Core dump overflow

Core dump in progress...

- RSS

Search

Navigate... Navigate...

Navigate...

- Blog
- Where to start
- Book corner
- Archives

Kali tools catalog - Password Attacks

Apr 4th, 2015 | Comments

Tools for password related attacks

GPU Tools

oclhashcat

```
Worlds fastest password cracker
Worlds first and only GPGPU based rule engine
Free
Multi-GPU (up to 128 gpus)
Multi-Hash (up to 100 million hashes)
Multi-OS (Linux & Windows native binaries)
Multi-Platform (OpenCL & CUDA support)
Multi-Algo (see below)
Low resource utilization, you can still watch movies or play games while cracking
Focuses highly iterated modern hashes
Focuses dictionary based attacks
Supports distributed cracking
Supports pause / resume while cracking
Supports sessions
Supports restore
Supports reading words from file
Supports reading words from stdin
Supports hex-salt
Supports hex-charset
Built-in benchmarking system
Integrated thermal watchdog
150+ Algorithms implemented with performance in mind
... and much more
```

Screenshot from the official site showing it in action:

```
root@sf:~/oclHashcat-1.32# ./oclHashcat64.bin -m 8900 -a 3 ?a?a?a?a?a?a?a -n 512 hash

Device #1: Hawaii, 3072MB, 1000Mhz, 44MCU

Hashes: 1 hashes; 1 unique digests, 1 unique salts
Bitmaps: 8 bits, 256 entries, 0x000000ff mask, 1024 bytes
Applicable Optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt
* Brute-Force
Watchdog: Temperature abort trigger set to 95c
Watchdog: Temperature retain trigger set to 80c

SCRYPT tmto optimizer value set to: 2

Device #1: Kernel ./kernels/4098/m08900_1024_1_1_2.Hawaii_1573.4_1573.4 (VM).kernel
Device #1: Kernel ./kernels/4098/markov_le_v1.Hawaii_1573.4_1573.4 (VM).kernel
Device #1: Kernel ./kernels/4098/amp_a3_v1.Hawaii_1573.4_1573.4 (VM).kernel

SCRYPT:1024:1:1:NjM0OA==:TWU9Rx1o8AnBlA7CC/jUoTsLE6RaVZkuLGzDudgdJaA=:gra2020

Session.Name...: oclHashcat
Status.........: Cracked
Input.Mode.....: Mask (?a?a?a?a?a?a?a) [7]
Hash.Target....: SCRYPT:1024:1:1:NjM0OA==:TWU9Rx1o8AnBlA7CC/jUoTsLE6RaVZkuLGzDudgdJaA=
Hash.Type......: scrypt
Time.Started...: Thu Jan 15 15:59:44 2015 (5 mins, 46 secs)
Speed.GPU.#1...:   862.1 kH/s
Recovered......: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.......: 295567360/69833729609375 (0.00%)
Skipped........: 0/295567360 (0.00%)
Rejected.......: 0/295567360 (0.00%)
HWMon.GPU.#1...:  0% Util, 76c Temp, 100% Fan

Started: Thu Jan 15 15:59:44 2015
Stopped: Thu Jan 15 16:05:31 2015
```

pyrit

Pyrit exploits the computational power of many-core- and GPGPU-platforms to create massive databases, pre-computing part of the WPA/WPA2-PSK authentication phase in a space-time tradeoff. It is a powerful attack against one of the world's most used security protocols.

```
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Usage: pyrit [options] command

Recognized options:
  -b               : Filters AccessPoint by BSSID
  -e               : Filters AccessPoint by ESSID
  -h               : Print help for a certain command
  -i               : Filename for input ('-' is stdin)
  -o               : Filename for output ('-' is stdout)
  -r               : Packet capture source in pcap-format
  -u               : URL of the storage-system to use
  --all-handshakes : Use all handshakes instead of the best one

Recognized commands:
  analyze                 : Analyze a packet-capture file
  attack_batch            : Attack a handshake with PMKs/passwords from the db
  attack_cowpatty         : Attack a handshake with PMKs from a cowpatty-file
  attack_db               : Attack a handshake with PMKs from the db
  attack_passthrough      : Attack a handshake with passwords from a file
  batch                   : Batchprocess the database
  benchmark               : Determine performance of available cores
  benchmark_long          : Longer and more accurate version of benchmark (~10 minutes)
  check_db                : Check the database for errors
  create_essid            : Create a new ESSID
  delete_essid            : Delete a ESSID from the database
  eval                    : Count the available passwords and matching results
  export_cowpatty         : Export results to a new cowpatty file
  export_hashdb           : Export results to an airolib database
  export_passwords        : Export passwords to a file
  help                    : Print general help
  import_passwords        : Import passwords from a file-like source
  import_unique_passwords : Import unique passwords from a file-like source
  list_cores              : List available cores
  list_essids             : List all ESSIDs but don't count matching results
```

```
37  passthrough            : Compute PMKs and write results to a file
38  relay                  : Relay a storage-url via RPC
39  selftest               : Test hardware to ensure it computes correct results
40  serve                  : Serve local hardware to other Pyrit clients
41  strip                  : Strip packet-capture files to the relevant packets
42  stripLive              : Capture relevant packets from a live capture-source
43  verify                 : Verify 10% of the results by recomputation
```

Offline Attacks

cachedump

Recover Windows password cache entries

```
1 usage: /usr/bin/cachedump <system hive> <security hive>
```

chntpw

chntpw is a utility to view some information and change user passwords in a Windows NT/2000 SAM userdatabase file, usually located at \WINDOWS\system32\config\SAM on the Windows file system. It is not necessary to know the old passwords to reset them. In addition it contains a simple registry editor (same size data writes) and hex-editor with which the information contained in a registry file can be browsed and modified.

```
chntpw version 0.99.6 080526 (sixtyfour), (c) Petter N Hagen
chntpw: change password of a user in a NT/2k/XP/2k3/Vista SAM file, or invoke registry editor.
chntpw [OPTIONS] <samfile> [systemfile] [securityfile] [otherreghive] [...]
 -h           This message
 -u <user>    Username to change, Administrator is default
 -l           list all users in SAM file
 -i           Interactive. List users (as -l) then ask for username to change
 -e           Registry editor. Now with full write support!
 -d           Enter buffer debugger instead (hex editor),
 -t           Trace. Show hexdump of structs/segments. (deprecated debug function)
 -v           Be a little more verbose (for debuging)
 -L           Write names of changed files to /tmp/changed
 -N           No allocation mode. Only (old style) same length overwrites possible
See readme file on how to get to the registry files, and what they are.
Source/binary freely distributable under GPL v2 license. See README for details.
NOTE: This program is somewhat hackish! You are on your own!
root@kali:~#
```

cmospwd

A cmos/bios password recovery tool

```
Terminal
File  Edit  View  Search  Terminal  Help
CmosPwd - BIOS Cracker 5.0, October 2007, Copyright 1996-2007
GRENIER Christophe, grenier@cgsecurity.org
http://www.cgsecurity.org/

Keyboard : US
Acer/IBM                      [   - ][        ]
AMI BIOS                      []
AMI WinBIOS (12/15/93)        []
AMI WinBIOS 2.5               [][][][]
AMI ?                         [][    ][][     ][]
Award 4.5x/6.0                [10101331][000100][000100]
Award 4.5x/6.0                [000100][000100][000100][000100]
Award Medallion 6.0           [1200031][1120210][000100][33332123]
Award 6.0                     [][][][]
Compaq (1992)                 []
Compaq DeskPro                [   ][   2]
Compaq                        [][]
DTK                           [][_47n ]
IBM (PS/2, Activa ...)        [ ][]
IBM Thinkpad boot pwd         []
Thinkpad x20/570/t20 EEPROM   [][]
Thinkpad 560x EEPROM          [][]
Thinkpad 765/380z EEPROM      [][]
IBM 300 GL                    [ ]
Packard Bell Supervisor/User  [        ] [        ]
Press Enter key to continue
```

crunch

Generate wordlists from a character set

```
crunch version 3.6

Crunch can create a wordlist based on criteria you specify.  The outout from crunch can be sent to the screen, f

Usage: crunch <min> <max> [options]
where min and max are numbers

Please refer to the man page for instructions and examples on how to use crunch.
```

dictstat

Generate dictionary file statistics

```
[?] Psyco is not available. Install Psyco on 32-bit systems for faster parsing.
Usage: dictstat [options] passwords.txt

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -l 8, --length=8      Password length filter.
  -c loweralpha, --charset=loweralpha
                        Password charset filter.
  -m stringdigit, --mask=stringdigit
                        Password mask filter
  -o masks.csv, --maskoutput=masks.csv
                        Save masks to a file
```

fcrackzip

Searches each zipfile given for encrypted files and tries to guess the password. All files must be encrypted with the same password, the more files you provide, the better.

```
fcrackzip version 1.0, a fast/free zip password cracker
written by Marc Lehmann <pcg@goof.com> You can find more info on
http://www.goof.com/pcg/marc/

USAGE: fcrackzip
         [-b|--brute-force]            use brute force algorithm
         [-D|--dictionary]             use a dictionary
         [-B|--benchmark]              execute a small benchmark
         [-c|--charset characterset]   use characters from charset
         [-h|--help]                   show this message
         [--version]                   show the version of this program
         [-V|--validate]               sanity-check the algortihm
         [-v|--verbose]                be more verbose
         [-p|--init-password string]   use string as initial password/file
         [-l|--length min-max]         check password with length min to max
         [-u|--use-unzip]              use unzip to weed out wrong passwords
         [-m|--method num]             use method number "num" (see below)
         [-2|--modulo r/m]             only calculcate 1/m of the password
         file...                    the zipfiles to crack

methods compiled in (* = default):

 0: cpmask
 1: zip1
*2: zip2, USE_MULT_TAB
```


