RainbowCrack (improved, multi-threaded) - Making a Faster Cryptanalytic Time-Memory Trade-Off

by Martin Westergaard <martinwj2005@gmail.com>

multi-threaded and enhanced by neinbrucke

\*nix/64-bit compatibility and co-maintainer - James Nobis <quel@quelrod.net>

http://www.freerainbowtables.com/

All code/binaries are under GPL2 Copyright at a minimum

original code by Zhu Shuanglei <shuanglei@hotmail.com>

usage: rcracki\_mt -h hash rainbow\_table\_pathname

rcracki\_mt -l hash\_list\_file rainbow\_table\_pathname

rcracki\_mt -f pwdump\_file rainbow\_table\_pathname

rcracki\_mt -c lst\_file rainbow\_table\_pathname

-h hash: 使用原始散列作为输入

-l hash\_list\_file: 使用哈希列表文件作为输入，每一个行中的哈希值

-f pwdump\_file: 使用pwdump文件作为输入，处理LAN Manager散列只

-c lst\_file: 使用。LST（该隐格式）文件作为输入

-r [-s session\_name]: 从以前的会话中恢复，可选的会话名称

rainbow\_table\_pathname: 路径名（S）的彩虹表（S）

Extra options: -t [nr]使用这个线程/内核，默认值为1

-o [output\_file] 写（临时）结果到这个文件

-s [session\_name] write session data with this name

-k keep precalculation on disk

-d run sha1 hashes against mysqlsha1 tables

-m [megabytes] limit memory usage

-v show debug information

example: rcracki\_mt -h 5d41402abc4b2a76b9719d911017c592 -t 2 [path]/MD5

rcracki\_mt -l hash.txt [path\_to\_specific\_table]/\*

rcracki\_mt -f hash.txt -t 4 -o results.txt \*.rti