



8 ΔΕΚΕΜΒΡΙΟΥ 2022

ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΕΡΓΑΣΙΑ 4

Συντελεστές εργασίας

Χριστοφορίδης Χαράλαμπος – Π19188

Γεωργιάδης Νικόλαος – Π19032

Καρκάνης Ευστράτιος – Π19064



Περιεχόμενα

| | |
|------------------------------|----|
| 1. Πρώτο συνθηματικό | 2 |
| 2. Δεύτερο συνθηματικό | 3 |
| 3. Τρίτο συνθηματικό | 5 |
| 4. Τέταρτο συνθηματικό..... | 10 |

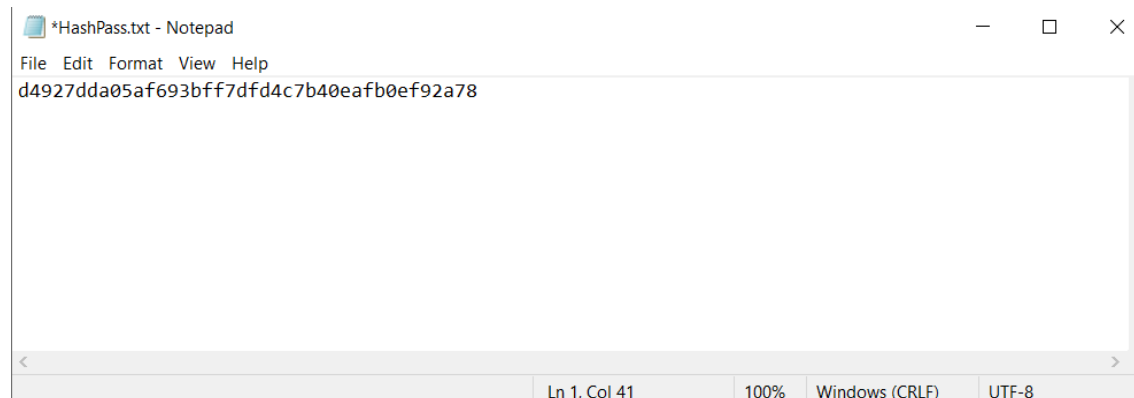
1. Πρώτο συνθηματικό

Στο πρώτο ερώτημα της εργασίας καλούμαστε, με βάση τα χαρακτηριστικά του «hash value», να βρούμε ποια συνάρτηση έχει χρησιμοποιηθεί και να εφαρμόσουμε την καταλληλότερη επίθεση με κάποιο «password cracker». Για να βρούμε τη συνάρτηση θα χρησιμοποιήσουμε ένα hash-identifier (<https://sourceforge.net/projects/hashidentifier/>) και για την επίθεση θα χρησιμοποιήσουμε το JohnTheRipper. Αναλυτικότερα:

- Θα βάλω το συνθηματικό (**d4927dda05af693bff7dfd4c7b40eafb0ef92a78**) στο hash-identifier και θα πάρω τη συνάρτηση που χρησιμοποιήθηκε για να παραχθεί.



- Αποθηκεύω το συνθηματικό σε ένα αρχείο



- Ύστερα θα κατευθυνθώ από το τερματικό μου στο φάκελο όπου έχω το JohnTheRipper και θα πραγματοποιήσω την επίθεση ορίζοντάς του και το είδος του Hash που χρησιμοποίησα, αφού το βρήκα στο προηγούμενο βήμα.

```
C:\JohnTheRipper\run>john --format=mysql-sha1 HashPass.txt ) Εντολή Τερματικού
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

C:\JohnTheRipper\run>john --format=mysql-sha1 C:\Users\30698\Desktop\HashPass.txt
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

C:\JohnTheRipper\run>john --format=raw-sha1 C:\Users\30698\Desktop\HashPass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 256/256 AVX2 8x])
Warning: no OpenMP support for this hash type, consider --fork=8
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:password.lst, rules:Wordlist
Warning: Only 6 candidates left, minimum 8 needed for performance.
Proceeding with incremental:ASCII
tiny (?) Ο κωδικός
1g 0:00:00:05 DONE 3/3 (2022-12-05 14:56) 0.1783g/s 361336p/s 361336c/s 361336C/s tine..tita
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed
```