hashcat

Advanced password recovery

```
root@kali:~# hashcat --help
hashcat, advanced password recovery

Usage: hashcat [options] hashfile [mask|wordfiles|directories]

=======
Options
=======

* General:

  -m,  --hash-type=NUM              Hash-type, see references below
  -a,  --attack-mode=NUM            Attack-mode, see references below
  -V,  --version                    Print version
  -h,  --help                       Print help
       --eula                       Print EULA
       --expire                     Print expiration date
       --quiet                      Suppress output

* Benchmark:

  -b,  --benchmark                  Run benchmark

* Misc:

       --hex-salt                   Assume salt is given in hex
       --hex-charset                Assume charset is given in hex
       --runtime=NUM                Abort session after NUM seconds of runtime

* Files:

  -o,  --outfile=FILE               Define outfile for recovered hash
       --outfile-format=NUM         Define outfile-format for recovered hash, see references below
       --outfile-autohex-disable    Disable the use of $HEX[] in output plains
  -p,  --separator=CHAR             Define separator char for hashlists/outfile
       --show                       Show cracked passwords only (see --username)
       --left                       Show uncracked passwords only (see --username)
       --username                   Enable ignoring of usernames in hashfile (Recommended: also use --show)
       --remove                     Enable remove of hash once it is cracked
       --stdout                     Stdout mode
       --potfile-disable            Do not write potfile
       --debug-mode=NUM             Defines the debug mode (hybrid only by using rules), see references below
       --debug-file=FILE            Output file for debugging rules (see --debug-mode)
```

```
  -e,  --salt-file=FILE              Salts-file for unsalted hashlists

* Resources:

  -c,  --segment-size=NUM            Size in MB to cache from the wordfile
  -n,  --threads=NUM                 Number of threads
  -s,  --words-skip=NUM              Skip number of words (for resume)
  -l,  --words-limit=NUM             Limit number of words (for distributed)

* Rules:

  -r,  --rules-file=FILE             Rules-file use: -r 1.rule
  -g,  --generate-rules=NUM          Generate NUM random rules
       --generate-rules-func-min=NUM Force NUM functions per random rule min
       --generate-rules-func-max=NUM Force NUM functions per random rule max
       --generate-rules-seed=NUM     Force RNG seed to NUM

* Custom charsets:

  -1,  --custom-charset1=CS          User-defined charsets
  -2,  --custom-charset2=CS          Example:
  -3,  --custom-charset3=CS          --custom-charset1=?dabcdef : sets charset ?1 to 0123456789abcdef
  -4,  --custom-charset4=CS          -2 mycharset.hcchr : sets charset ?2 to chars contained in file

* Toggle-Case attack-mode specific:

       --toggle-min=NUM              Number of alphas in dictionary minimum
       --toggle-max=NUM              Number of alphas in dictionary maximum

* Mask-attack attack-mode specific:

       --pw-min=NUM                  Password-length minimum
       --pw-max=NUM                  Password-length maximum

* Permutation attack-mode specific:

       --perm-min=NUM                Filter words shorter than NUM
       --perm-max=NUM                Filter words larger than NUM

* Table-Lookup attack-mode specific:

  -t,  --table-file=FILE             Table file
       --table-min=NUM               Number of chars in dictionary minimum
       --table-max=NUM               Number of chars in dictionary maximum

* Prince attack-mode specific:

       --pw-min=NUM                  Password-length minimum
       --pw-max=NUM                  Password-length maximum
       --elem-cnt-min=NUM            Minimum number of elements per chain
       --elem-cnt-max=NUM            Maximum number of elements per chain

==========
References
==========

* Outfile formats:

    1 = hash[:salt]
    2 = plain
    3 = hash[:salt]:plain
    4 = hex_plain
    5 = hash[:salt]:hex_plain
    6 = plain:hex_plain
    7 = hash[:salt]:plain:hex_plain
    8 = crackpos
    9 = hash[:salt]:crackpos
   10 = plain:crackpos
   11 = hash[:salt]:plain:crackpos
   12 = hex_plain:crackpos
   13 = hash[:salt]:hex_plain:crackpos
   14 = plain:hex_plain:crackpos
   15 = hash[:salt]:plain:hex_plain:crackpos

* Debug mode output formats (for hybrid mode only, by using rules):

    1 = save finding rule
    2 = save original word
    3 = save original word and finding rule
    4 = save original word, finding rule and modified plain

```

```
* Built-in charsets:

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

* Attack modes:

    0 = Straight
    1 = Combination
    2 = Toggle-Case
    3 = Brute-force
    4 = Permutation
    5 = Table-Lookup
    6 = Prince

* Hash types:

     0 = MD5
    10 = md5($pass.$salt)
    20 = md5($salt.$pass)
    30 = md5(unicode($pass).$salt)
    40 = md5($salt.unicode($pass))
    50 = HMAC-MD5 (key = $pass)
    60 = HMAC-MD5 (key = $salt)
   100 = SHA1
   110 = sha1($pass.$salt)
   120 = sha1($salt.$pass)
   130 = sha1(unicode($pass).$salt)
   140 = sha1($salt.unicode($pass))
   150 = HMAC-SHA1 (key = $pass)
   160 = HMAC-SHA1 (key = $salt)
   200 = MySQL323
   300 = MySQL4.1/MySQL5
   400 = phpass, MD5(Wordpress), MD5(phpBB3), MD5(Joomla)
   500 = md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
   900 = MD4
  1000 = NTLM
  1100 = Domain Cached Credentials, mscash
  1400 = SHA256
  1410 = sha256($pass.$salt)
  1420 = sha256($salt.$pass)
  1430 = sha256(unicode($pass).$salt)
  1440 = sha256($salt.unicode($pass))
  1450 = HMAC-SHA256 (key = $pass)
  1460 = HMAC-SHA256 (key = $salt)
  1600 = md5apr1, MD5(APR), Apache MD5
  1700 = SHA512
  1710 = sha512($pass.$salt)
  1720 = sha512($salt.$pass)
  1730 = sha512(unicode($pass).$salt)
  1740 = sha512($salt.unicode($pass))
  1750 = HMAC-SHA512 (key = $pass)
  1760 = HMAC-SHA512 (key = $salt)
  1800 = SHA-512(Unix)
  2400 = Cisco-PIX MD5
  2410 = Cisco-ASA MD5
  2500 = WPA/WPA2
  2600 = Double MD5
  3200 = bcrypt, Blowfish(OpenBSD)
  3300 = MD5(Sun)
  3500 = md5(md5(md5($pass)))
  3610 = md5(md5($salt).$pass)
  3710 = md5($salt.md5($pass))
  3720 = md5($pass.md5($salt))
  3810 = md5($salt.$pass.$salt)
  3910 = md5(md5($pass).md5($salt))
  4010 = md5($salt.md5($salt.$pass))
  4110 = md5($salt.md5($pass.$salt))
  4210 = md5($username.0.$pass)
  4300 = md5(strtoupper(md5($pass)))
  4400 = md5(sha1($pass))
  4500 = Double SHA1
  4600 = sha1(sha1(sha1($pass)))
  4700 = sha1(md5($pass))
  4710 = sha1($salt.$pass.$salt)
  4800 = MD5(Chap), iSCSI CHAP authentication
  5000 = SHA-3(Keccak)
```

```
  5100 = Half MD5
  5200 = Password Safe SHA-256
  5300 = IKE-PSK MD5
  5400 = IKE-PSK SHA1
  5500 = NetNTLMv1-VANILLA / NetNTLMv1-ESS
  5600 = NetNTLMv2
  5700 = Cisco-IOS SHA256
  5800 = Android PIN
  6300 = AIX {smd5}
  6400 = AIX {ssha256}
  6500 = AIX {ssha512}
  6700 = AIX {ssha1}
  6900 = GOST, GOST R 34.11-94
  7000 = Fortigate (FortiOS)
  7100 = OS X v10.8 / v10.9
  7200 = GRUB 2
  7300 = IPMI2 RAKP HMAC-SHA1
  7400 = sha256crypt, SHA256(Unix)
  7900 = Drupal7
  8400 = WBB3, Woltlab Burning Board 3
  8900 = scrypt
  9200 = Cisco $8$
  9300 = Cisco $9$
  9800 = Radmin2
 10000 = Django (PBKDF2-SHA256)
 10200 = Cram MD5
 10300 = SAP CODVN H (PWDSALTEDHASH) iSSHA-1
 99999 = Plaintext

* Specific hash types:

   11 = Joomla < 2.5.18
   12 = PostgreSQL
   21 = osCommerce, xt:Commerce
   23 = Skype
  101 = nsldap, SHA-1(Base64), Netscape LDAP SHA
  111 = nsldaps, SSHA-1(Base64), Netscape LDAP SSHA
  112 = Oracle 11g/12c
  121 = SMF > v1.1
  122 = OS X v10.4, v10.5, v10.6
  123 = EPi
  124 = Django (SHA-1)
  131 = MSSQL(2000)
  132 = MSSQL(2005)
  133 = PeopleSoft
  141 = EPiServer 6.x < v4
 1421 = hMailServer
 1441 = EPiServer 6.x > v4
 1711 = SSHA-512(Base64), LDAP {SSHA512}
 1722 = OS X v10.7
 1731 = MSSQL(2012 & 2014)
 2611 = vBulletin < v3.8.5
 2612 = PHPS
 2711 = vBulletin > v3.8.5
 2811 = IPB2+, MyBB1.2+
 3711 = Mediawiki B type
 3721 = WebEdition CMS
 7600 = Redmine Project Management Web App
```

hashid

hashID is a tool written in Python 3.x which supports the identification of over 200 unique hash types using regular expressions.

Usage example from the project's Github page:

```
$ ./hashid.py '$P$8ohUJ.1sdFw09/bMaAQPTGDNi2BIUt1'
Analyzing '$P$8ohUJ.1sdFw09/bMaAQPTGDNi2BIUt1'
[+] Wordpress ≥ v2.6.2
[+] Joomla ≥ v2.5.18
[+] PHPass' Portable Hash

$ ./hashid.py -mj '$racf$*AAAAAAAA*3c44ee7f409c9a9b'
Analyzing '$racf$*AAAAAAAA*3c44ee7f409c9a9b'
[+] RACF [Hashcat Mode: 8500][JtR Format: racf]

$ ./hashid.py hashes.txt
--File 'hashes.txt'--
Analyzing '*85ADE5DDF71E348162894C71D73324C043838751'
```

```
[+] MySQL5.x
[+] MySQL4.1
Analyzing '$2a$08$VPzNKPAY60FsAbnq.c.h5.XTCZtC1z.j3hnlDFGImN9FcpfR1QnLq'
[+] Blowfish(OpenBSD)
[+] Woltlab Burning Board 4.x
[+] bcrypt
--End of file 'hashes.txt'--
```

hash-identifier

Identify different types of hashes

```
   ######################################################################
   #                                                                    #
   #  __  __               __      _____      _____                    #
   #  /\ \/\ \            /\ \     /\__  _\   /\  _ `\                 #
   #  \ \ \_\ \     __      ____ \ \ \___ \/_/\ \/  \ \ \/\ \ \    #
   #   \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \ \       #
   #    \ \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \ \      #
   #     \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #      \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.1 #
   #                                By Zion3R #
   #                           www.Blackploit.com #
   #                              Root@Blackploit.com #
   ######################################################################

   ----------------------------------------------------------------------
 HASH:
```

