NETSEC

Ramblings of a NetSec addict

Cracking Hashes (oclHashcat)

Peleus

Sometimes you obtain passwords that are in a hashed form. Due to the mathematical properties of (secure) hashes there are limited ways of recovering the plain text. Primarily this will be through brute force, or alternatively using word lists. oclHashcat is a fantastic hash cracking tool that takes advantage of your GPU to dramatically ramp up your hash calculating (and hence cracking) ability. For a dictionary attack the hash of each word in the dictionary is calculated and compared against your target hash. If the hashes match clearly the original value of the target must have been the same, hence the password is revealed. If the hash does not match the next word is calculated until the list is exhausted. Word lists have the advantage of being a lot higher more likely to contain the password (passwords are usually derived from humans), so your limited computing power is more targeted with a word list. The downside is that if the word is not in the list the hash will not be broken. Brute force attacks alternatively will always eventually get the hash value, but typically it takes so long to cycle through possible values it's infeasible. (Hint: Calculate how many combinations can be made with 8 characters of numbers / upper / lower characters. How long would it take to cycle through all of these if we were guessing at 1 million guesses a second?)

Tool

oclHashcat – http://hashcat.net/oclhashcat/

Basic Syntax

```
oclHashcat -m <hash type><hash list> <word list> -o <found list> --remove
```

Break Down

-m : signifies the type of hash being attacked. A list of hash types and their value can be found here – http://hashcat.net/wiki/doku.php?id=example_hashes

: a text file containing a list of all the hashes you wish to attack. Can be an individual hash if you wish.

: a file containing likely passwords.

-o : store recovered values in a separate file

–remove : remove successfully recovered hashes from the original list. Useful for running the same file against several lists without having to waste time searching for hashes already broken.

Example

```
oclhashcat -m 500 example500.hash /usr/share/wordlists/rockyou.txt -o found.txt
```

This example took 11 seconds to test 14,343,297 passwords against a hash. The hash was not found.

Advanced Reading

There are ton’s more options you can do with hash cracking. Brute force, mask attacks, word list mangling with John the Ripper, customized word lists based off company websites etc.

Filed Under: Passwords
Tagged With: cracking, hash, oclHashcat

Copyright © 2017 · Genesis Sample Theme on Genesis Framework · WordPress · Log in