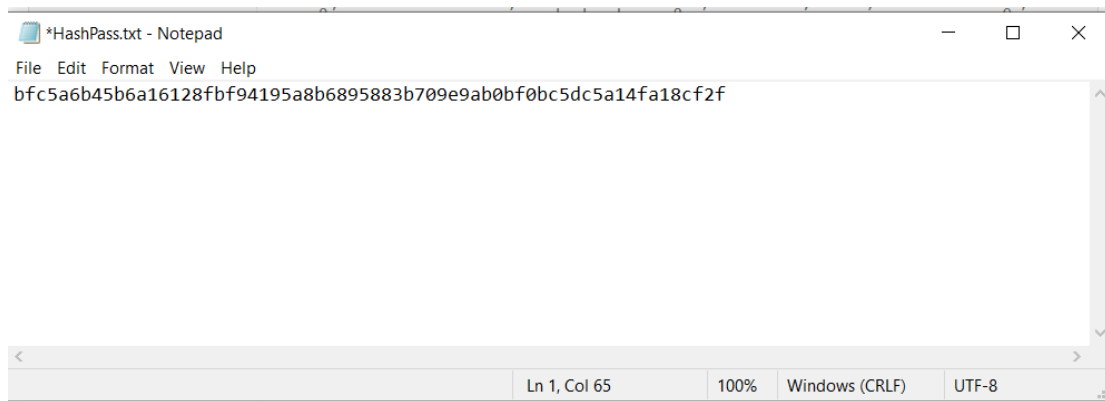
- Ο κωδικός είναι «tiny» και ο χρόνος που χρειάστηκε ήταν ελάχιστος όπως φαίνεται και στο τερματικό.

2. Δεύτερο συνθηματικό

Σε αυτό το ερώτημα, θα επαναλάβουμε την παραπάνω διαδικασία για ένα διαφορετικό Hash, το:

bfc5a6b45b6a16128fbf94195a8b6895883b709e9ab0bf0bc5dc5a14fa18cf2f .

Θα βρούμε την συνάρτηση που χρησιμοποιήθηκε και πάλι και θα βάλουμε το φάκελο στο JohnTheRipper ώστε να κάνει την πιο αποδοτική επίθεση.



```
C:\JohnTheRipper\run>john --format=raw-sha256 C:\Users\30698\Desktop\HashPass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:01:00 3/3 0g/s 26312Kp/s 26312Kc/s 26312KC/s scufrcub..scpmuis7
Session aborted
```

- Παρατηρώ ότι με τον απλό τρόπο παίρνει πάρα πολύ χρόνο καθώς δεν είναι εύκολο να σπάσω το Hash λόγω του sha-256. Αφού ξέρω όμως το μέγεθος του Password θα το περάσω ως όρισμα στον JohnTheRipper και θα ξαναδοκιμάσω. Πάλι παίρνει μεγάλο χρόνο για να ολοκληρωθεί.

```
C:\JohnTheRipper\run>john --format=raw-sha256 -max-len=12 C:\Users\30698\Desktop\HashPass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Proceeding with single, rules:Single, lengths:0-12
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:password.lst, rules:Wordlist, lengths: 0-12
Proceeding with incremental:ASCII, lengths: 0-12
Session aborted
```

3. Τρίτο συνθηματικό

Το hashed password που προσπαθούμε να σπάσουμε είναι το εξής:


23997786f8c60122a711f040a7acb4faa5ddd0681d5d658807ab0f9e987d6042

Από την εκφώνηση γνωρίζουμε πως το password είναι μία Ελληνική λέξη. Για αυτό τον λόγο επιλέγουμε να εφαρμόσουμε dictionary attack.

Μεταβαίνουμε στην ιστοσελίδα:

https://hashes.com/en/tools/hash_identifier

για να εντοπίσουμε τον τύπο του hash αλγόριθμου που έχει εφαρμοστεί στο password.