john

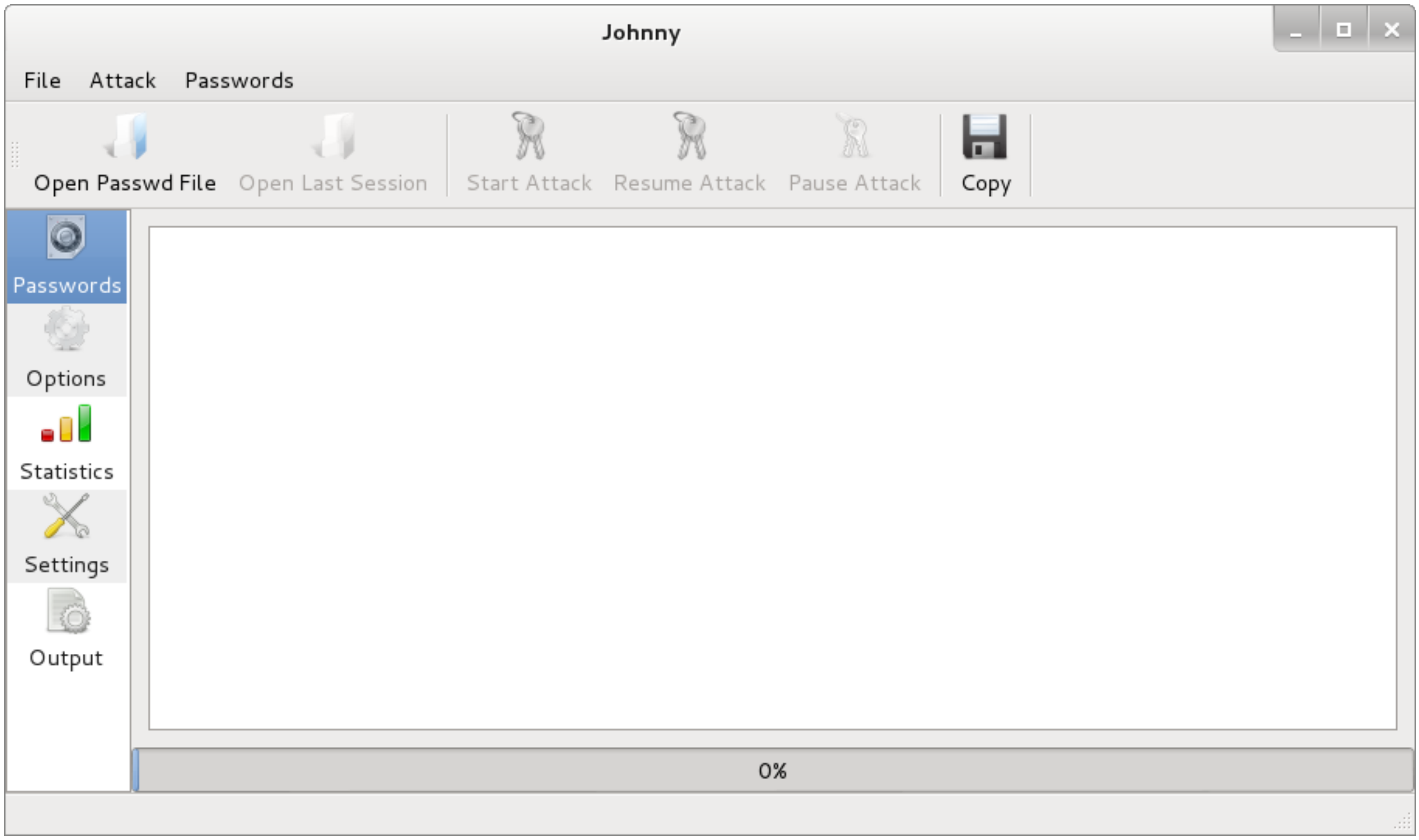
John the Ripper is a tool to find weak passwords of users in a server. John can use a dictionary or some search pattern as well as a password file to check for passwords. John supports different cracking modes and understands many ciphertext formats, like several DES variants, MD5 and blowfish. It can also be used to extract AFS and Windows NT passwords.

```
John the Ripper password cracker, ver: 1.7.9-jumbo-7_omp [linux-x86-64]
Copyright (c) 1996-2012 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--config=FILE             use FILE instead of john.conf or john.ini
--single[=SECTION]        "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                  --pipe  like --stdin, but bulk reads, and allows rules
--loopback[=FILE]         like --wordlist, but fetch words from a .pot file
--dupe-suppression        suppress all dupes in wordlist (and force preload)
--encoding=NAME           input data is non-ascii (eg. UTF-8, ISO-8859-1).
                          For a full list of NAME use --list=encodings
--rules[=SECTION]         enable word mangling rules for wordlist modes
--incremental[=MODE]      "incremental" mode [using section MODE]
--markov[=OPTIONS]        "Markov" mode (see doc/MARKOV)
--external=MODE           external mode or word filter
--stdout[=LENGTH]         just output candidate passwords [cut at LENGTH]
--restore[=NAME]          restore an interrupted session [called NAME]
--session=NAME            give a new session the NAME
--status[=NAME]           print status of a session [called NAME]
--make-charset=FILE       make a charset file. It will be overwritten
--show[=LEFT]             show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]             run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]      load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]    load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]    load salts with[out] COUNT [to MAX] hashes
--pot=NAME                pot file to use
--format=NAME             force hash type NAME: afs bf bfegg bsdi crc32 crypt
                          des django dmd5 dominosec dragonfly3-32 dragonfly3-64
                          dragonfly4-32 dragonfly4-64 drupal7 dummy dynamic_n
                          epi episerver gost hdaa hmac-md5 hmac-sha1
                          hmac-sha224 hmac-sha256 hmac-sha384 hmac-sha512
                          hmailserver ipb2 keepass keychain krb4 krb5 lm lotus5
                          md4-gen md5 md5ns mediawiki mscash mscash2 mschapv2
                          mskrb5 mssql mssql05 mysql mysql-sha1 nethalflm netlm
                          netlmv2 netntlm netntlmv2 nsldap nt nt2 odf office
                          oracle oracle11 osc pdf phpass phps pix-md5 pkzip po
                          pwsafe racf rar raw-md4 raw-md5 raw-md5u raw-sha
                          raw-sha1 raw-sha1-linkedin raw-sha1-ng raw-sha224
```

```
                              raw-sha256 raw-sha384 raw-sha512 salted-sha1 sapb
                              sapg sha1-gen sha256crypt sha512crypt sip ssh
                              sybasease trip vnc wbb3 wpapsk xsha xsha512 zip
--list=WHAT                   list capabilities, see --list=help or doc/OPTIONS
--save-memory=LEVEL           enable memory saving, at LEVEL 1..3
--mem-file-size=SIZE          size threshold for wordlist preload (default 5 MB)
--nolog                       disables creation and writing to john.log file
--crack-status                emit a status line whenever a password is cracked
--max-run-time=N              gracefully exit after this many seconds
--regen-lost-salts=N          regenerate lost salts (see doc/OPTIONS)
--plugin=NAME[,..]            load this (these) dynamic plugin(s)
```

johnny

GUI for the John the Ripper password cracking tool.



lsadump

Dump LSA secrets

[Usage example from Kali site](#):

```
root@kali:~# lsadump system security
_SC_ALG

_SC_Dnscache

_SC_upnphost

20ed87e2-3b82-4114-81f9-5e219ed4c481-SALEMHELPACCOUNT

_SC_WebClient

_SC_RpcLocator

0083343a-f925-4ed7-b1d6-d95d17a0b57b-RemoteDesktopHelpAssistantSID
0000   01 05 00 00 00 00 00 05 15 00 00 00 B6 44 E4 23    .............D.#
0010   F4 50 BA 74 07 E5 3B 2B E8 03 00 00                .P.t..;+....

0083343a-f925-4ed7-b1d6-d95d17a0b57b-RemoteDesktopHelpAssistantAccount
0000   00 38 00 48 00 6F 00 31 00 49 45 00 4A 00 26 00    E.J.&.8.H.o.1.I.
```

```
0010   00 63 00 72 00 48 00 68 00 53 6B 00 00 00          h.S.c.r.H.k...

_SC_MSDTC

_SC_SSDPSRV

_SC_Alerter

_SC_RpcSs

_SC_LmHosts

_SC_BthServ
```

maskgen

Generate hashcat masks

```
Usage: maskgen [options] masksfile.csv

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  --minlength=8         Minimum password length
  --maxlength=8         Maximum password length
  --mintime=MINTIME     Minimum time to crack
  --maxtime=MAXTIME     Maximum time to crack
  --complexity=COMPLEXITY
                        maximum password complexity
  --occurence=OCCURENCE
                        minimum times mask was used
  --checkmask=?u?l ?l ?l ?l ?l ?d
                        check mask coverage
  --showmasks           Show matching masks
  --pps=1000000000      Passwords per Second
```

multiforcer

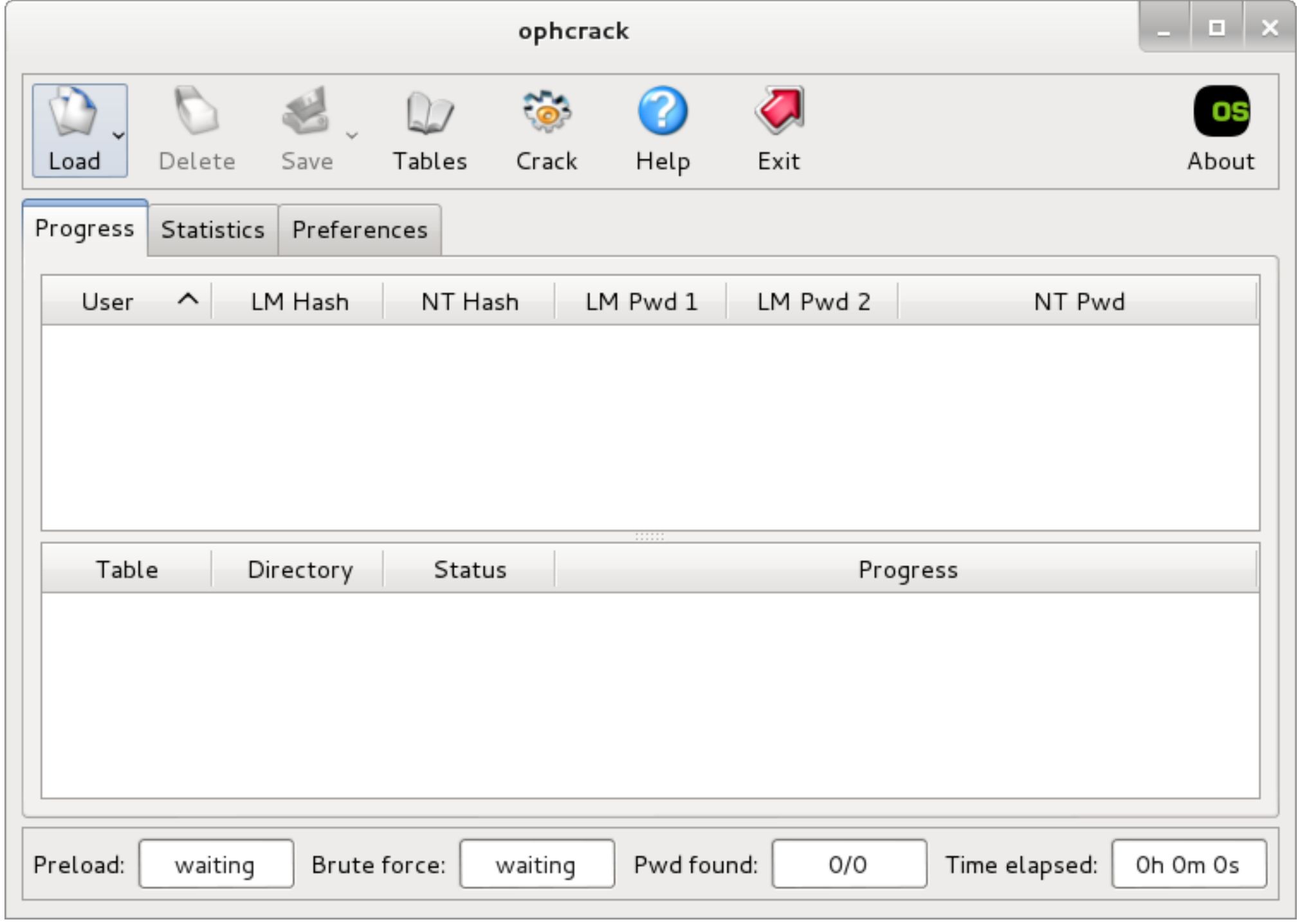
A CUDA & OpenCL accelerated rainbow table implementation from the ground up, and a CUDA hash brute forcing tool with support for many hash types including MD5, SHA1, LM, NTLM, and lots more.

