | Straight
1 | Combination (unoptimized) OpenCL kernels se
3 | Brute-force (cracking passwords and salts > les
6 | Hybrid Wordlist + Mask (optimized OpenCL kern
7 | Hybrid Mask + Wordlist
Watchdog: Hardware monitoring interface not fo
[ Built-in Charsets ] abort trigger disabled.
```

-1 ?l?l?l?l?l?l?l είναι για να βάλουμε μικρά γράμματα και να περιέχει 7 χαρακτήρες

```
Passwd-generate-rules-seed Num Force RNG seed set to X
-1, --custom-charset1 CS User-defined charset ?1 -1 ?l?d?u
-2, --custom-charset2 CS User-defined charset ?2 -2 ?l?d?s
-3, --custom-charset3 CS User-defined charset ?3 price of practical reduced p
-4, --custom-charset4 CS User-defined charset ?4 per commandline
-i, --increment Enable mask increment mode
watchdog-increment-min Num Start mask incrementing at X --increment-min=4
watchdog-increment-max Num Stop mask incrementing at X --increment-max=8
watchdog-temperature-retain-trigger disabled.
- [ Hash modes ] -
```

```
- [ Built-in Charsets ] -
This enables cracking passwords and salts
?| Charset switch to optimized OpenCL
====+====
walc| abcdefghijklmnopqrstuvwxyzterface no
wadc| ABCDEFGHIJKLMNOPQRSTUVWXYZger disabl
wadc| 0123456789ature retain trigger disab
b | 0123456789abcdef
```

Το l είναι για τα μικρά γράμματα. Και οτι επαναλαμβάνω το ?l 7 φορές τότε το hashcat περιέχει 7 γράμματα μικρά ακριβώς.

hash-md5.txt Είναι το αρχείο που αποθηκεύσαμε τον κωδικό
rockyou.txt Εχουμε το λεξικό που χρησιμοποιεί ο hashcat



```
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.....: hashcat
Status.....: Exhausted
Hash.Type.....: MD5
Hash.Target.....: 95dcc3ceac9865a72b66c84546a3549c
Time.Started.....: Thu Nov 29 17:29:11 2018 (0 secs)
Time.Estimated...: Thu Nov 29 17:29:11 2018 (0 secs)
Guess.Mask.....: manuel [6]
Guess.Queue.....: 181/14336793 (0.00%)
Speed.Dev.#1.....: 0 H/s (0.08ms)
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point....: 1/1 (100.00%)
Candidates.#1....: manuel -> manuel
HWMon.Dev.#1.....: N/A

- Device #1: autotuned kernel-accel to 1024
- Device #1: autotuned kernel-loops to 1
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => 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.....: hashcat
Status.....: Exhausted
Hash.Type.....: MD5
Hash.Target.....: 95dcc3ceac9865a72b66c84546a3549c
Time.Started.....: Thu Nov 29 17:29:11 2018 (0 secs)
Time.Estimated...: Thu Nov 29 17:29:11 2018 (0 secs)
Guess.Mask.....: myspace [7]
Guess.Queue.....: 182/14336793 (0.00%)
Speed.Dev.#1.....: 0 H/s (0.01ms)
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point....: 1/1 (100.00%)
Candidates.#1....: myspace -> myspace
HWMon.Dev.#1.....: N/A

- Device #1: autotuned kernel-accel to 1024
```

Το Hashcat έχει ξεκινήσει .



```
Approaching final keypace - workload adjusted.

Session.....: hashcat
Status.....: Exhausted
Hash.Type.....: MD5
Hash.Target.....: 95dcc3ceac9865a72b66c84546a3549c
Time.Started.....: Thu Nov 29 17:55:30 2018 (0 secs)
Time.Estimated...: Thu Nov 29 17:55:30 2018 (0 secs)
Guess.Mask.....: pink00 [6]
Guess.Queue.....: 20713/14336793 (0.14%)
Speed.Dev.#1.....: 0 H/s (1.29ms)
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point....: 1/1 (100.00%)
Candidates.#1....: pink00 -> pink00
HWMon.Dev.#1.....: N/A

- Device #1: autotuned kernel-accel to 1
- Device #1: autotuned kernel-loops to 1
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => 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 keypace - workload adjusted.