 Identify hash types

Identify and detect unknown hashes using this tool. This page will tell you what type of hash a given string is. If you want to attempt to Decrypt them, click this link instead. [Decrypt Hashes](#)

Hashes (max. 25 separated by newline, format 'hash[:salt]')

23997786f8c60122a711f040a7acb4faa5ddd0681d5d658807ab0f9e987d6042

☐ Include all possibilities (expert mode)

SUBMIT & IDENTIFY

Copy-paste του hashed password

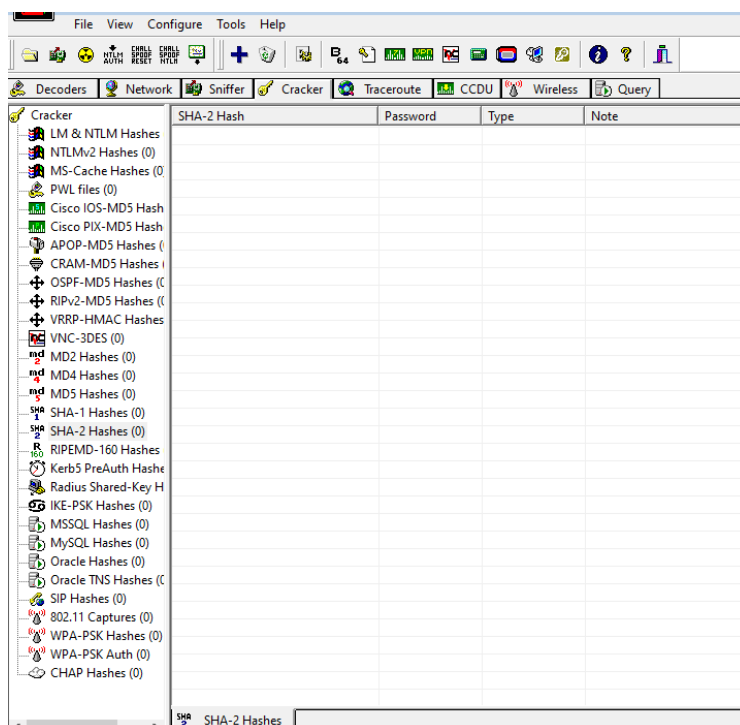
Πατάμε το κουμπί «SUBMIT & IDENTIFY» για να δούμε τα αποτελέσματα:

✓ Possible identifications: [Decrypt Hashes](#)

23997786f8c60122a711f040a7acb4faa5ddd0681d5d658807ab0f9e987d6042 - Possible algorithms: SHA256

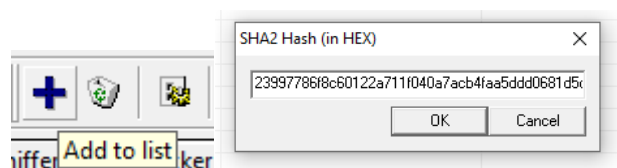
Πιθανός hash αλγόριθμος: SHA256

Αφού εντοπίσουμε τον hash αλγόριθμο, ανοίγουμε το περιβάλλον Cain και επιλέγουμε το tab «Cracker»:

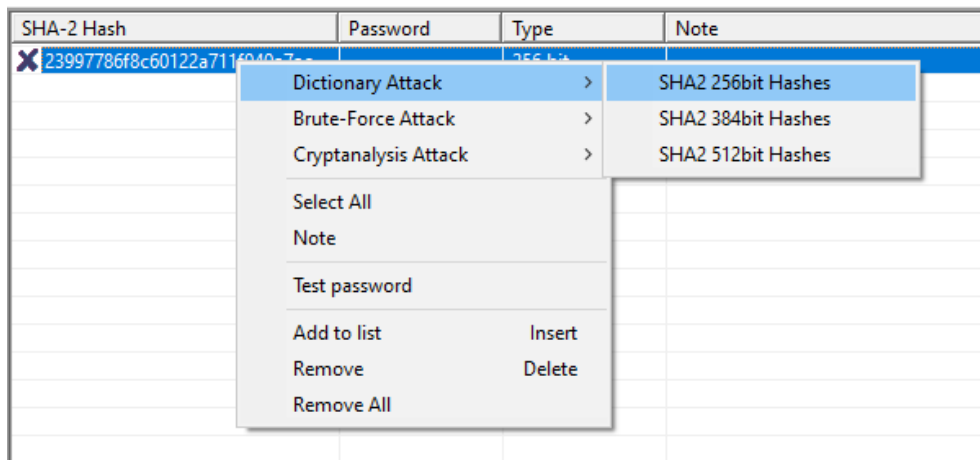


Cracker tab στο Cain

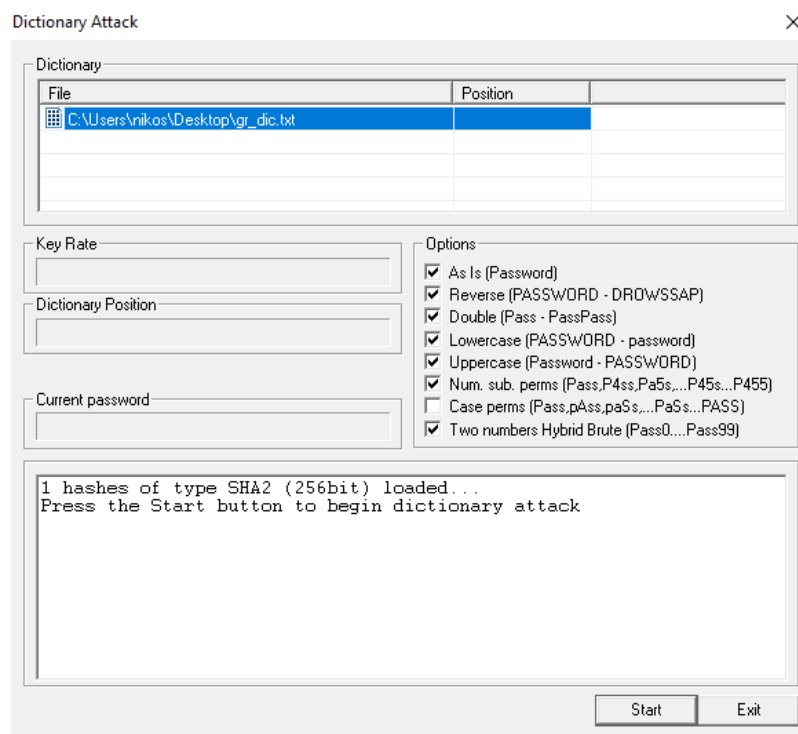
Επιλέγουμε το SHA-2 από το αριστερό μενού καθώς ο SHA256 είναι αυτής της κατηγορίας. Πατάμε το κουμπί «+» για να εισάγουμε το hashed password:



Στη συνέχεια κάνουμε δεξί κλικ στο hashed password→Dictionary Attack→SHA2 256 Hashes:

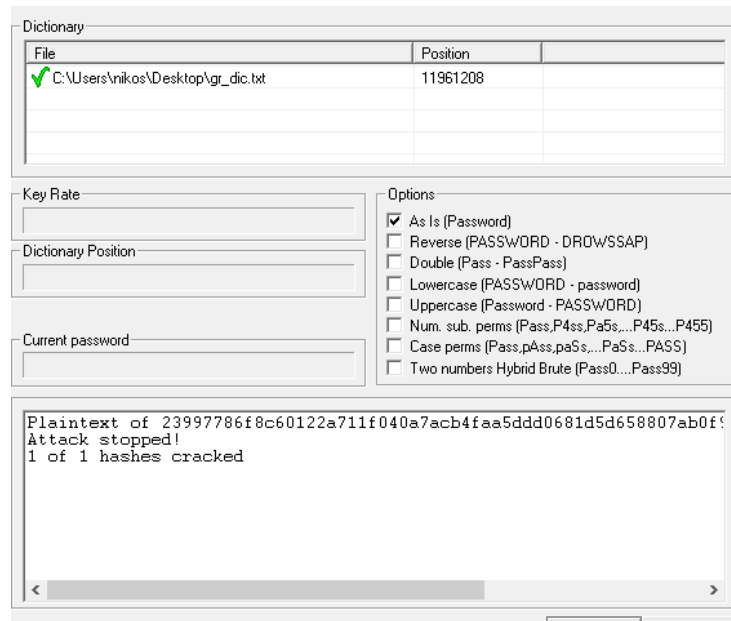


Στο παράθυρο που εμφανίζεται κάνουμε δεξί κλικ στη περιοχή «Dictionary» και επιλέγουμε «Add to list» για να εισάγουμε το αρχείο με το ελληνικό λεξικό:

