Get started Open in app

Péter Gombos

355 Followers About Follow

LM, NTLM, Net-NTLMv2, oh my!

A Pentester's Guide to Windows Hashes

Péter Gombos Feb 20, 2018 · 4 min read

When attacking AD, passwords are stored and sent in different ways, depending on both where you find it and the age of the domain. Most of these hashes are confusingly named, and both the hash name and the authentication protocol is named almost the same thing. It doesn't help that every tool, post and guide that mentions credentials on Windows manage to add to the confusion. This is my attempt at clearing things up.

This post is geared towards pentesters in an AD environment, and it favors practical attacks against the different hash formats. A lot of inspiration is taken from byt3bl33der's awesome article, "Practical guide to NTLM Relaying in 2017". If I'm missing something, please hit me up.

All example hashes are taken from Hashcat's example hashes page. The hashes I'm looking at is LM, NT, and NTLM (version 1 and 2).

LM

About the hash

LM-hashes is the oldest password storage used by Windows, dating back to OS/2 in the 1980's. Due to the limited charset allowed, they are fairly easy to crack. You can obtain them, if still available, from the SAM database on a Windows system, or the NTDS database on the Domain Controller.

LM was turned off by default starting in Windows Vista/Server 2008, but might still linger in a network if there older systems are still used. It is possible to enable it in later versions through a GPO setting (even Windows 2016/10).

When dumping the SAM/NTDS database, they are shown together with the NTHash, before the colon.

Example

```
299BD128C1101FD6
```

The algorithm

```
1. Convert all lower case to upper case
2. Pad password to 14 characters with NULL characters
3. Split the password to two 7 character chunks
4. Create two DES keys from each 7 character chunk
5. DES encrypt the string "KGS!@#$%" with these two chunks
6. Concatenate the two DES encrypted strings. This is the LM hash.
```

Cracking it

```
john --format=lm hash.txt
hashcat -m 3000 -a 3 hash.txt
```

NTHash (A.K.A. NTLM)

About the hash

This is the way passwords are stored on modern Windows systems, and can be obtained by dumping the SAM database, or using Mimikatz. They are also stored on domain controllers in the NTDS file. These are the hashes you can use to pass-the-hash.

Usually people call this the NTLM hash (or just NTLM), which is misleading, as Microsoft refers to this as the NTHash (at least in some places). I personally recommend to call it the NTHash, to try to avoid confusion.

Example

```
B4B9B02E6F09A9BD760F388B67351E2B
```

The algorithm

```
MD4(UTF-16-LE(password))
```

UTF-16-LE is the little endian UTF-16. Windows used this instead of the standard big endian, because *Microsoft*.

Cracking it

```
john --format=nt hash.txt
hashcat -m 1000 -a 3 hash.txt
```

NTLMv1 (A.K.A. Net-NTLMv1)

About the hash

The NTLM protocol uses the NTHash in a challenge/response between a server and a client. The v1 of the protocol uses both the NT and LM hash, depending on configuration and what is available. The Wikipedia page on NT Lan Manager has a good explanation.

A way of obtaining a response to crack from a client, Responder is a great tool. The value to crack would be the `K1 | K2 | K3` from the algorithm below. Version 1 is deprecated, but might still be used in some old systems on the network.

Example

```
u4-
netntlm::kNS:338d08f8e26de93300000000000000000000000000000000:9526fb8
c23a90751cdd619b6cea564742e1e4bf33006ba41:cb8086049ec4736c
```

The algorithm

```
C = 8-byte server challenge, random
K1 | K2 | K3 = LM/NT-hash | 5-bytes-0
response = DES(K1,C) | DES(K2,C) | DES(K3,C)
```

Cracking it

```
john --format=netntlm hash.txt
hashcat -m 5500 -a 3 hash.txt
```

NTLMv2 (A.K.A. Net-NTLMv2)

About the hash

This is the new and improved version of the NTLM protocol, which makes it a bit harder to crack. The concept is the same as NTLMv1, only different algorithm and responses sent to the server. Also captured through Responder or similar. Default in Windows since Windows 2000.

Example

```
admin::N46iSNekpT:08ca45b7d7ea58ee:88dcbe4446168966a153a0064958dac6:5
c7830315c7830310000000000000b45c67103d07d7b95acd12ffa11230e0000000052
920b85f78d013c31cdb3b92f5d765c783030
```

The algorithm

```
SC = 8-byte server challenge, random
CC = 8-byte client challenge, random
```

```
  CC* = (X, time, CC2, domain name)
  v2-Hash = HMAC-MD5(NT-Hash, user name, domain name)
  LMv2 = HMAC-MD5(v2-Hash, SC, CC)
  NTv2 = HMAC-MD5(v2-Hash, SC, CC*)
  response = LMv2 | CC | NTv2 | CC*
```

Cracking it

```
  john --format=netntlmv2 hash.txt
  hashcat -m 5600 -a 3 hash.txt
```

IN SUMMARY

LM- and NT-hashes are ways Windows stores passwords. NT is confusingly also known as NTLM. Can be cracked to gain password, or used to pass-the-hash.

NTLMv1/v2 are challenge response protocols used for authentication in Windows environments. These use the NT-hash in the algorithm, which means it can be used to recover the password through Brute Force/Dictionary attacks. They can also be used in a relay attack, see byt3bl33d3r's article [1].

If you're still confused, I would recommend reading the Wikipedia articles. I do hope this intro clears up the confusing language and can somehow help you.

Sources

[1] https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-under-5-minutes.html

[2] https://technet.microsoft.com/en-us/library/dd277300.aspx#ECAA

[3] https://en.wikipedia.org/wiki/LAN_Manager

[4] https://en.wikipedia.org/wiki/NT_LAN_Manager

[5] https://en.wikipedia.org/wiki/Security_Account_Manager

[6] https://hashcat.net/wiki/doku.php?id=example_hashes

Security Pentesting Active Directory Windows Passwords