Session.....: hashcat
Status.....: Exhausted
Hash.Type.....: MD5
Hash.Target.....: 95dcc3ceac9865a72b66c84546a3549c
Time.Started.....: Thu Nov 29 17:55:35 2018 (1 sec)
Time.Estimated...: Thu Nov 29 17:55:35 2018 (0 secs)
Guess.Mask.....: pianos [6]
Guess.Queue.....: 20714/14336793 (0.14%)
Speed.Dev.#1.....: 0 H/s (2.24ms)
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point....: 1/1 (100.00%)
Candidates.#1....: pianos -> pianos
HWMon.Dev.#1.....: N/A
```

Όπως βλέπουμε είναι στο 0.14% είναι θα πρέπει να το αφήσουμε να συνεχίσει μέχρι να βρει τον κωδικό.

3)B54B9E2BE205AB4D239258FEE9735952D19C9762 : Πρόκειται για SHA-1 συνθηματικό, το οποίο είναι πάνω από 10 χαρακτήρες. Είναι passphrase το οποίο περιλαμβάνει μόνο γράμματα, μικρά και κεφαλαία. (Υπόδειξη: διαλέξτε την κατάλληλη μέθοδο ανάκτησης του συνθηματικού και τις κατάλληλες παραμέτρους)

```
root@kali:~/Desktop# hashcat -m 100 -a 3 -1 ?ul -i --increment-min 10 hash-sha1.txt rockyou.txt --force
hashcat (v4.0.1) starting...

OpenCL Platform #1: The pocl project
```

Με αντίστοιχο τρόπο έχουμε την εντολή:

hashcat -m 100 -a 3 -1 ?ul -i --increment-min 10 hash-md5.txt rockyou.txt --force



-m 100 είναι sha1

5100	Half MD5	Raw Hash
100	SHA1	Raw Hash
1300	SHA-224	Raw Hash
1400	SHA-256	Raw Hash

Το **-a** είναι το είδος της επίθεσης

Options Short / Long	Type	Description	Example
-m --hash-type	Num	Hash-type, see references below	-m 1000
-a --attack-mode	Num	Attack-mode, see references below	-a 3
-V --version		Print version	
-h --help		Print help	
-q --quiet		Suppress output	
-C --hex-charset		Assume charset is given in hex	
-S --hex-salt		Assume salt is given in hex	
-W --hex-wordlist		Assume words in wordlist are given in hex	
-i --ignore-warnings		Ignore warnings	
-u --update-status		Enable automatic update of the status screen	

Το **3** είναι brute-force

#	Mode	length minimum: 0
0	Straight	length maximum: 256
1	Combination	(unoptimized) OpenCL kernels
3	Brute-force	Brute-force (unoptimized) OpenCL kernels
6	Hybrid Wordlist + Mask	Hybrid Wordlist + Mask (unoptimized) OpenCL kernels
7	Hybrid Mask + Wordlist	Hybrid Mask + Wordlist (unoptimized) OpenCL kernels

-i ?ul συμβολίζει ότι μπορεί να είναι μικρά ή κεφαλαία

u	ABCDEFGHIJKLMNOPQRSTUVWXYZ
l	abcdefghijklmnopqrstuvwxyz
d	0123456789
h	0123456789abcdef

-i --increment-min 10

-3 --custom-charsets3	CS	User-defined charset 73	
-4 --custom-charsets4	CS	User-defined charset 74	
-i --increment		Enable mask increment mode	
--increment-min	Num	Start mask incrementing at X	--increment-min=4
--increment-max	Num	Stop mask incrementing at X	--increment-max=8



Εδώ θέλουμε να είναι ο κωδικός μεγαλύτερος απο 10 χαρακτήρες.