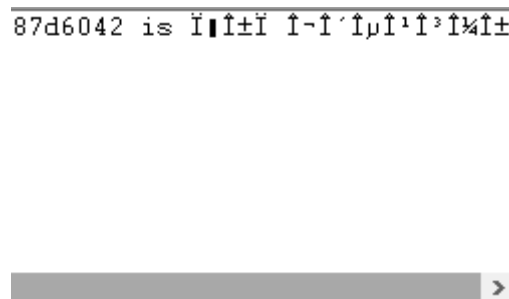


Εισαγωγή του αρχείου gr_dic.txt από τα χρήσιμα έγγραφα του μαθήματος(gunet2)

Αλλάζοντας τα Options αφήνοντας μόνο το πρώτο κουτάκι τικαρισμένο, πατάμε «Start». Σε μερικά δευτερόλεπτα έχουμε αποτέλεσμα:



Επιτυχής cracking του password



Το password δεν έχει εκτυπώσιμη τιμή στο Cain καθώς χρησιμοποιούνται Unicode χαρακτήρες(Ελληνικά)

Να σημειωθεί, πως κρατήσαμε μόνο το option «As Is (Password)» καθώς γνωρίζουμε πώς το password είναι μια λέξη του λεξικού που βγάζει νόημα. Έτσι γλυτώνουμε να ελέγξουμε πολλές περιπτώσεις.

Σαν έξτρα κομμάτι αυτής της άσκησης, υλοποιήθηκε ένα script σε python που διαβάσει μία μία τις λέξεις από το αρχείο λεξικό και βρίσκει την sha256 τιμή τους. Αν αυτή συμπίπτει με αυτήν που θέλουμε τότε τυπώνει στην κονσόλα «found» και σταματάει:

```
παράδεδοι  
f8ce251e628185e01968d18a8769e86507d1bfba32670de461331b6b8b21724f  
παράδεδοι  
a95f9bd4f8f07b6a87ccd7989c098879a2f707085683c534518e8c762ac0126a  
παράδεδοι  
5b2683fb75340bfafaa5ca9c87b979dcde5062802c428a19ec43e2460bc1d2b6  
παράδεδοι  
c1462ef83c579d10f7c353850de5818dd424be0d4003af0bbab41624a334b56b  
παράδεδοι  
9501077618534fcae51f18c2a5f7da1aa59e9caa9bc04ca0a3c4e96349e9b1f6  
παράδεδοι  
92f2203c001e7cdb04edf912fe59171ac05f103c476177ed39bc0ef8e5fad241  
παράδεδοι  
23997786f8c60122a711f040a7acb4faa5ddd0681d5d658807ab0f9e987d6042  
found  
PS C:\Users\nikos\Desktop\python\find_hash>
```

To password είναι «παράδειγμα»

To script:

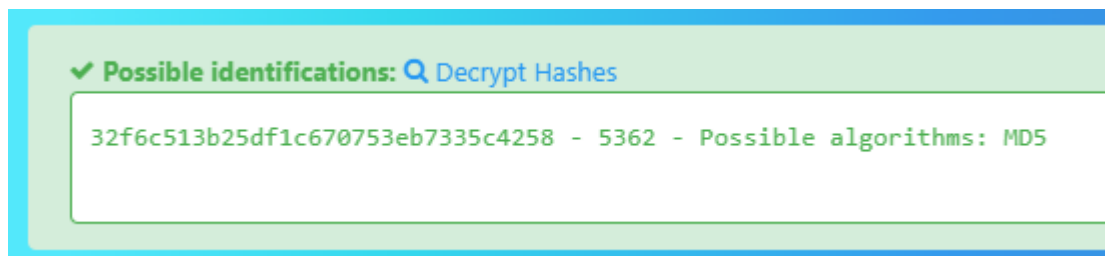
```
import hashlib  
import io  
  
#hash value we are looking for  
wanted_hash = "23997786f8c60122a711f040a7acb4faa5ddd0681d5d658807ab0f9e987d6042"  
  
for word in io.open("gr_dic.txt", mode="r", encoding="utf-8").readlines():  
    m = hashlib.sha256()  
    m.update(word[:-1].encode()) #ascii to bytes  
    print(word[:-1])  
    print(m.digest().hex()) #byte form of hash, then hex form  
    print()  
    if(m.digest().hex() == wanted_hash):  
        print("found")  
        break
```

