

Windows' Domain Cached Credentials v2 — Passlib v1.7.4

Documentation

Archived: 2026-04-05 23:48:05 UTC

New in version 1.6.

This class implements the DCC2 (Domain Cached Credentials version 2) hash, used by Windows Vista and newer to cache and verify remote credentials when the relevant server is unavailable. It is known by a number of other names, including “mscache2” and “mscash2” (Microsoft CAched haSH). It replaces the weaker [msdcc v1](#) hash used by previous releases of Windows. Security wise it is not particularly weak, but due to its use of the username as a salt, it should probably not be used for anything but verifying existing cached credentials. This class can be used directly as follows:

```
>>> from passlib.hash import msdcc2

>>> # hash password using specified username
>>> hash = msdcc2.hash("password", user="Administrator")
>>> hash
'4c253e4b65c007a8cd683ea57bc43c76'

>>> # verify correct password
>>> msdcc2.verify("password", hash, user="Administrator")
True
>>> # verify correct password w/ wrong username
>>> msdcc2.verify("password", hash, user="User")
False
>>> # verify incorrect password
>>> msdcc2.verify("letmein", hash, user="Administrator")
False
```

See also

- [password hash usage](#) – for more usage examples
- [msdcc](#) – the predecessor to this hash

Security Issues¶

This hash is essentially [msdcc v1](#) with a fixed-round PBKDF2 function wrapped around it. The number of rounds of PBKDF2 is currently sufficient to make this a semi-reasonable way to store passwords, but the use of the lowercase username as a salt, and the fact that the rounds can't be increased, means this hash is not particularly future-proof, and should not be used for new applications.

Deviations¶

- Max Password Size

Windows appears to enforce a maximum password size, but the actual value of this limit is unclear; sources report it to be set at assorted values from 26 to 128 characters, and it may in fact vary between Windows releases. The one consistent piece of information is that passwords above the limit are simply not allowed (rather than truncated ala `des_crypt`). Because of this, Passlib does not currently enforce a size limit: any hashes this class generates should be correct, provided Windows is willing to accept a password of that size.

Footnotes

[1]	Description of DCC v2 algorithm - http://openwall.info/wiki/john/MSCash2
-----	---

Source: <https://passlib.readthedocs.io/en/stable/lib/passlib.hash.msdc2.html>