Basic command line parameters:

```
-h / --hashtype [hash type] (required) This specifies the hash type to search. See the wiki for a current list c
-c / --charsetfile <filename> This specifies the charset file for single charset use.
-u / --charsetfilemulti <filename> This specifies the charset file for per-position charset use.
-o / --outputfile (optional) This specifies the output for found hashes. The file will be appended, not overwrit
-f / --hashfile (required) This specifies the file of hashes. Hashes should be in ASCII-hex format (as they are
--min / --max (required) These set the minimum and maximum password lengths to search. Lengths of 0 through 14 a
-m / --ms (optional) This specifies the target kernel time, in milliseconds (1/1000th of a second). When using a
-l / --lookup (optional) Use a 512MB chunk of GPU RAM to improve performance on very large hashlists. Requires a
```

ophcrack

A Microsoft Windows password cracker using rainbow tables



ophcrack-cli

Command line interface for ophcrack

```
ophcrack 3.4.0 by Objectif Securite (http://www.objectif-securite.ch)

Usage: ophcrack [OPTIONS]
Cracks Windows passwords with Rainbow tables

  -a              disable audit mode (default)
  -A              enable audit mode
  -b              disable bruteforce
  -B              enable bruteforce (default)
  -c config_file  specify the config file to use
  -D              display (lots of!) debugging information
  -d dir          specify tables base directory
  -e              do not display empty passwords
  -f file         load hashes from the specified file (pwdump or session)
  -g              disable GUI
  -h              display this information
  -i              hide usernames
  -I              show usernames (default)
  -l file         log all output to the specified file
  -n num          specify the number of threads to use
  -o file         write cracking output to file in pwdump format
  -p num          preload (0 none, 1 index, 2 index+end, 3 all default)
  -q              quiet mode
  -r              launch the cracking when ophcrack starts (GUI only)
  -s              disable session auto-saving
  -S session_file specify the file to use to automatically save the progress of the search
  -u              display statistics when cracking ends
  -t table1[,a[,b,...]][:table2[,a[,b,...]]]
                  specify which table to use in the directory given by -d
  -v              verbose
  -w dir          load hashes from encrypted SAM file in directory dir
  -x file         export data in CSV format to file


Example:  ophcrack -g -d /path/to/tables -t xp_free_fast,0,3:vista_free -f in.txt

     Launch ophcrack in command line using tables 0 and 3 in
     /path/to/tables/xp_free_fast and all tables in /path/to/tables/vista_free
```

```
      and cracks hashes from pwdump file in.txt
```

policygen

Generate hashcat masks

```
Usage: policygen [options]

Type --help for more options

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  --length=8            Password length
  -o masks.txt, --output=masks.txt
                        Save masks to a file
  --pps=1000000000      Passwords per Second
  -v, --verbose

  Password Policy:
    Define the minimum (or maximum) password strength policy that you
    would like to test

    --mindigits=1       Minimum number of digits
    --minlower=1        Minimum number of lower-case characters
    --minupper=1        Minimum number of upper-case characters
    --minspecial=1      Minimum number of special characters
    --maxdigits=3       Maximum number of digits
    --maxlower=3        Maximum number of lower-case characters
    --maxupper=3        Maximum number of upper-case characters
    --maxspecial=3      Maximum number of special characters
```

pwdump

Dump password hashes

[Usage example from Kali site](#):

```
root@kali:~# pwdump system sam
Administrator:500:41aa818b512a8c0e72381e4c174e281b:1896d0a309184775f67c14d14b5c365a:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:667d6c58d451dbf236ae37ab1de3b9f7:af733642ab69e156ba0c219d3bbc3c83:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:8dffa305e2bee837f279c2c0b082affb:::
```

rainbowcrack

Cracks hashes with rainbow tables.

```
RainbowCrack 1.5
Copyright 2003-2010 RainbowCrack Project. All rights reserved.
Official Website: http://project-rainbowcrack.com/

usage: rcrack rt_files [rt_files ...] -h hash
       rcrack rt_files [rt_files ...] -l hash_list_file
       rcrack rt_files [rt_files ...] -f pwdump_file
       rcrack rt_files [rt_files ...] -n pwdump_file
rt_files:               path to the rainbow table(s), wildchar(*, ?) supported
-h hash:                load single hash
-l hash_list_file:      load hashes from a file, each hash in a line
-f pwdump_file:         load lanmanager hashes from pwdump file
-n pwdump_file:         load ntlm hashes from pwdump file

hash algorithms implemented in alglib0.so:
    lm, plaintext_len limit: 0 - 7
    ntlm, plaintext_len limit: 0 - 15
    md5, plaintext_len limit: 0 - 15
    sha1, plaintext_len limit: 0 - 20
    mysqlsha1, plaintext_len limit: 0 - 20
    halflmchall, plaintext_len limit: 0 - 7
    ntlmchall, plaintext_len limit: 0 - 15
    oracle-SYSTEM, plaintext_len limit: 0 - 10
    md5-half, plaintext_len limit: 0 - 15

```

```
example: rcrack *.rt -h 5d41402abc4b2a76b9719d911017c592
         rcrack *.rt -l hash.txt
```

rcracki_mt

A modified version of rcrack which supports hybrid and indexed tables. In addition to that, it also adds multi-core support.

```
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:                use raw hash as input
-l hash_list_file:      use hash list file as input, each hash in a line
-f pwdump_file:         use pwdump file as input, handles lanmanager hash only
-c lst_file:            use .lst (cain format) file as input
-r [-s session_name]:   resume from previous session, optional session name
rainbow_table_pathname: pathname(s) of the rainbow table(s)

Extra options:    -t [nr] use this amount of threads/cores, default is 1
                  -o [output_file] write (temporary) results to this file
                  -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
```

rsmangler

RSMangler will take a wordlist and perform various manipulations on it similar to those done by John the Ripper, the main difference being that it will first take the input words and generate all permutations and the acronym of the words (in the order they appear in the file) before it applies the rest of the mangles.

```
rsmangler v 1.4 Robin Wood (robin@digininja.org) <www.randomstorm.com>

To pass the initial words in on standard in do:

cat wordlist.txt | ./rsmangler.rb --file - > new_wordlist.rb

All options are ON by default, these parameters turn them OFF

Usage: rsmangler.rb [OPTION]
  --help, -h: show help
  --file, -f: the input file, use - for STDIN
  --max, -x: maximum word length
  --min, -m: minimum word length
  --perms, -p: permutate all the words
  --double, -d: double each word
  --reverse, -r: reverser the word
  --leet, -t: l33t speak the word
  --full-leet, -T: all posibilities l33t
  --capital, -c: capitalise the word
  --upper, -u: uppercase the word
  --lower, -l: lowercase the word
  --swap, -s: swap the case of the word
  --ed, -e: add ed to the end of the word
  --ing, -i: add ing to the end of the word
  --punctuation: add common punctuation to the end of the word
  --years, -y: add all years from 1990 to current year to start and end
  --acronym, -a: create an acronym based on all the words entered in order and add to word list
  --common, -C: add the following words to start and end: admin, sys, pw, pwd
  --pna: add 01 - 09 to the end of the word
  --pnb: add 01 - 09 to the beginning of the word
```

```
--na: add 1 - 123 to the end of the word
--nb: add 1 - 123 to the beginning of the word
--force - don't check ooutput size
--space - add spaces between words
```

samdump2

Dumps Windows 2k/NT/XP/Vista password hashes

```
samdump2 1.1.1 by Objectif Securite
http://www.objectif-securite.ch
original author: ncuomo@studenti.unina.it

Usage:
samdump2 samhive keyfile
```

sipcrack

SIPcrack is a SIP login sniffer/cracker that contains 2 programs: sipdump to capture the digest authentication and sipcrack to bruteforce the hash using a wordlist or standard input.

sipcrack bruteforces the user's password with the dump file generated by sipdump. If a password is found, the sniffed and cracked login will be updated in the dump file.

```
SIPcrack 0.2  ( MaJoMu | www.codito.de )
----------------------------------------

Usage: sipcrack [OPTIONS] [ -s | -w <wordlist> ] <dump file>

      <dump file>   = file containing logins sniffed by SIPdump

      Options:
      -s            = use stdin for passwords
      -w wordlist   = file containing all passwords to try
      -p num        = print cracking process every n passwords (for -w)
                      (ATTENTION: slows down heavily)

* Invalid arguments
```

sucrack

Multithreaded Linux/UNIX tool for brute-force cracking of local user accounts via su.

```
sucrack [options] wordlist
```

truecrack

TrueCrack is a brute-force password cracker for TrueCrypt volumes. It works on Linux and it is optimized for Nvidia Cuda technology. It supports:

PBKDF2 (defined in PKCS5 v2.0) based on key derivation functions: Ripemd160, Sha512 and Whirlpool.

XTS block cipher mode for hard disk encryption based on encryption algorithms: AES, SERPENT, TWOFISH.

File-hosted (container) and Partition/device-hosted.

Hidden volumes and Backup headers.

TrueCrack is able to perform a brute-force attack based on:

Dictionary: read the passwords from a file of words.

Alphabet: generate all passwords of given length from given alphabet.

