Hash Encryption in Windows 10 Anniversary Update

Passcape Software

1 Abstract

Microsoft recently introduced a large anniversary update for Windows 10, one of its most popular operating systems.

Windows 10 Anniversary Update makes Windows 10 better than ever before. You can enjoy a multitude of new features, including:

- shuffle buttons in the Start menu
- take advantage of additional highly requested adware
- resize and reshape the adorable tiles
- admire how fast your personal data is sent to Microsoft
- try a brilliant monochrome skin for those suffering from moon blindness
- view brand new minimalistic icons developed by thousands of hard-working UI designers
- spend even more time searching for system options across multiple windows, thus raising the bar for your extrasensory perception

In all seriousness, however, the update does actually introduce some important improvements that deserve our attention. These include a Linux shell, pure re-installation, improved intelligence in Cortana, new login options based on Windows Hello, and much more.

The funny thing is that, despite the fact that the standard login workflow of Windows 10 has been slightly changed, this is not mentioned in the release notes at all. Due to these slight, yet significant changes, most hacker tools for pulling password hashes out of Windows will not work anymore. These changes may have been motivated by Microsoft's desire to discontinue support for legacy and vulnerable cryptographic algorithms. In our example, Microsoft has decided to discontinue support for RC4. Luckily, the latest version of Windows Password Recovery that is used for auditing Windows security has already got support for new SAM encryption scheme.

2 Hash Encryption in Windows 10 Anniversary Update

According to Microsoft, user passwords are stored as hashes (instead of plain-text representation) that can be accessed in the corresponding section of the Windows registry (only by the system itself):

HKLM/SAM/SAM/Domains/Account/users/<RID>/V.

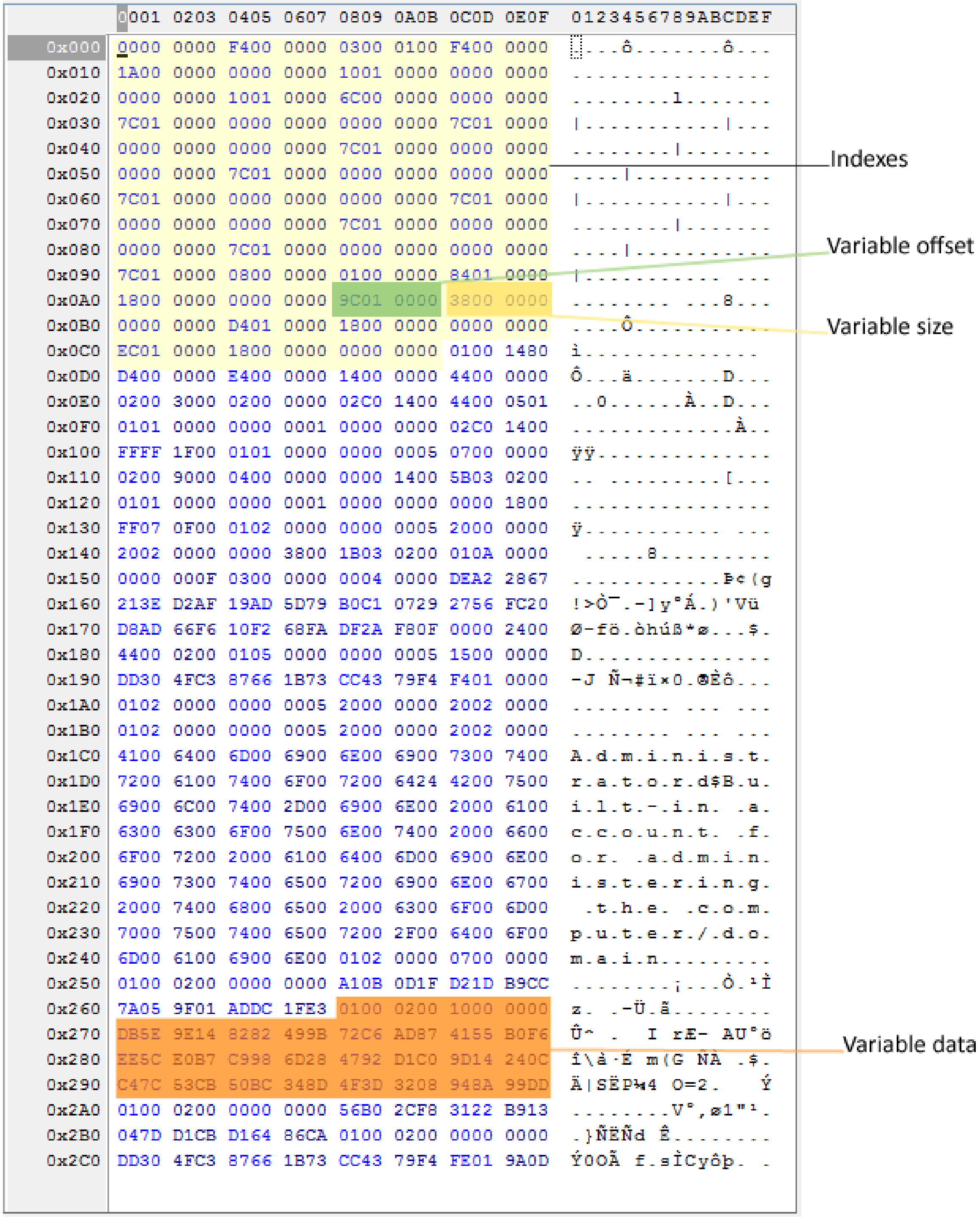
Where **<RID>** - is the unique user ID.

