# Farewell Syskey!


Passcape Software

1. Introduction 3

2. How the Syskey works 3

2.1 What is the Syskey ..... 3

2.2 The syskey.exe utility ..... 3

2.3 The Syskey encryption key storage ..... 3

2.3.1 System registry ..... 4

2.3.2 Startup diskette ..... 4

2.3.3 Startup password ..... 4

2.4 Good-bye the syskey.exe ..... 4

3. Does the Syskey obsolete? 5

3.1 The syskey.exe is no longer considered secure ..... 5

3.2 Weak cryptography ..... 5

3.3 Limitted data protection ..... 5

3.4 Used by hackers ..... 6

4. Conclusion 6

5. Appendix 1. Deriving SAM encryption key 7

6. Appendix 2. Decrypting Syskey 7

# 1 Introduction

Shortly before the official release of the next big update for Windows 10 and Windows Server 2016 operating systems, Microsoft has published a list of changes in new versions. Among the components for replacement and removal, is the syskey.exe utility.
Microsoft explains the fact for syskey.exe deletion this way:

- The syskey encryption key and the use of syskey.exe are no longer considered secure
- Syskey is based on weak cryptography that can easily be broken in modern times
- The data that is protected by syskey is very limited and does not cover all files or data
- The syskey.exe utility is used by hackers as a part of ransomware scam

Many news portals and sites paid no attention to the news, although this is a landmark event. Knowing about the functioning of the Syskey from within, we will try to clarify and show you that most of the arguments against Syskey is not true. At least there is yet a specific innuendo. Are you Intrigued and want to know why? Go ahead then.

# 2 How the Syskey works

## 2.1 What is the Syskey

First, let's figure out what the **Syskey** is. Syskey is a transparent software layer that is aimed to enforce critical data protection like users' hashes, passwords, system secrets, etc. It was also designed to prevent offline attacks on SAM database and protect users' hashes from being attacked using rainbow tables.

## 2.2 The syskey.exe utility

Syskey was first introduced in Windows NT 4.0 SP3. By default, this extra protection is included in Windows XP and later OSes, and **cannot be turned off**. You may have wondered why the last phrase was bolded? All right. Many sites misinterpreted or gave the news of the total rejection of the Syskey. That's wrong. Microsoft refuses the use of syskey.exe utility only, but the Syskey protection will stay as it is. As will be shown below, this is a big difference.

## 2.3 The Syskey encryption key storage

On the physical level, Syskey is just an encryption algorithm with a unique 128-bit key. And the syskey.exe utility just provides the way the Syskey encryption key is stored in the system. There are three possible storage areas:

- System registry

- Startup Diskette
- Startup password

### 2.3.1 System registry

By default, the Syskey is stored in the registry and scattered across the following four branches:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\JD
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Skew1
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Data
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\GBG

Access to these branches is denied to all except the system. But having admin rights, it is not that hard to read, extract and decrypt it.

### 2.3.2 Startup diskette

If you set up the syskey.exe to use this method of storage, each time you boot your PC you will be prompted for a floppy with Syskey on it. The disk will also be required in order to opt out of this type of storage.

### 2.3.3 Startup password

In this case, the key is not stored anywhere and is generated every time system starts. Once you select the option, the user will be prompted to type in the startup password on every system boot before the user login screen appears. The Syskey is then generated hashing the input password using an MD5 algorithm. Without providing the correct startup password, the user will not be able to move on and load the system.

## 2.4 Good-bye the syskey.exe

So, the new versions of Windows 10 and Windows Server 2016 will have no syskey.exe utility. Therefore, there will be no way to select a method of storing Syskey because it will always reside in the registry! But what's the catch? The catch is that the last two ways of storing the Syskey, which are to be thrown away, ensure stronger protection for users' data. Even though not so many users know about it, computer professionals and security gurus enjoy the stronger protection. For example, once a user switches the Syskey to require the startup diskette, no intruders or hackers can get their hands on the user's personal information, passwords or EFS-encrypted files even if they gain physical access to the PC. No chance to decrypt the data unless the startup diskette is provided.

# 3 Does the Syskey obsolete?

Let's go further through all the reasons for the syskey.exe removal referenced by Microsoft.

## 3.1 The syskey.exe is no longer considered secure

***"The syskey encryption key and the use of syskey.exe are no longer considered secure".***

That's wrong. Refusing to store Syskey in the registry allows you to improve the safety of your data against offline attacks drastically. For example, when the Syskey startup password is set, no attacker can decrypt your Skype/IE/Chrome/WiFi/LAN passwords (without providing the startup password), even if he can gain access to and read the data. After the Windows 10 Fall Creator Update is out, it will be much easier for potential malefactors.