TrueCrack works on gpu and cpu

```
TrueCrack v3.0
Website: http://code.google.com/p/truecrack
Contact us: infotruecrack@gmail.com
Bruteforce password cracker for Truecrypt volume. Optimazed with Nvidia Cuda technology.
Based on TrueCrypt, freely available at http://www.truecrypt.org/
Copyright (c) 2011 by Luca Vaccaro.

Usage:
 truecrack -t <truecrypt_file> -k <ripemd160|sha512|whirlpool> -w <wordlist_file> [-b <parallel_block>]
 truecrack -t <truecrypt_file> -k <ripemd160|sha512|whirlpool> -c <charset> [-s <minlength>] -m <maxlength> [-b

Options:
 -h --help                          Display this information.
 -t --truecrypt <truecrypt_file>    Truecrypt volume file.
 -k --key <ripemd160 | sha512 | whirlpool>      Key derivation function (default ripemd160).
 -b --blocksize <parallel_blocks>       Number of parallel computations (board dependent).
 -w --wordlist <wordlist_file>      File of words, for Dictionary attack.
 -c --charset <alphabet>        Alphabet generator, for Alphabet attack.
 -s --startlength <minlength>       Starting length of passwords, for Alphabet attack (default 1).
 -m --maxlength <maxlength>     Maximum length of passwords, for Alphabet attack.
 -r --restore <number>          Restore the computation.
 -v --verbose                     Show computation messages.

Sample:
 Dictionary mode: truecrack --truecrypt ./volume --wordlist ./dictionary.txt
 Charset mode: truecrack --truecrypt ./volume --charset ./dictionary.txt --maxlength 10
```

Online Attacks

cewl

CeWL is a ruby app which spiders a given url to a specified depth, optionally following external links, and returns a list of words which can then be used for password crackers such as John the Ripper.

```
CeWL 5.0 Robin Wood (robin@digininja.org) (www.digininja.org)

Usage: cewl [OPTION] ... URL
        --help, -h: show help
        --keep, -k: keep the downloaded file
        --depth x, -d x: depth to spider to, default 2
        --min_word_length, -m: minimum word length, default 3
        --offsite, -o: let the spider visit other sites
        --write, -w file: write the output to the file
        --ua, -u user-agent: useragent to send
        --no-words, -n: don't output the wordlist
        --meta, -a include meta data
        --meta_file file: output file for meta data
        --email, -e include email addresses
        --email_file file: output file for email addresses
        --meta-temp-dir directory: the temporary directory used by exiftool when parsing files, default
        --count, -c: show the count for each word found

        Authentication
        --auth_type: digest or basic
        --auth_user: authentication username
        --auth_pass: authentication password

        Proxy Support
        --proxy_host: proxy host
        --proxy_port: proxy port, default 8080
        --proxy_username: username for proxy, if required
        --proxy_password: password for proxy, if required

        --verbose, -v: verbose

        URL: The site to spider.
```

findmyhash

Crack different types of hashes using free online services

```
/usr/bin/findmyhash 1.1.2 ( http://code.google.com/p/findmyhash/ )

Usage:
------

  python /usr/bin/findmyhash <algorithm> OPTIONS


Accepted algorithms are:
------------------------

  MD4       - RFC 1320
  MD5       - RFC 1321
  SHA1      - RFC 3174 (FIPS 180-3)
  SHA224    - RFC 3874 (FIPS 180-3)
  SHA256    - FIPS 180-3
  SHA384    - FIPS 180-3
  SHA512    - FIPS 180-3
  RMD160    - RFC 2857
  GOST      - RFC 5831
  WHIRLPOOL - ISO/IEC 10118-3:2004
  LM        - Microsoft Windows hash
  NTLM      - Microsoft Windows hash
  MYSQL     - MySQL 3, 4, 5 hash
  CISCO7    - Cisco IOS type 7 encrypted passwords
  JUNIPER   - Juniper Networks $9$ encrypted passwords
  LDAP_MD5  - MD5 Base64 encoded
  LDAP_SHA1 - SHA1 Base64 encoded

  NOTE: for LM / NTLM it is recommended to introduce both values with this format:
         python /usr/bin/findmyhash LM   -h 9a5760252b7455deaad3b435b51404ee:0d7f1f2bdeac6e574d6e18ca85fb58a7
         python /usr/bin/findmyhash NTLM -h 9a5760252b7455deaad3b435b51404ee:0d7f1f2bdeac6e574d6e18ca85fb58a7


Valid OPTIONS are:
------------------

  -h <hash_value>  If you only want to crack one hash, specify its value with this option.

  -f <file>        If you have several hashes, you can specify a file with one hash per line.
                   NOTE: All of them have to be the same type.

  -g               If your hash cannot be cracked, search it in Google and show all the results.
                   NOTE: This option ONLY works with -h (one hash input) option.


Examples:
---------

  -> Try to crack only one hash.
     python /usr/bin/findmyhash MD5 -h 098f6bcd4621d373cade4e832627b4f6

  -> Try to crack a JUNIPER encrypted password escaping special characters.
     python /usr/bin/findmyhash JUNIPER -h "\$9\$LbHX-wg4Z"

  -> If the hash cannot be cracked, it will be searched in Google.
     python /usr/bin/findmyhash LDAP_SHA1 -h "{SHA}cRDtpNCeBiql5KOQsKVyrA0sAiA=" -g

  -> Try to crack multiple hashes using a file (one hash per line).
     python /usr/bin/findmyhash MYSQL -f mysqlhashesfile.txt


Contact:
--------

[Web]           http://laxmarcaellugar.blogspot.com/
[Mail/Google+]  bloglaxmarcaellugar@gmail.com
[twitter]       @laXmarcaellugar
```

hydra

Hydra is a parallized login cracker which supports numerous protocols to attack. New modules are easy to add, beside that, it is flexible and very fast.

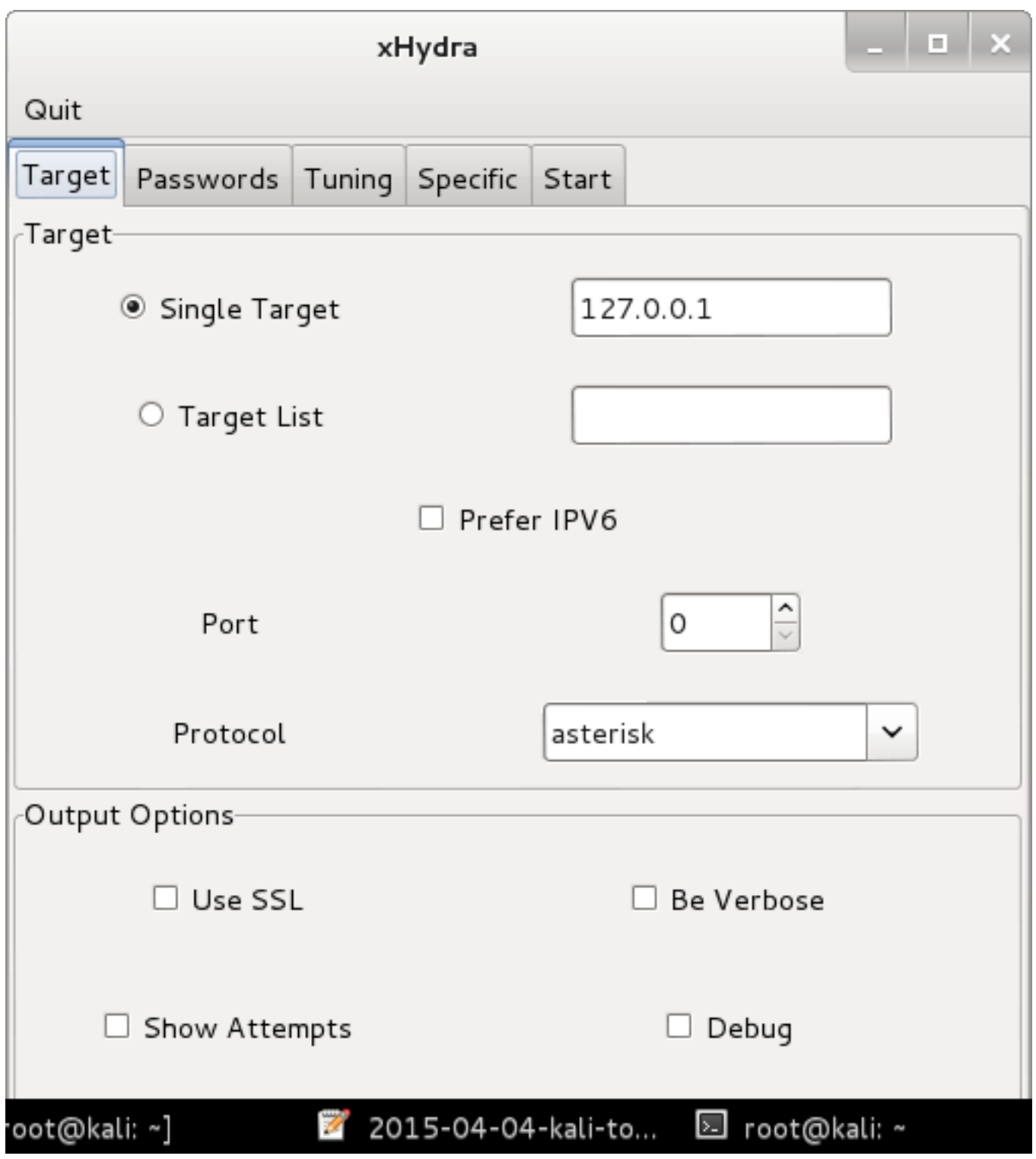
Currently this tool supports:

```
AFP, Cisco AAA, Cisco auth, Cisco enable, CVS, Firebird, FTP, FTPS,
HTTP-FORM-GET, HTTP-FORM-POST, HTTP-GET, HTTP-HEAD, HTTP-PROXY,
```

```
HTTP-PROXY-URLENUM, ICQ, IMAP, IRC, LDAP2, LDAP3, MS-SQL, MYSQL, NCP, NNTP,
Oracle, Oracle-Listener, Oracle-SID, PC-Anywhere, PCNFS, POP3, POSTGRES,
RDP, REXEC, RLOGIN, RSH, SAP/R3, SIP, SMB, SMTP, SMTP-Enum, SNMP,
SOCKS5, SSH(v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet,
VMware-Auth, VNC and XMPP.
For most protocols, SSL mode is available (e.g. https-get, ftp-ssl, etc.)


Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret service organizations, or for i

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T T

Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -t TASKS  run TASKS number of connects in parallel (per host, default: 16)
  -U        service module usage details
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get} http[s]-{get|post}-for

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at http://www.thc.org/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.

Example:  hydra -l user -P passlist.txt ftp://192.168.0.1
```

hydra-gtk

Hydra GUI



keimpx

keimpx is an open source tool, released under a modified version of Apache License 1.1.

It can be used to quickly check for valid credentials across a network over SMB. Credentials can be:

- Combination of user / plain-text password.
- Combination of user / NTLM hash.
- Combination of user / NTLM logon session token.

