GitHub - CiscoCXSecurity/creddump7

By timb-machine

Archived: 2026-04-05 19:40:16 UTC

Information

This repo is for my modifications to the original 'creddump' program available at:

https://code.google.com/p/creddump/

I did not write the original program.

I have combined many patches and fixes I have seen from different forums and user suggestions, as well as modified the usage to make it a little more clear.

I followed patches and fixes from the following links:

- https://code.google.com/p/creddump/issues/detail?id=4
- https://code.google.com/p/volatility/issues/detail?id=92

Enjoy! Ronnie Flathers (@ropnop)

Usage

Mount a Windows 7/Vista partition:

```
 # mkdir /mnt/win
# ntfs-3g /dev/sda1 /mnt/win
```

Run cachedump.py on the SYSTEM and SECURITY hives to extract cached domain creds:

```
 # ./cachedump.py
usage: ./cachedump.py <system hive> <security hive> <Vista/7>

Example (Windows Vista/7):
./cachedump.py /path/to/System32/config/SYSTEM /path/to/System32/config/SECURITY true

Example (Windows XP):
./cachedump.py /path/to/System32/SYSTEM /path/to/System32/config/SECURITY false

# ./cachedump.py /mnt/win/Windows/System32/config/SYSTEM /mnt/win/Windows/System32/config/SECURITY true |tee has
```

```
nharpsis:6b29dfa157face3f3d8db489aec5cc12:acme:acme.local
god:25bd785b8ff1b7fa3a9b9e069a5e7de7:acme:acme.local
```

If you want to crack the hashes and have a good wordlist, John can be used. The hashes are in the 'mscash2' format:

```
 # john --format=mscash2 --wordlist=/usr/share/wordlists/rockyou.txt hashes
Loaded 2 password hashes with 2 different salts (M$ Cache Hash 2 (DCC2) PBKDF2-HMAC-SHA-1 [128/128 SSE2 intrins
g0d              (god)
Welcome1!        (nharpsis)
```

We now have the passwords for two domain users. Note: these passwords are really simple and I knew they were in the wordlist I used. Normally if you want to actually bruteforce the passwords, I wouldn't recommend John. Pull the hashes and use a GPU powered cracking box with oclHashcat.

Below is the original README file

```
 OVERVIEW

creddump is a python tool to extract various credentials and secrets from
Windows registry hives. It currently extracts:
* LM and NT hashes (SYSKEY protected)
* Cached domain passwords
* LSA secrets

It essentially performs all the functions that bkhive/samdump2,
cachedump, and lsadump2 do, but in a platform-independent way.

It is also the first tool that does all of these things in an offline
way (actually, Cain & Abel does, but is not open source and is only
available on Windows).

REQUIREMENTS

alldump has only been tested on python 2.5. It should work on 2.4 as
well, but will likely need modification before it will work on 2.3 or
below.

python-crypto is required for its MD5/DES/RC4 support. To obtain it,
see: http://www.amk.ca/python/code/crypto

For lsadump: system and SECURITY hives
For cachedump: system and SECURITY hives
For pwdump: system and SAM hives
```

```
USAGE

Dump cached domain hashes:
  usage: ./cachedump.py <system hive> <security hive>

Dump LSA secrets:
  usage: ./lsadump.py <system hive> <security hive>

Dump local password hashes:
  usage: ./pwdump.py <system hive> <SAM hive>

FEATURES

* Platform independent operation. The only inputs are the hive files
  from the system--we don't rely on any Windows functionality at all.
* Open-source and (hopefully!) readble implementations of Windows
  obfuscation algorithms used to protect LSA secrets, cached domain
  passwords, and
* A reasonably forgiving registry file parser in pure Python. Look
  through framework/types.py and framework/win32/rawreg.py to see how it
  works.
* The first complete open-source implementation of advapi32's
  SystemFunction005. The version in the Wine source code does not
  appear to allow for keys longer than 7 bytes, while the Windows
  version (and this version) does. See decrypt_secret() in
  framework/win32/lsasecrets.py

AUTHOR

creddump is written by Brendan Dolan-Gavitt (bdolangavitt@wesleyan.edu).
For more information on Syskey, LSA secrets, cached domain credentials,
and lots of information on volatile memory forensics and reverse
engineering, check out:

http://moyix.blogspot.com/

CREDITS
* AAron Walters. Much of the data type parsing code is taken from
  Volatility, an excellent memory analysis framework written in Python.
  He's also a really nice guy, and has helped me out a lot in my
  research.

  https://www.volatilesystems.com/default/volatility

* Massimiliano Montoro (mao), for reversing the mechanism Windows uses
  to derive the LSA key so that it can be computed directly from the
  hive files, as decribed in this post:
```

```
  http://oxid.netsons.org/phpBB2/viewtopic.php?t=149
  http://www.oxid.it/

* Jeremy Allison, for the details of the obfuscation applied to password
  hashes in the SAM, as implemented in the original pwdump.

  http://us4.samba.org/samba/ftp/pwdump/

* Nicola Cuomo, for his excellent description of the syskey mechanism
  and how it is used to encrypt the SAM in Windows 2000 and above.

  http://www.studenti.unina.it/~ncuomo/syskey/

* Eyas[at]xfocus.org, for x_dialupass2.cpp, which demonstrates how to
  read LSA secrets directly from the registry, given the LSA key.

  http://www.xfocus.net/articles/200411/749.html

  [Note: the above is in Chinese, but quite comprehensible if you use
   Google Translate and can read C ;)]

* Nicholas Ruff, for his perl implementation of des_set_odd_parity,
  which he apparently took from SSLEAY:

  http://seclists.org/pen-test/2005/Jan/0180.html

* Arnaud Pilon, for the details of how to retrieve cached domain, as
  implemented in cachedump.

  http://www.securiteam.com/tools/5JP0I2KFPA.html

* S�bastien Ke, for his cute hexdump recipe:

  http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/142812

LICENSE

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```