```
crackmapexec-wf -u root -H root --hashes hash-sha1.txt --wordlist rockyou.txt --force
root@kali:~/Desktop# hashcat -m 100 -a 3 -I ?ul -i --increment-min 10 hash-sha1.txt rockyou.txt --force
hashcat (v4.0.1) starting...
OpenCL Platform #1: The pocl project
=====
* Device #1: pthread-Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 1024/2961 MB allocatable, 4MCU
Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Applicable optimizers:
* Zero-Byte      Num 0
* Early-Skip     Num 0
* Not-Salted     Num 0
* Not-Iterated   Num 0
* Single-Hash    Num 0
* Single-Salt    Num 0
* Brute-Force    Num 0
* Raw-Hash       Rule 0
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: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.
Watchdog: Temperature retain trigger disabled.
* Device #1: build_opts '-I /usr/share/hashcat/OpenCL -D VENDOR_ID=64 -D CUDA_ARCH=0 -D AMD_ROCM=0 -D VECT_SIZE=8 -D DEVICE_TYPE=2 -D
100 -D unroll'
- Device #1: autotuned kernel-accel to 1024
- Device #1: autotuned kernel-loops to 1
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.....: hashcat
Status.....: Exhausted
Hash.Type.....: SHA1
```



Έχει ξεκινήσει η αποκρυπτογράφηση του κωδικού.

```
File Edit View Search Terminal Help
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
1, --opencl-devices=0, --opencl-platforms=0, --opencl-vector=4
Approaching final key space - workload adjusted.
Session..... hashcat
Status..... Exhausted
Hash.Type..... SHA1
Hash.Target..... b54b9e2be205ab4d239258fee9735952d19c9762
Time.Started..... Fri Nov 30 06:35:04 2018 (0 secs)
Time.Estimated..... Fri Nov 30 06:35:04 2018 (0 secs)
Guess.Mask..... siempretuya [11]
Guess.Queue..... 8106/11380621 (0.07%)
Speed.Dev.#1..... 0 H/s (0.01ms)
Recovered..... 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress..... 1/1 (100.00%)
Rejected..... 0/1 (0.00%)
Restore.Point..... 1/1 (100.00%)
Candidates.#1..... siempretuya -> siempretuya
HWMon.Dev.#1..... N/A
- Device #1: autotuned kernel-accel to 1024
- Device #1: autotuned kernel-loops to 1
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => 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
1, --custom-charset1=1, --custom-charset2=2, --custom-charset3=3, --custom-charset4=4
Approaching final key space - workload adjusted.
Session..... hashcat
Status..... Exhausted
Hash.Type..... SHA1
Hash.Target..... b54b9e2be205ab4d239258fee9735952d19c9762
Time.Started..... Fri Nov 30 06:35:04 2018 (0 secs)
Time.Estimated..... Fri Nov 30 06:35:04 2018 (0 secs)
Guess.Mask..... shutthefuc [10]
Guess.Queue..... 8107/11380621 (0.07%)
Speed.Dev.#1..... 0 H/s (0.01ms)
Recovered..... 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress..... 1/1 (100.00%)
Rejected..... 0/1 (0.00%)
Restore.Point..... 1/1 (100.00%)
Candidates.#1..... shutthefuc -> shutthefuc
HWMon.Dev.#1..... N/A
=====
Category
Raw Hash
Raw Hash
Raw Hash
Raw Hash
Raw Hash
Raw Hash
Raw Hash
```

Οσο τρέχει το πρόγραμμα βλέπουμε οτι οι χαρακτήρες που χρησιμοποιεί είναι 10 και πάνω.
Το πρόγραμμα είναι στο 0.7%

4)2D4027D6DF9C0256B8D4474CE88F8C88 : Πρόκειται για το MD5 hash του συνθηματικού. Όμως επειδή γνωρίζουμε ότι το σύστημα-στόχος χρησιμοποιεί μόνο 4-ψήφια PIN, καλείστε να δημιουργήσετε το rainbow table για όλα τα συνθηματικά-PIN 4 ψηφίων για τη συνάρτηση MD5. Περιλάβετε στην απάντηση αυτού του βήματος πόσος χρόνος απαιτήθηκε στο σύστημά σας για να υπολογιστεί το rainbow table το οποίο θα έχει δημιουργηθεί με τέτοιες παραμέτρους ώστε (α) να παρέχει 100% επιτυχία για την εύρεση οποιουδήποτε PIN 4 ψηφίων και (β) να έχει το ελάχιστο δυνατό μέγεθος. Επίσης περιλάβετε screenshot από την δημιουργία του rainbow table.