If any valid credentials has been discovered across the network after its attack phase, the user is asked to choose which host to connect to and which valid credentials to use, then he will be prompted with an interactive SMB shell where the user can:

- Spawn an interactive command prompt.
- Navigate through the remote SMB shares: list, upload, download files, create, remove files, etc.
- Deploy and undeploy his own service, for instance, a backdoor listening on a TCP port for incoming connections.
- List users details, domains and password policy.

```
keimpx 0.3-dev
    by Bernardo Damele A. G. <bernardo.damele@gmail.com>

Usage: keimpx.py [options]

Options:
  --version       show program's version number and exit
  -h, --help      show this help message and exit
  -v VERBOSE      Verbosity level: 0-2 (default: 0)
  -t TARGET       Target address
  -l LIST         File with list of targets
  -U USER         User
  -P PASSWORD     Password
  --nt=NTHASH     NT hash
  --lm=LMHASH     LM hash
  -c CREDSFILE    File with list of credentials
  -D DOMAIN       Domain
  -d DOMAINSFILE  File with list of domains
  -p PORT         SMB port: 139 or 445 (default: 445)
  -n NAME         Local hostname
  -T THREADS      Maximum simultaneous connections (default: 10)
  -b              Batch mode: do not ask to get an interactive SMB shell
  -x EXECUTELIST  Execute a list of commands against all hosts
```

medusa

Medusa is intended to be a speedy, massively parallel, modular, login brute-forcer. The goal is to support as many services which allow remote authentication as possible. The author considers following items to some of the key features of this application:

- Thread-based parallel testing. Brute-force testing can be performed against multiple hosts, users or passwords concurrently.
- Flexible user input. Target information (host/user/password) can be specified in a variety of ways. For example, each item can be either a single entry or a file containing multiple entries. Additionally, a combination file format allows the user to refine their target listing.
- Modular design. Each service module exists as an independent .mod file. This means that no modifications are necessary to the core application in order to extend the supported list of services for brute-forcing.

```
Medusa v2.0 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ALERT: Host information must be supplied.

Syntax: Medusa [-h host|-H file] [-u username|-U file] [-p password|-P file] [-C file] -M module [OPT]
  -h [TEXT]    : Target hostname or IP address
  -H [FILE]    : File containing target hostnames or IP addresses
  -u [TEXT]    : Username to test
  -U [FILE]    : File containing usernames to test
  -p [TEXT]    : Password to test
  -P [FILE]    : File containing passwords to test
  -C [FILE]    : File containing combo entries. See README for more information.
  -O [FILE]    : File to append log information to
  -e [n/s/ns]  : Additional password checks ([n] No Password, [s] Password = Username)
  -M [TEXT]    : Name of the module to execute (without the .mod extension)
  -m [TEXT]    : Parameter to pass to the module. This can be passed multiple times with a
                 different parameter each time and they will all be sent to the module (i.e.
```

```
                -m Param1 -m Param2, etc.)
  -d            : Dump all known modules
  -n [NUM]      : Use for non-default TCP port number
  -s            : Enable SSL
  -g [NUM]      : Give up after trying to connect for NUM seconds (default 3)
  -r [NUM]      : Sleep NUM seconds between retry attempts (default 3)
  -R [NUM]      : Attempt NUM retries before giving up. The total number of attempts will be NUM + 1.
  -t [NUM]      : Total number of logins to be tested concurrently
  -T [NUM]      : Total number of hosts to be tested concurrently
  -L            : Parallelize logins using one username per thread. The default is to process
                  the entire username before proceeding.
  -f            : Stop scanning host after first valid username/password found.
  -F            : Stop audit after first valid username/password found on any host.
  -b            : Suppress startup banner
  -q            : Display module's usage information
  -v [NUM]      : Verbose level [0 - 6 (more)]
  -w [NUM]      : Error debug level [0 - 10 (more)]
  -V            : Display version
  -Z [TEXT]     : Resume scan based on map of previous scan
```

ncrack

Ncrack is an open source tool for network authentication cracking. It was designed for high-speed parallel cracking using a dynamic engine that can adapt to different network situations. Ncrack can also be extensively fine-tuned for special cases, though the default parameters are generic enough to cover almost every situation. It is built on a modular architecture that allows for easy extension to support additional protocols. Ncrack is designed for companies and security professionals to audit large networks for default or weak passwords in a rapid and reliable way. It can also be used to conduct fairly sophisticated and intensive brute force attacks against individual services.

The output from Ncrack is a list of found credentials, if any, for each of the targets specified. Ncrack can also print an interactive status report of progress so far and possibly additional debugging information that can help track problems, if the user selected that option.

A typical Ncrack scan is shown in Example 1. The only Ncrack arguments used in this example are the two target IP addresses along with the the corresponding ports for each of them. The two example ports 21 and 22 are automatically resolved to the default services listening on them: ftp and ssh.

```
Example 1. A representative Ncrack scan

           $ ncrack 10.0.0.130:21 192.168.1.2:22

           Starting Ncrack 0.01ALPHA ( http://ncrack.org ) at 2009-07-24 23:05 EEST

           Discovered credentials for ftp on 10.0.0.130 21/tcp:
           10.0.0.130 21/tcp ftp: admin hello1
           Discovered credentials for ssh on 192.168.1.2 22/tcp:
           192.168.1.2 22/tcp ssh: guest 12345
           192.168.1.2 22/tcp ssh: admin money$

           Ncrack done: 2 services scanned in 156.03 seconds.

           Ncrack finished.
```

patator

A multi-purpose brute-forcer, with a modular design and a flexible usage.

```
Patator v0.5 (http://code.google.com/p/patator/)
Usage: patator.py module --help

Available modules:
  + ftp_login     : Brute-force FTP
  + ssh_login     : Brute-force SSH
  + telnet_login  : Brute-force Telnet
  + smtp_login    : Brute-force SMTP
  + smtp_vrfy     : Enumerate valid users using SMTP VRFY
  + smtp_rcpt     : Enumerate valid users using SMTP RCPT TO
  + finger_lookup : Enumerate valid users using Finger
  + http_fuzz     : Brute-force HTTP
  + pop_login     : Brute-force POP3
  + pop_passd     : Brute-force poppassd (http://netwinsite.com/poppassd/)
  + imap_login    : Brute-force IMAP4
  + ldap_login    : Brute-force LDAP
  + smb_login     : Brute-force SMB
  + smb_lookupsid : Brute-force SMB SID-lookup
  + vmauthd_login : Brute-force VMware Authentication Daemon
```

```
+ mssql_login   : Brute-force MSSQL
+ oracle_login  : Brute-force Oracle
+ mysql_login   : Brute-force MySQL
+ mysql_query   : Brute-force MySQL queries
+ pgsql_login   : Brute-force PostgreSQL
+ vnc_login     : Brute-force VNC
+ dns_forward   : Forward lookup names
+ dns_reverse   : Reverse lookup subnets
+ snmp_login    : Brute-force SNMP v1/2/3
+ unzip_pass    : Brute-force the password of encrypted ZIP files
+ keystore_pass : Brute-force the password of Java keystore files
+ tcp_fuzz      : Fuzz TCP services
+ dummy_test    : Testing module
```

phrasendrescher

phrasen|drescher (pld) is a modular and multi processing pass phrase cracking tool. It comes with a number of plugins but a simple plugin API allows an easy development of new plugins. The main features of pld are:

- Modular with the use of plugins
- Multi processing
- Dictionary attack with or without permutations (uppercase, lowercase, l33t, etc.)
- Incremental brute force attack with custom character maps
- Runs on FreeBSD, NetBSD, OpenBSD, MacOS and Linux

```
phrasen|drescher 1.2.2 - the passphrase cracker
Copyright (C) 2008 Nico Leidecker; http://www.leidecker.info

Usage: pd plugin [options]

 Available plugins:
   enc-file  pkey  ssh  mssql  http-raw

 General Options:
   h           : print this message
   v           : verbose mode
   i from[:to] : incremental mode beginning with word length `from'
                 and going to `to'
   d file      : run dictionary based with words from `file'
   w number    : number of worker threads (default is one)
   r rules     : specify rewriting rules for the dictionary mode:
                   A = all characters upper case
                   F = first character upper case
                   L = last character upper case
                   W = first letter of each word to upper case
                   a = all characters lower case
                   f = first character lower case
                   l = last character lower case
                   w = first letter of each word to lower case
                   D = prepend digit
                   d = append digit
                   e = 1337 characters
                   x = all rules

 Environment Variables:
   PD_PLUGINS : the directory containing plugins
                (current is /usr/lib/phrasendrescher)
   PD_CHARMAP : the characters for the incremental mode are
                taken from a character list. A customized list
                can be specified in the environment variable
```

thc-pptp-bruter

Brute force program against pptp vpn endpoints (tcp port 1723). Fully standalone. Supports latest MSChapV2 authentication. Tested against Windows and Cisco gateways. Exploits a weakness in Microsoft's anti-brute force implementation which makes it possible to try 300 passwords the second.

```
Target IP missing.
thc-pptp-bruter [options] <remote host IP>
  -v        Verbose output / Debug output
```

```
  -W        Disable windows hack [default: enabled]
  -u <user> User [default: administrator]
  -w <file> Wordlist file [default: stdin]
  -p <n>    PPTP port [default: 1723]
  -n <n>    Number of parallel tries [default: 5]
  -l <n>    Limit to n passwords / sec [default: 100]

Windows-Hack reuses the LCP connection with the same caller-id. This
gets around MS's anti-brute forcing protection. It's enabled by default.
```

