

[TOOL] CacheDump – Recovering Windows Password Cache Entries

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-05/0108.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 05/19/05

To: list@securiteam.com

Date: 19 May 2005 16:28:02 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.secureteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.secureteam.com/maillinglist.html>

CacheDump – Recovering Windows Password Cache Entries

SUMMARY

DETAILS

CacheDump will create a CacheDump NT Service to get SYSTEM right and make his stuff on the registry. Then, it will retrieve the LSA Cipher Key to decrypt (rc4/hmac_md5 GloubiBoulga) cache entries values. A John The Ripper module has been developed to attack the hashed values that are retrieved (timing equivalent to MD4(MD4(password|U(username))).

Download Information:

* The tool's source can be obtained from:

<<http://www.cr0.net:8040/misc/cachedump-1.1.zip>>

<http://www.cr0.net:8040/misc/cachedump-1.1.zip> and at:

<<http://www.off-by-one.net/misc/cachedump-1.1.zip>>

<http://www.off-by-one.net/misc/cachedump-1.1.zip>

* The <<http://www.cr0.net:8040/misc/john-1.6.37-bigpatch-11.diff.gz>>

john bigpatch adds support for a wide range of password hashes to John the Ripper 1.6.37. Among other, it allows offline brute forcing of Windows Cache (mscash) password entries.

Securiteam: [TOOL] CacheDump – Recovering Windows Password Cache Entries

Introduction:

Users authenticate themselves on a Domain Controller (DC) using NTLM/NTLMv2. However the DC sometimes goes offline or the network cable is unplugged; in this situation, the Local Security Authority System Service (LSASS) uses password cache entries from the registry to perform offline logon.

Description of the Authentication Process:

The WINLOGON process displays the msgina dialog and prompts for the username, password and domain. The authentication process itself is handled by LSASS:

WinLogon ----> LSASS ----> LSASRV -> MSV1_0 -> [Registry Cache Entries]

||

MSGina

The most important part of the authentication process happens in MSV1_0.dll. LSASS calls the LSAApLogonUserEx2 function which first checks if the DC is unavailable; in this case, it attempts to match the password entered by the user against the cached password.

The cache entries do not include the authentication credentials in the clear:

an LSA key is used to decrypt them. Credentials are stored in:

HKLM\SECURITY\CACHE\NL\$n with n ranging between 1 and 10. The default ACL does not allow Administrators to read these registry values, which can only be accessed with SYSTEM privileges.

The size of these values may differ but they are roughly composed of 4 parts:

MD CH T EDATA

NL\$ = [metadata in the clear][Text][Text][Encrypted Data]

64 bytes 16 Bytes 16 bytes > 100 bytes

* MD contains several informations about elements of the cache entry structure, such as the username size in the first 2 bytes.

* CH is an array of 16 random(?) bytes used to generate a RC4 key.

* EDATA contains encrypted authentication credential: username (unicode), domain name (unicode), NT-hash, LM-hash (optional). It can be decrypted using the decrypted LSA secret NL\$KM. specific to each computer.

EDATA is decrypted by performing these steps:

0. LSA keyB = DES(NL\$KM, static in-memory LSA keyA)

1. RC4 keyC = HMAC_MD5(LSA keyB, CH)

2. DATA = RC4(EDATA, RC4 keyC);

DATA contains the following informations:

* [96, 102] : MSCASH = MD4(MD4(password) || lowercase(username))

* [168, 168 + username_length * 2] : username

* [168 + username_length * 2 + 2, ...] : domain name

The password hash is salted with the unicode username.

The CacheDump Tool:

CacheDump, licensed under the GPL, demonstrates how to recover cache entry

Securiteam: [TOOL] CacheDump – Recovering Windows Password Cache Entries

information: username and MSCASH. Administrators or security consultants are welcomed to use this program; malicious users can't do anything with it as

long as they do not have Administrator privileges.

CacheDump does not rely on the dll–injection method used in pwdump or lsadump2; it creates a NT service on the fly in order to read the static LSA key from LSASS.EXE's process memory, and deciphers the cache entries to expose the MSCASH values.

CacheDump's output is similar to pwdump's, with of course a different hash function; a plugin for john the ripper password cracker has been developed for offline dictionary and bruteforce cracking.

John The Ripper plugin:

1 – Prerequisites

This plugin for John the Ripper should work on all architectures supported by the cracker. It will run on most unices. Under Microsoft Windows, it will only work under Cygwin.

2 – Installation

Patch John the Ripper version 1.6.37:

```
wget http://www.openwall.com/john/b/john-1.6.37.tar.gz
```

```
tar xzf john-1.6.37.tar.gz
```

```
gunzip -c john-1.6.37.mscash.x.gz | patch -p0
```

Then build john as usual:

```
cd john-1.6.37/src/
```

```
make
```

Installing John is a bit tricky. Version 1.6.37 of John does not include documentation and charset files. Configuring John is beyond the scope of this document. However, you can apply the patch from:

<<http://www.cr0.net:8040/misc/patch-john.html>>

<http://www.cr0.net:8040/misc/patch-john.html>

It features numerous additionnal hashes, and the produced binary should "play nice" with other John packages from the Debian distribution or other distributions.

3 – Usage

John expects the CacheDump output file format. Usernames and hashed passwords should be separated by ':', and there should only be one username/password couple by line. The format MUST be specified on the command line:

```
/john -format:mscash file.txt
```

4 – Technical details

This patch is invasive. John's current framework does not provide support for hashes algorithms that rely on the username to salt the password hashes. Many core files have been patched and there could be various side effects; this patch has only been tested on Linux/i686.

Securiteam: [TOOL] CacheDump – Recovering Windows Password Cache Entries

5 – Example

Cachedump: c:\cachedump.exe

user:2d9f0b052932ad18b87f315641921cda:lab:lab.internal

Copy the result in mscash.txt

c:\cachedump.exe -v

Service not found. Installing CacheDump Service (C:\cachedump.exe -s)

CacheDump service successfully installed.

Service started.

user:2d9f0b052932ad18b87f315641921cda:lab:lab.internal

Service currently active. Stopping service...

Service successfully removed.

John Plugin:

\$./john -format:mscash ./mscash.txt

Loaded 1 password hash (M\$ Cache Hash [mscash])

password (user)

Prevention:

In order to prevent a malicious user from recovering cached passwords, we recommend to:

Revoke local administrator privileges from all users;

Reduce the number of cached password. Change to 1 the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON\CACHEDLOGONSCOUNT

ADDITIONAL INFORMATION

The information has been provided by <mailto:pilon[@]off-by-one.net>

Arnaud Pilon.

To keep updated with the tool visit the project's homepage at:

<<http://www.cr0.net:8040/misc/cachedump.html>>

<http://www.cr0.net:8040/misc/cachedump.html>

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.