```
rtgen md5 toweralpha 1 7 0 -bench
root@kali:~/Desktop# rtgen md5 numeric 1 4 0 3800 33554432 0
rainbow table md5_numeric#1-4_0_3800x33554432_0.rt parameters
hash algorithm:      md5
hash length:         16
charset name:        numeric
charset data:        0123456789
charset data in hex: 30 31 32 33 34 35 36 37 38 39
charset length:      10
plaintext length range: 1 - 4
reduce offset:       0x00000000
plaintext total:     11110
bash: hashid.py: command not found
sequential starting point begin from 0 (0x0000000000000000)
generating...
131072 of 33554432 rainbow chains generated (0 m 28.6 s)
262144 of 33554432 rainbow chains generated (0 m 30.1 s)
393216 of 33554432 rainbow chains generated (0 m 28.7 s)
524288 of 33554432 rainbow chains generated (0 m 28.5 s)
655360 of 33554432 rainbow chains generated (0 m 29.8 s)
786432 of 33554432 rainbow chains generated (0 m 29.2 s)
917504 of 33554432 rainbow chains generated (0 m 27.2 s)
1048576 of 33554432 rainbow chains generated (0 m 31.5 s)
1179648 of 33554432 rainbow chains generated (0 m 32.2 s)
1310720 of 33554432 rainbow chains generated (0 m 33.2 s)
1441792 of 33554432 rainbow chains generated (0 m 29.2 s)
1572864 of 33554432 rainbow chains generated (0 m 27.5 s)
1703936 of 33554432 rainbow chains generated (0 m 30.7 s)
1835008 of 33554432 rainbow chains generated (0 m 29.9 s)
1966080 of 33554432 rainbow chains generated (0 m 33.2 s)
2097152 of 33554432 rainbow chains generated (0 m 27.9 s)
2228224 of 33554432 rainbow chains generated (0 m 28.3 s)
2359296 of 33554432 rainbow chains generated (0 m 30.7 s)
2490368 of 33554432 rainbow chains generated (0 m 30.7 s)
2621440 of 33554432 rainbow chains generated (0 m 32.4 s)
2752512 of 33554432 rainbow chains generated (0 m 32.7 s)
2883584 of 33554432 rainbow chains generated (0 m 29.9 s)
3014656 of 33554432 rainbow chains generated (0 m 27.6 s)
3145728 of 33554432 rainbow chains generated (0 m 29.0 s)
3276800 of 33554432 rainbow chains generated (0 m 32.4 s)
3407872 of 33554432 rainbow chains generated (0 m 33.7 s)
3538944 of 33554432 rainbow chains generated (0 m 32.2 s)
3670016 of 33554432 rainbow chains generated (0 m 35.7 s)
3801088 of 33554432 rainbow chains generated (0 m 33.9 s)
3932160 of 33554432 rainbow chains generated (0 m 31.4 s)
```

Δημιουργώ τους rainbowtables με την εξής εντολή:
rtgen md5 numeric 1 4 0 3800 33554432 0

Θέλω να περιέχει 4 ψηφία που να είναι αριθμοί.



```
File Edit View Search Terminal Help
root@kali:~# rcrack numeric.rt -h 2D4027D6DF9C0256B8D4474CE88F8C88
```

Μολις φτιάξουμε τον rainbow table χρησιμοποιουμε την εντολή rcrack το όνομα του rainbow table που μόλις φτιαξαμε και το τέλος τον κωδικο που θέλουμε να αποκρυπτογραφήσουμε.