Passing the Hash

Pass the Hash Toolkit

This is a collection of scripts for pass-the-hash scenarios

pth-curl

```
root@kali:~# pth-curl --help
Usage: curl [options...] <url>
Options: (H) means HTTP/HTTPS only, (F) means FTP only
     --anyauth       Pick "any" authentication method (H)
 -a, --append        Append to target file when uploading (F/SFTP)
     --basic         Use HTTP Basic Authentication (H)
     --cacert FILE   CA certificate to verify peer against (SSL)
     --capath DIR    CA directory to verify peer against (SSL)
 -E, --cert CERT[:PASSWD] Client certificate file and password (SSL)
     --cert-type TYPE Certificate file type (DER/PEM/ENG) (SSL)
     --ciphers LIST  SSL ciphers to use (SSL)
     --compressed    Request compressed response (using deflate or gzip)
 -K, --config FILE   Specify which config file to read
     --connect-timeout SECONDS  Maximum time allowed for connection
 -C, --continue-at OFFSET  Resumed transfer offset
 -b, --cookie STRING/FILE  String or file to read cookies from (H)
 -c, --cookie-jar FILE  Write cookies to this file after operation (H)
     --create-dirs   Create necessary local directory hierarchy
     --crlf          Convert LF to CRLF in upload
     --crlfile FILE  Get a CRL list in PEM format from the given file
 -d, --data DATA     HTTP POST data (H)
     --data-ascii DATA  HTTP POST ASCII data (H)
     --data-binary DATA  HTTP POST binary data (H)
     --data-urlencode DATA  HTTP POST data url encoded (H)
     --delegation STRING GSS-API delegation permission
     --digest        Use HTTP Digest Authentication (H)
     --disable-eprt  Inhibit using EPRT or LPRT (F)
     --disable-epsv  Inhibit using EPSV (F)
 -D, --dump-header FILE  Write the headers to this file
     --egd-file FILE  EGD socket path for random data (SSL)
     --engine ENGINGE  Crypto engine (SSL). "--engine list" for list
 -f, --fail          Fail silently (no output at all) on HTTP errors (H)
 -F, --form CONTENT  Specify HTTP multipart POST data (H)
     --form-string STRING  Specify HTTP multipart POST data (H)
     --ftp-account DATA  Account data string (F)
     --ftp-alternative-to-user COMMAND  String to replace "USER [name]" (F)
     --ftp-create-dirs  Create the remote dirs if not present (F)
     --ftp-method [MULTICWD/NOCWD/SINGLECWD] Control CWD usage (F)
     --ftp-pasv      Use PASV/EPSV instead of PORT (F)
 -P, --ftp-port ADR  Use PORT with given address instead of PASV (F)
     --ftp-skip-pasv-ip Skip the IP address for PASV (F)
     --ftp-pret      Send PRET before PASV (for drftpd) (F)
     --ftp-ssl-ccc   Send CCC after authenticating (F)
     --ftp-ssl-ccc-mode ACTIVE/PASSIVE  Set CCC mode (F)
     --ftp-ssl-control Require SSL/TLS for ftp login, clear for transfer (F)
 -G, --get           Send the -d data with a HTTP GET (H)
 -g, --globoff       Disable URL sequences and ranges using {} and []
 -H, --header LINE   Custom header to pass to server (H)
 -I, --head          Show document info only
 -h, --help          This help text
     --hostpubmd5 MD5  Hex encoded MD5 string of the host public key. (SSH)
 -0, --http1.0       Use HTTP 1.0 (H)
     --ignore-content-length  Ignore the HTTP Content-Length header
 -i, --include       Include protocol headers in the output (H/F)
 -k, --insecure      Allow connections to SSL sites without certs (H)
     --interface INTERFACE  Specify network interface/address to use
 -4, --ipv4          Resolve name to IPv4 address
 -6, --ipv6          Resolve name to IPv6 address
 -j, --junk-session-cookies Ignore session cookies read from file (H)
```

```
     --keepalive-time SECONDS  Interval between keepalive probes
     --key KEY       Private key file name (SSL/SSH)
     --key-type TYPE Private key file type (DER/PEM/ENG) (SSL)
     --krb LEVEL     Enable Kerberos with specified security level (F)
     --libcurl FILE  Dump libcurl equivalent code of this command line
     --limit-rate RATE  Limit transfer speed to this rate
  -l, --list-only     List only names of an FTP directory (F)
     --local-port RANGE  Force use of these local port numbers
  -L, --location      Follow redirects (H)
     --location-trusted like --location and send auth to other hosts (H)
  -M, --manual        Display the full manual
     --mail-from FROM  Mail from this address
     --mail-rcpt TO  Mail to this receiver(s)
     --mail-auth AUTH  Originator address of the original email
     --max-filesize BYTES  Maximum file size to download (H/F)
     --max-redirs NUM  Maximum number of redirects allowed (H)
  -m, --max-time SECONDS  Maximum time allowed for the transfer
     --negotiate     Use HTTP Negotiate Authentication (H)
  -n, --netrc         Must read .netrc for user name and password
     --netrc-optional Use either .netrc or URL; overrides -n
     --netrc-file FILE  Set up the netrc filename to use
  -N, --no-buffer     Disable buffering of the output stream
     --no-keepalive  Disable keepalive use on the connection
     --no-sessionid  Disable SSL session-ID reusing (SSL)
     --noproxy       List of hosts which do not use proxy
     --ntlm          Use HTTP NTLM authentication (H)
  -o, --output FILE   Write output to <file> instead of stdout
     --pass PASS     Pass phrase for the private key (SSL/SSH)
     --post301       Do not switch to GET after following a 301 redirect (H)
     --post302       Do not switch to GET after following a 302 redirect (H)
     --post303       Do not switch to GET after following a 303 redirect (H)
  -#, --progress-bar  Display transfer progress as a progress bar
     --proto PROTOCOLS  Enable/disable specified protocols
     --proto-redir PROTOCOLS  Enable/disable specified protocols on redirect
  -x, --proxy [PROTOCOL://]HOST[:PORT] Use proxy on given port
     --proxy-anyauth Pick "any" proxy authentication method (H)
     --proxy-basic   Use Basic authentication on the proxy (H)
     --proxy-digest  Use Digest authentication on the proxy (H)
     --proxy-negotiate Use Negotiate authentication on the proxy (H)
     --proxy-ntlm    Use NTLM authentication on the proxy (H)
  -U, --proxy-user USER[:PASSWORD]  Proxy user and password
     --proxy1.0 HOST[:PORT]  Use HTTP/1.0 proxy on given port
  -p, --proxytunnel   Operate through a HTTP proxy tunnel (using CONNECT)
     --pubkey KEY    Public key file name (SSH)
  -Q, --quote CMD     Send command(s) to server before transfer (F/SFTP)
     --random-file FILE  File for reading random data from (SSL)
  -r, --range RANGE   Retrieve only the bytes within a range
     --raw           Do HTTP "raw", without any transfer decoding (H)
  -e, --referer       Referer URL (H)
  -J, --remote-header-name Use the header-provided filename (H)
  -O, --remote-name   Write output to a file named as the remote file
     --remote-name-all Use the remote file name for all URLs
  -R, --remote-time   Set the remote file's time on the local output
  -X, --request COMMAND  Specify request command to use
     --resolve HOST:PORT:ADDRESS  Force resolve of HOST:PORT to ADDRESS
     --retry NUM   Retry request NUM times if transient problems occur
     --retry-delay SECONDS When retrying, wait this many seconds between each
     --retry-max-time SECONDS  Retry only within this period
  -S, --show-error    Show error. With -s, make curl show errors when they occur
  -s, --silent        Silent mode. Don't output anything
     --socks4 HOST[:PORT]  SOCKS4 proxy on given host + port
     --socks4a HOST[:PORT]  SOCKS4a proxy on given host + port
     --socks5 HOST[:PORT]  SOCKS5 proxy on given host + port
     --socks5-hostname HOST[:PORT] SOCKS5 proxy, pass host name to proxy
     --socks5-gssapi-service NAME  SOCKS5 proxy service name for gssapi
     --socks5-gssapi-nec  Compatibility with NEC SOCKS5 server
  -Y, --speed-limit RATE  Stop transfers below speed-limit for 'speed-time' secs
  -y, --speed-time SECONDS  Time for trig speed-limit abort. Defaults to 30
     --ssl           Try SSL/TLS (FTP, IMAP, POP3, SMTP)
     --ssl-reqd      Require SSL/TLS (FTP, IMAP, POP3, SMTP)
  -2, --sslv2         Use SSLv2 (SSL)
  -3, --sslv3         Use SSLv3 (SSL)
     --ssl-allow-beast Allow security flaw to improve interop (SSL)
     --stderr FILE   Where to redirect stderr. - means stdout
     --tcp-nodelay   Use the TCP_NODELAY option
  -t, --telnet-option OPT=VAL  Set telnet option
     --tftp-blksize VALUE  Set TFTP BLKSIZE option (must be >512)
  -z, --time-cond TIME  Transfer based on a time condition
  -1, --tlsv1         Use TLSv1 (SSL)
     --trace FILE    Write a debug trace to the given file
     --trace-ascii FILE  Like --trace but without the hex output
```

```
     --trace-time    Add time stamps to trace/verbose output
     --tr-encoding   Request compressed transfer encoding (H)
-T, --upload-file FILE  Transfer FILE to destination
     --url URL       URL to work with
-B, --use-ascii     Use ASCII/text transfer
-u, --user USER[:PASSWORD]  Server user and password
     --tlsuser USER  TLS username
     --tlspassword STRING TLS password
     --tlsauthtype STRING  TLS authentication type (default SRP)
-A, --user-agent STRING  User-Agent to send to server (H)
-v, --verbose       Make the operation more talkative
-V, --version       Show version number and quit
-w, --write-out FORMAT  What to output after completion
     --xattr        Store metadata in extended file attributes
-q                  If used as the first parameter disables .curlrc
```

pth-net

```
Usage:
net rpc             Run functions using RPC transport
net rap             Run functions using RAP transport
net ads             Run functions using ADS transport
net file            Functions on remote opened files
net share           Functions on shares
net session         Manage sessions
net server          List servers in workgroup
net domain          List domains/workgroups on network
net printq          Modify printer queue
net user            Manage users
net group           Manage groups
net groupmap        Manage group mappings
net sam             Functions on the SAM database
net validate        Validate username and password
net groupmember     Modify group memberships
net admin           Execute remote command on a remote OS/2 server
net service         List/modify running services
net password        Change user password on target server
net changetrustpw   Change the trust password
net changesecretpw  Change the secret password
net setauthuser     Set the winbind auth user
net getauthuser     Get the winbind auth user settings
net time            Show/set time
net lookup          Look up host names/IP addresses
net g_lock          Manipulate the global lock table
net join            Join a domain/AD
net dom             Join/unjoin (remote) machines to/from a domain/AD
net cache           Operate on the cache tdb file
net getlocalsid     Get the SID for the local domain
net setlocalsid     Set the SID for the local domain
net setdomainsid    Set domain SID on member servers
net getdomainsid    Get domain SID on member servers
net maxrid          Display the maximum RID currently used
net idmap           IDmap functions
net status          Display server status
net usershare       Manage user-modifiable shares
net usersidlist     Display list of all users with SID
net conf            Manage Samba registry based configuration
net registry        Manage the Samba registry
net eventlog        Process Win32 *.evt eventlog files
net printing        Process tdb printer files
net serverid        Manage the serverid tdb
net help            Print usage information
Valid targets: choose one (none defaults to localhost)
  -S or --server=<server>       server name
  -I or --ipaddress=<ipaddr>    address of target server
  -w or --workgroup=<wg>        target workgroup or domain

Valid miscellaneous options are:
  -p or --port=<port>       connection port on target
  -W or --myworkgroup=<wg>  client workgroup
  -d or --debuglevel=<level>    debug level (0-10)
  -n or --myname=<name>     client name
  -U or --user=<name>       user name
  -s or --configfile=<path> pathname of smb.conf file
  -l or --long            Display full information
  -V or --version         Print samba version information
  -P or --machine-pass        Authenticate as machine account
  -e or --encrypt         Encrypt SMB transport (UNIX extended servers only)
  -k or --kerberos        Use kerberos (active directory) authentication
```

pth-openchangeclient

```
Usage: openchangeclient [OPTION...]
  -f, --database=STRING          set the profile database path
      --pf                       access public folders instead of mailbox
  -p, --profile=STRING           set the profile name
  -P, --password=STRING          set the profile password
      --username=STRING          set the username of the mailbox to use
  -S, --sendmail                 send a mail
      --sendappointment          send an appointment
      --sendcontact              send a contact
      --sendtask                 send a task
      --sendnote                 send a note
  -F, --fetchmail                fetch user INBOX mails
      --fetchsummary             fetch message summaries only
  -G, --storemail=STRING         retrieve a mail on the filesystem
  -i, --fetch-items=STRING       fetch specified user INBOX items
      --freebusy=STRING          display free / busy information for the specified user
      --force                    force openchangeclient behavior in some circumstances
      --delete=STRING            delete a message given its unique ID
  -u, --update=STRING            update the specified item
  -m, --mailbox                  list mailbox folder summary
  -D, --deletemail               delete a mail from user INBOX
  -A, --attachments=STRING       send a list of attachments
  -I, --html-inline=STRING       send PR_HTML content
  -W, --html-file=STRING         use HTML file as content
  -t, --to=STRING                set the To recipients
  -c, --cc=STRING                set the Cc recipients
  -b, --bcc=STRING               set the Bcc recipients
  -s, --subject=STRING           set the mail subject
  -B, --body=STRING              set the mail body
      --location=STRING          set the item location
      --label=STRING             set the event label
      --dtstart=STRING           set the event start date
      --dtend=STRING             set the event end date
      --busystatus=STRING        set the item busy status
      --taskstatus=STRING        set the task status
      --importance=STRING        Set the item importance
      --email=STRING             set the email address
      --fullname=STRING          set the full name
      --cardname=STRING          set a contact card name
      --color=STRING             set the note color
      --notifications            monitor INBOX newmail notifications
      --folder=STRING            set the folder to use instead of inbox
      --mkdir                    create a folder
      --rmdir                    delete a folder
      --userlist                 list Address Book entries
      --folder-name=STRING       set the folder name
      --folder-comment=STRING    set the folder comment
  -d, --debuglevel=STRING        set Debug Level
      --dump-data                dump the hex data
      --private                  set the private flag on messages
      --ocpf-file=STRING         set OCPF file
      --ocpf-dump=STRING         dump message into OCPF file
      --ocpf-syntax              check OCPF files syntax
      --ocpf-sender              send message using OCPF files contents

Help options:
  -?, --help                     Show this help message
      --usage                    Display brief usage message

Common openchange options:
  -V, --version                  Print version
```

pth-rpcclient

```
Usage: rpcclient [OPTION...]
  -c, --command=COMMANDS                Execute semicolon separated cmds
  -I, --dest-ip=IP                      Specify destination IP address
  -p, --port=PORT                       Specify port number

Help options:
  -?, --help                            Show this help message
      --usage                           Display brief usage message