Unique user IDs can be figured out by scanning the following registry tree:

HKLM/SAM/SAM/Domains/Account/users/names/<NAME>

Each key containing a username is associated with a corresponding RID. For example, the RID of the Administrator account is always equal to 500 (0x1F4 in the hexadecimal notation), while the Guest's RID is 501 (0x1F5).

Any user's registry key also holds at least '*C*' and 'V' records. A '*V*' record contains the variable-length data that corresponds to this account. The names themselves seem to be abbreviations – '*V*' stands for 'variable' and '*C*' means 'constant.' Each variable in the '*V*' records is represented as a constant within the interval from 0 to 0xCC, e.g. a username is encoded as 0xC. Therefore, if we know the constant, we can identify an offset to the index that refers to actual data. LM and NT hashes correspond to 0x9C and 0xA8 respectively. However, obtaining the final password hash will require several additional decryption steps.



```
       0001 0203 0405 0607 0809 0A0B 0C0D 0E0F  0123456789ABCDEF
0x000  0000 0000 F400 0000 0300 0100 F400 0000  ....ô.......ô...
0x010  1A00 0000 0000 0000 1001 0000 0000 0000  ................
0x020  0000 0000 1001 0000 6C00 0000 0000 0000  ........l.......
0x030  7C01 0000 0000 0000 0000 0000 7C01 0000  |...........|...
0x040  0000 0000 0000 0000 7C01 0000 0000 0000  ........|.......
0x050  0000 0000 7C01 0000 0000 0000 0000 0000  ....|...........
0x060  7C01 0000 0000 0000 0000 0000 7C01 0000  |...........|...
0x070  0000 0000 0000 0000 7C01 0000 0000 0000  ........|.......
0x080  0000 0000 7C01 0000 0000 0000 0000 0000  ....|...........
0x090  7C01 0000 0800 0000 0100 0000 8401 0000  |........... ...
0x0A0  1800 0000 0000 0000 9C01 0000 3800 0000  ........ ...8...
0x0B0  0000 0000 D401 0000 1800 0000 0000 0000  ....Ô...........
0x0C0  EC01 0000 1800 0000 0000 0000 0100 1480  ì..............
0x0D0  D400 0000 E400 0000 1400 0000 4400 0000  Ô...ä.......D...
0x0E0  0200 3000 0200 0000 02C0 1400 4400 0501  ..0......À..D...
0x0F0  0101 0000 0000 0001 0000 0000 02C0 1400  .............À..
0x100  FFFF 1F00 0101 0000 0000 0005 0700 0000  ÿÿ..............
0x110  0200 9000 0400 0000 0000 1400 5B03 0200  .. .........[...
0x120  0101 0000 0000 0001 0000 0000 0000 1800  ................
0x130  FF07 0F00 0102 0000 0000 0005 2000 0000  ÿ........... ...
0x140  2002 0000 0000 3800 1B03 0200 010A 0000   .....8.........
0x150  0000 000F 0300 0000 0004 0000 DEA2 2867  ............Þ¢(g
0x160  213E D2AF 19AD 5D79 B0C1 0729 2756 FC20  !>Ò¯.-]y°Á.)'Vü
0x170  D8AD 66F6 10F2 68FA DF2A F80F 0000 2400  Ø-fö.òhúß*ø...$.
0x180  4400 0200 0105 0000 0000 0005 1500 0000  D...............
0x190  DD30 4FC3 8766 1B73 CC43 79F4 F401 0000  -J Ñ¬‡ï×0.®Èô...
0x1A0  0102 0000 0000 0005 2000 0000 2002 0000  ........ ... ...
0x1B0  0102 0000 0000 0005 2000 0000 2002 0000  ........ ... ...
0x1C0  4100 6400 6D00 6900 6E00 6900 7300 7400  A.d.m.i.n.i.s.t.
0x1D0  7200 6100 7400 6F00 7200 6424 4200 7500  r.a.t.o.r.d$B.u.
0x1E0  6900 6C00 7400 2D00 6900 6E00 2000 6100  i.l.t.-.i.n. .a.
0x1F0  6300 6300 6F00 7500 6E00 7400 2000 6600  c.c.o.u.n.t. .f.
0x200  6F00 7200 2000 6100 6400 6D00 6900 6E00  o.r. .a.d.m.i.n.
0x210  6900 7300 7400 6500 7200 6900 6E00 6700  i.s.t.e.r.i.n.g.
0x220  2000 7400 6800 6500 2000 6300 6F00 6D00   .t.h.e. .c.o.m.
0x230  7000 7500 7400 6500 7200 2F00 6400 6F00  p.u.t.e.r./.d.o.
0x240  6D00 6100 6900 6E00 0102 0000 0700 0000  m.a.i.n.........
0x250  0100 0200 0000 0000 A10B 0D1F D21D B9CC  ........¡...Ò.¹Ì
0x260  7A05 9F01 ADDC 1FE3 0100 0200 1000 0000  z. .-Ü.ã........
0x270  DB5E 9E14 8282 499B 72C6 AD87 4155 B0F6  Û^ .  I rÆ- AU°ö
0x280  EE5C E0B7 C998 6D28 4792 D1C0 9D14 240C  î\à·É m(G ÑÀ .$.
0x290  C47C 53CB 50BC 348D 4F3D 3208 948A 99DD  Ä|SËP¼4 O=2.   Ý
0x2A0  0100 0200 0000 0000 56B0 2CF8 3122 B913  ........V°,ø1"¹.
0x2B0  047D D1CB D164 86CA 0100 0200 0000 0000  .}ÑËÑd Ê........
0x2C0  DD30 4FC3 8766 1B73 CC43 79F4 FE01 9A0D  Ý0OÃ f.sÌCyôþ. .
```