4. Τέταρτο συνθηματικό

Στην συγκεκριμένη περίπτωση προσπαθήσαμε να σπάσουμε το συνθηματικό: **32f6c513b25df1c670753eb7335c4258**, το οποίο είναι ένα PIN με 4 ακέραια ψηφία. Επομένως, γνωρίζουμε ότι οι αποδεκτές τιμές του κωδικού είναι τα ψηφία 0,1,2,3,4,5,6,7,8 και 9.

Το πρώτο βήμα είναι να βρούμε με ποιον αλγόριθμο έχει γίνει hash το μήνυμά μας. Μεταβαίνοντας στην ιστοσελίδα

https://hashes.com/en/tools/hash_identifier και πληκτρολογώντας το hash, η εφαρμογή μας απαντάει πως έχει γίνει χρήση του αλγορίθμου MD5.



Επομένως, χρησιμοποιώντας τις παραπάνω πληροφορίες, μπορούμε να φτιάξουμε ένα rainbow table χρησιμοποιώντας κωδικοποίηση MD5, αλφάβητο "numeric" και μήκος κωδικού 4. Η σχετική εντολή για δημιουργία του πίνακα αυτού φαίνεται παρακάτω:

```
C:\Users\strat\OneDrive\Έγγραφα\rainbowcrack\rainbowcrack-1.8-win64>rtgen md5 numeric 4 4 0 3800 10000 0
rainbow table md5_numeric#4-4_0_3800x10000_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: 4 - 4
reduce offset:        0x00000000
plaintext total:      10000

sequential starting point begin from 0 (0x0000000000000000)
generating...
10000 of 10000 rainbow chains generated (0 m 2.7 s)

C:\Users\strat\OneDrive\Έγγραφα\rainbowcrack\rainbowcrack-1.8-win64>rtsort .
.\md5_numeric#4-4_0_3800x10000_0.rt:
5164945408 bytes memory available
loading data...
sorting data...
writing sorted data...
```

Δημιουργία rainbow table με το rtgen

Στην συνέχεια, μπορούμε με μία εντολή (φαίνεται στην εικόνα παρακάτω) να σπάσουμε το hash, να βρούμε δηλαδή σε ποιον αριθμό που έχει καταχωρηθεί στον πίνακα (rainbow table) αντιστοιχεί το εν λόγω hash που έχουμε να σπάσουμε.

```
C:\Users\strat\OneDrive\Εγγραφα\rainbowcrack\rainbowcrack-1.8-win64>rcrack . -h 32f6c513b25df1c670753eb7335c4258
1 rainbow tables found
memory available: 3782619955 bytes
memory for rainbow chain traverse: 60800 bytes per hash, 60800 bytes for 1 hashes
memory for rainbow table buffer: 2 x 160016 bytes
disk: .\md5_numeric#4-4_0_3800x10000_0.rt: 160000 bytes read
disk: finished reading all files
plaintext of 32f6c513b25df1c670753eb7335c4258 is 5362

statistics
-----
plaintext found:                1 of 1
total time:                    0.63 s
time of chain traverse:         0.61 s
time of alarm check:           0.00 s
time of disk read:             0.00 s
hash & reduce calculation of chain traverse: 7216200
hash & reduce calculation of alarm check:    2212
number of alarm:                2212
performance of chain traverse:   11.83 million/s
performance of alarm check:      2.21 million/s

result
-----
32f6c513b25df1c670753eb7335c4258  5362  hex:35333632
```

Συνολικός χρόνος που χρειάστηκε: 0.63 δευτερόλεπτα. Ο 4-ψήφιος κωδικός είναι ο αριθμός 5362!