```

```
Common samba options:
  -d, --debuglevel=DEBUGLEVEL            Set debug level
  -s, --configfile=CONFIGFILE            Use alternate configuration file
  -l, --log-basename=LOGFILEBASE         Base name for log files
  -V, --version                          Print version
      --option=name=value                Set smb.conf option from command line

Connection options:
  -O, --socket-options=SOCKETOPTIONS     socket options to use
  -n, --netbiosname=NETBIOSNAME          Primary netbios name
  -W, --workgroup=WORKGROUP              Set the workgroup name
  -i, --scope=SCOPE                      Use this Netbios scope

Authentication options:
  -U, --user=USERNAME                    Set the network username
  -N, --no-pass                          Don't ask for a password
  -k, --kerberos                         Use kerberos (active directory)
                                         authentication
  -A, --authentication-file=FILE         Get the credentials from a file
  -S, --signing=on|off|required          Set the client signing state
  -P, --machine-pass                     Use stored machine account password
  -e, --encrypt                          Encrypt SMB transport (UNIX extended
                                         servers only)
  -C, --use-ccache                       Use the winbind ccache for
                                         authentication
      --pw-nt-hash                       The supplied password is the NT hash
```

pth-smbclient

```
Usage: smbclient [-?EgBVNkPeC] [-?|--help] [--usage]
         [-R|--name-resolve=NAME-RESOLVE-ORDER] [-M|--message=HOST]
         [-I|--ip-address=IP] [-E|--stderr] [-L|--list=HOST]
         [-m|--max-protocol=LEVEL] [-T|--tar=<c|x>IXFqgbNan]
         [-D|--directory=DIR] [-c|--command=STRING] [-b|--send-buffer=BYTES]
         [-p|--port=PORT] [-g|--grepable] [-B|--browse]
         [-d|--debuglevel=DEBUGLEVEL] [-s|--configfile=CONFIGFILE]
         [-l|--log-basename=LOGFILEBASE] [-V|--version] [--option=name=value]
         [-O|--socket-options=SOCKETOPTIONS] [-n|--netbiosname=NETBIOSNAME]
         [-W|--workgroup=WORKGROUP] [-i|--scope=SCOPE] [-U|--user=USERNAME]
         [-N|--no-pass] [-k|--kerberos] [-A|--authentication-file=FILE]
         [-S|--signing=on|off|required] [-P|--machine-pass] [-e|--encrypt]
         [-C|--use-ccache] [--pw-nt-hash] service <password>
```

pth-smbget

```
Usage: smbget [OPTION...]
  -a, --guest                 Work as user guest
  -e, --encrypt               Encrypt SMB transport (UNIX extended servers
                              only)
  -r, --resume                Automatically resume aborted files
  -U, --update                Download only when remote file is newer than
                              local file or local file is missing
  -R, --recursive             Recursively download files
  -u, --username=STRING       Username to use
  -p, --password=STRING       Password to use
  -w, --workgroup=STRING      Workgroup to use (optional)
  -n, --nonprompt             Don't ask anything (non-interactive)
  -d, --debuglevel=INT        Debuglevel to use
  -o, --outputfile=STRING     Write downloaded data to specified file
  -O, --stdout                Write data to stdout
  -D, --dots                  Show dots as progress indication
  -q, --quiet                 Be quiet
  -v, --verbose               Be verbose
  -P, --keep-permissions      Keep permissions
  -b, --blocksize=INT         Change number of bytes in a block
  -f, --rcfile=STRING         Use specified rc file

Help options:
  -?, --help                  Show this help message
      --usage                 Display brief usage message
```

pth-sqsh

```
Use: sqsh [-a count] [-A packet_size] [-b] [-B] [-c [cmdend]] [-C sql]
```

```
          [-d severity] [-D database] [-e] [-E editor] [-f severity]
          [-G TDS version] [-h] [-H hostname] [-i filename] [-I interfaces]
          [-J charset] [-k keywords] [-K keytab] [-l level|flags]
          [-L var=value] [-m style] [-n {on|off}] [-N appname] [-o filename]
          [-p] [-P [password]] [-Q query_timeout] [-r [sqshrc]]
          [-R principal] [-s colsep] [-S server] [-t [filter]]
          [-T login_timeout] [-U username] [-v] [-V [bcdimoqru]] [-w width]
          [-X] [-y directory] [-z language] [-Z [secmech]]

-a  Max. # of errors before abort       -m  Set display mode
-A  Adjust TDS packet size              -n  Set chained transaction mode
-b  Suppress banner message on startup  -N  Set Application Name (sqsh)
-B  Turn off file buffering on startup  -o  Direct all output to file
-c  Alias for the 'go' command          -p  Display performance stats
-C  Send sql statement to server        -P  Sybase password (NULL)
-d  Min. severity level to display      -Q  Query timeout period in seconds
-D  Change database context on startup  -r  Specify name of .sqshrc
-e  Echo batch prior to executing       -R  Network security server principal
-E  Replace default editor (vi)         -s  Alternate column separator (\t)
-f  Min. severity level for failure     -S  Name of Sybase server ($DSQUERY)
-G  TDS version to use                  -t  Filter batches through program
-h  Disable headers and footers         -T  Login timeout period in seconds
-H  Set the client hostname             -U  Name of Sybase user
-i  Read input from file                -v  Display current version and exit
-I  Alternate interfaces file           -V  Request network security services
-J  Client character set                -w  Adjust result display width
-k  Specify alternate keywords file     -X  Enable client password encryption
-K  Network security keytab file (DCE)  -y  Override value of $SYBASE
-l  Set debugging level                 -z  Alternate display language
-L  Set the value of a given variable   -Z  Network security mechanism
```

pth-winexe

```
winexe version 1.1
This program may be freely redistributed under the terms of the GNU GPLv3
Usage: winexe [OPTION]... //HOST COMMAND
Options:
  -?, --help                                  Display help message
  -U, --user=[DOMAIN/]USERNAME[%PASSWORD]     Set the network username
  -A, --authentication-file=FILE              Get the credentials from a file
  -k, --kerberos=STRING                       Use Kerberos, -k [yes|no]
  -d, --debuglevel=DEBUGLEVEL                 Set debug level
      --uninstall                             Uninstall winexe service after remote execution
      --reinstall                             Reinstall winexe service before remote execution
      --system                                Use SYSTEM account
      --profile                               Load user profile
      --convert                               Try to convert characters between local and remote code-pages
      --runas=[DOMAIN\]USERNAME%PASSWORD      Run as user (BEWARE: password is sent in cleartext over net)
      --runas-file=FILE                       Run as user options defined in a file
      --interactive=0|1                       Desktop interaction: 0 - disallow, 1 - allow. If you allow use al
                                              requirement). Vista do not support this option.
      --ostype=0|1|2                          OS type: 0 - 32-bit, 1 - 64-bit, 2 - winexe will decide. Determin
                                              of service will be installed.
```

pth-wmic

```
Usage: [-?|--help] [--usage] [-d|--debuglevel DEBUGLEVEL] [--debug-stderr]
         [-s|--configfile CONFIGFILE] [--option=name=value]
         [-l|--log-basename LOGFILEBASE] [--leak-report] [--leak-report-full]
         [-R|--name-resolve NAME-RESOLVE-ORDER]
         [-O|--socket-options SOCKETOPTIONS] [-n|--netbiosname NETBIOSNAME]
         [-W|--workgroup WORKGROUP] [--realm=REALM] [-i|--scope SCOPE]
         [-m|--maxprotocol MAXPROTOCOL] [-U|--user [DOMAIN\]USERNAME[%PASSWORD]]
         [-N|--no-pass] [--password=STRING] [-A|--authentication-file FILE]
         [-S|--signing on|off|required] [-P|--machine-pass]
         [--simple-bind-dn=STRING] [-k|--kerberos STRING]
         [--use-security-mechanisms=STRING] [-V|--version] [--namespace=STRING]
         [--delimiter=STRING]
        //host query

Example: wmic -U [domain/]adminuser%password //host "select * from Win32_ComputerSystem"
```

pth-wmis

```
Usage: [-?|--help] [--usage] [-d|--debuglevel DEBUGLEVEL] [--debug-stderr]
         [-s|--configfile CONFIGFILE] [--option=name=value]
         [-l|--log-basename LOGFILEBASE] [--leak-report] [--leak-report-full]
         [-R|--name-resolve NAME-RESOLVE-ORDER]
         [-O|--socket-options SOCKETOPTIONS] [-n|--netbiosname NETBIOSNAME]
         [-W|--workgroup WORKGROUP] [--realm=REALM] [-i|--scope SCOPE]
         [-m|--maxprotocol MAXPROTOCOL] [-U|--user [DOMAIN\]USERNAME[%PASSWORD]]
         [-N|--no-pass] [--password=STRING] [-A|--authentication-file FILE]
         [-S|--signing on|off|required] [-P|--machine-pass]
         [--simple-bind-dn=STRING] [-k|--kerberos STRING]
         [--use-security-mechanisms=STRING] [-V|--version]
        //host

Example: wmis -U [domain/]adminuser%password //host cmd.exe /c dir c:\ > c:\windows\temp\output.txt
```

People are beginning to notice you. Try dressing before you leave the house.

Posted by chousensha Apr 4th, 2015 kali, penetration testing, tools

« Pentest lab - Flick Kali tools catalog - Wireless Attacks »

Comments

whoami

switch (interests){
case INFORMATION SECURITY:
Mostly offensive security, but trying to be well-rounded in everything;
case PYTHON:
Mainly security and sysadmin related scripting;
case LINUX:
Greetings from /dev/null;
case JAPANESE:
Language, anime, samurai;
case MARTIAL ARTS:
If it's fighting I like it;
case MILITARY SCIENCE:
Ancient, medieval, modern;
default: GAMING;}

Recent Posts

- There be Tr0lls - Part 3
- No Mercy
- Pond. Analoguepond
- Derpnstink
- Donkey Docker

GitHub Repos

- cyber-support-base

  Collection of bookmarked tools for security, red teaming, blue teaming, pentesting and other

- automation

  Various automation tasks

- network_scripts

  Collection of miscellaneous scripts

- linux_privcheck

Check privileges, settings and other information on Linux systems and suggest exploits based on kernel versions

- kloggy

@chousensha on GitHub

Latest Tweets

Blogroll

g0tmi1k

Red Team Journal

Corelan Team

Mad Irish

redteams.net

MattAndreko.com

Portswigger Web Security

Cobalt Strike blog

HighOn.Coffee

Penetration Testing Lab