Let's see how the system generally retrieves the NTLM hash of a user:

1. First of all, the system identifies a path to the key in the Windows registry where the account settings are stored, e.g. **HKLM/SAM/SAM/Domains/Account/Users/00001F4**

2. The next step is to read the variable that contains the NTLM hash. This variable corresponds to the constant 0xA8. The system thus reads the data index based on the offset in this constant, i.e. 0x19C. Adding the data index to 0xCC will give the offset 0x268 from which we can access the actual data (our 'raw' NTLM hash) as shown in the picture. Now the system can read the hash and decrypt it.
3. Using **SYSKEY**, the system decrypts the SAM session key. The SAM session key is stored in the registry section called **HKLM/SAM/SAM/Domains/Account/V**. This data structure actually keeps two encryption keys: the current one and the previous one. In this step, the system uses the **MD5** and **RC4** algorithms. In Windows 10 Anniversary Update, RC4 has been replaced with **AES**.
4. The system then uses the SAM session key to decrypt the 'raw' hash obtained in Step 2 through the RC4 or AES (for Windows 10 Anniversary Update) algorithm.
5. And, finally, the data that has been obtained is transformed once again into the actual data by means of the **DES** algorithm and the user's RID as the encryption key. Now our NTLM hash is ready.

As you can see, in Windows 10 Anniversary Update the **RC4** stream cipher in Steps 3 and 4 has been replaced with the **AES** block cipher. This has led to certain changes in the data storage structure (at least because the data length in AES blocks must be multiple to 16 bytes) but has not resulted in stronger security of the operating system.

3 Conclusion

In Windows 10 AU, the encryption algorithms of SAM accounts have been changed. Did the new algorithms make password hashes safer? No. Was it worth it? Yes, since the unified changes applied to domain users as well – some of their private data was at risk of being compromised due to vulnerabilities in the legacy RC4 algorithm. However, that is another matter entirely.