## 3.2 Weak cryptography

***"Syskey is based on weak cryptography that can easily be broken in modern times".***

Not exactly. The algorithm for deriving SAM encryption key out of Syskey key is shown at the end of the article. This algorithm, invented about two decades ago, of course, has some drawbacks: the hash function MD5 is considered to be compromised, weak stream cipher RC4 is out of date too. However, in recent versions of Windows 10, the encryption algorithms were replaced with SHA-256 and AES correspondingly. Let's try to calculate how quickly one can break startup Syskey password or Syskey diskette. The Startup password guessing speed varies around 10-20 million passwords per second for the old algorithm (and much lower for the new one), depending on the hardware used. Let's take the most pessimistic scenario and round this value up to 100. But even in this case, the cracking speed will be about 100 times slower compared to Windows account passwords. To guess the password 'Letmein123' using full brute force will take over 260 years. If you want to brute-force the startup diskette, you will have to wait for the death of the solar system.

## 3.3 Limitted data protection

***"The data that is protected by syskey is very limited and does not cover all files or data".***

It is not very clear what is meant. Most likely the inability to protect some files with EFS encryption. For example, the system registry. As an alternative, the Microsoft proposes to use

BitLocker instead. The funny thing is that when you enable BitLocker for the Microsoft account, the data recovery key is automatically sent to the Microsoft servers. In a domain, all BitLocker recovery keys are stored in Active Directory, and anyone who has access to the server can easily decrypt any PC in the domain. We already outlined the security weakness in our previous article dedicated to security flaw of the domain accounts.

## 3.4 Used by hackers

***"The syskey.exe utility is used by hackers as a part of ransomware scam".***

In 2014-2017 Windows users have faced one of the most extensive attacks using the built-in syskey.exe utility. A user usually got a phone call. Someone at the other end with a thick Indian accent and an incredibly Anglicized name like John Smith or something like that told that the computer had been compromised and needed to be repaired. After the unsuspecting victim opened the access to his/her computer, the attacker ran the syskey.exe tool and set the startup Syskey password. The main problem of the recovery computers after the syskey-attack is that it is not enough to reset the Syskey or switch it back to be stored in registry again. To recover the system fully, it is required to know the Syskey startup password.

Without a doubt, new Microsoft update will bring this fraudulent scheme to zero. But it too naive to hope or claim that the scammers will never use other schemes of deception. Microsoft security guys should instead tell us, what can stop scammers from resetting user logon password instead of Syskey? It is even much easier to do that. Do you know that after a scammer gains access to the victim's PC, he can just use the command prompt with administrator privileges and then type in
**NET USER <USERNAME> <NEWPASSWORD>**
to set the new password without knowing the current one.

# 4 Conclusion

So, after the new release of Windows 10 Fall Creators Update and the death of the syskey.exe, all user's files and passwords will become more open to both scammers and intelligence agencies. If Microsoft continues its thoughtless policy of simplifying access to users' personal data, this will end up migrating common users, organizations and entire countries to other operating systems.

One final chord left to do is to stop support for regular local accounts, having left Microsoft accounts only. After this step, users will no longer be able to consider their private data as completely secure. On the other hand, for the average network user, this is more of a plus.The consciousness of the fact that your whole private life can become not only yours but not quite private is depressing.

Is Syskey good or evil? decide for yourself.

## 5 Appendix 1. Deriving SAM encryption key

```
BYTE hash[16], pDecodedData[32];
static BYTE samc1[]="!@#$%^&*()qwertyUIOPAzxcvbnmQQQQQQQQQQQQ)(*@&%";
static BYTE samc2[]="0123456789012345678901234567890123456789";

//Generate decrypt key
Md5Init();
Md5Update(pSamSessionKey+8,16);
Md5Update(samc1,0x2F);
Md5Update(m_pSyskey,16);
Md5Update(samc2,0x29);
Md5Final(hash);

//Decrypt
memcpy(pDecodedData,pSamSessionKey+24,32);
Rc4SetKey(hash,16);
Rc4Decrypt(pDecodedData,32);

//Check sign
Md5Init();
Md5Update(pDecodedData,16);
Md5Update(samc2,0x29);
Md5Update(pDecodedData,16);
Md5Update(samc1,0x2F);
Md5Final(hash);

if( memcmp(pDecodedData+16,hash,16)==0 )
{
   memcpy(pDecodedSamSessionKey,pDecodedData,16);
   return TRUE;
}
return FALSE;
```

## 6 Appendix 2. Decrypting Syskey

```
BOOL DecryptSyskey(LPCTSTR szJD, LPCTSTR szSkew1, LPCTSTR szGBG, LPCTSTR szData)
{
   BYTE p[]={0xb,0x6,0x7,0x1,0x8,0xa,0xe,0x0,0x3,0x5,0x2,0xf,0xd,0x9,0xc,0x4};
   BYTE pEncryptedKey[16];

   if( !szJD || !szSkew1 || !szGBG || !szData )
      return FALSE;

   //convert to binary
   _stscanf_s(szJD,_T("%X"),(LPDWORD)&pEncryptedKey[0]);
   _stscanf_s(szSkew1,_T("%X"),(LPDWORD)&pEncryptedKey[4]);
   _stscanf_s(szGBG,_T("%X"),(LPDWORD)&pEncryptedKey[8]);
   _stscanf_s(szData,_T("%X"),(LPDWORD)&pEncryptedKey[12]);
```

```
    //Permutate
    for( int i=0; i<16; i++ )
        m_pSyskey[i]=pEncryptedKey[p[i]];

    return TRUE;
}
```