Прощай, Syskey!

Passcape Software

1 Введение

Незадолго до официального выхода очередного большого обновления операционных систем Windows 10 Fall Creators Update и Windows Server 2016 RS3 компания Microsoft опубликовала список изменений в новых версиях. Среди компонентов для замены и удаления значится утилита syskey.exe. Microsoft объясняет этот факт так:

- В современных реалиях эта утилита не обеспечивает необходимый уровень безопасности данных
- Устаревшие технологии криптографии Syskey легко взламываются
- Утилита не защищает все критически важные данные
- Часто используется хакерами при развёртывании вредоносного программного обеспечения

Многие информационные порталы и агентства обошли эту новость стороной, хотя событие знаковое. Зная о функционировании Syskey не понаслышке, мы постараемся разобраться и показать, что большинство аргументов против Syskey, мягко выражаясь, неправда. По крайней мере, присутствует определенная недосказанность. Заинтригованы и хотите узнать подробности? Тогда приступим.

2 Принцип работы Syskey

2.1 Что такое Syskey

Для начала давайте разберемся, чтобы представляет из себя Syskey. **Syskey** это прозрачная программная надстройка, придуманная для дополнительной защиты хэшей пользователей, предотвращения оффлайн атак на базу данных хэшей SAM и защиты хэшей пользователей от атак с использованием радужных таблиц.

2.2 Утилита syskey.exe

Впервые Syskey была представлена в операционной системе Windows NT 4.0 SP3. По-умолчанию функция дополнительной защиты изначально включена в Windows XP и более поздних операционных системах, и **отключить ее невозможно**. Вы уже догадались, почему последняя фраза выделена жирным? Все верно. Многие сайты неправильно перевели или преподнесли новость отказа от Syskey. Компания Microsoft отказывается не от защиты Syskey, а от утилиты syskey.exe. Как будет показано ниже, это большая разница.

2.3 Хранение ключа шифрования Syskey

На физическом уровне Syskey представляет собой алгоритм шифрования при помощи уникального 128-битного ключа Syskey. А утилита syskey.exe просто обеспечивает выбор места хранения этого ключа. Всего предусмотрено три места хранения:

- Реестр компьютера
- Дискета запуска
- Стартовый пароль

2.3.1 Реестр компьютера

По-умолчанию, ключ Syskey хранится именно в реестре и разбросан по четырем веткам

```
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
```

Доступ к этим веткам закрыт всем, кроме системы. Но имея права администратора, несложно прочитать и извлечь Syskey.

2.3.2 Дискета запуска

Если в утилите syskey.exe выбрать этот способ хранения ключа, то каждый раз при загрузке ПК будет запрошена дискета с сохраненным на ней 128-битным ключом. Дискета потребуется также и для того, чтобы отказаться от этого типа хранения.

2.3.3 Стартовый пароль

В этом случае, ключ Syskey нигде не хранится, а создается каждый раз из введенного при загрузке системы стартового пароля. Если выбрать этот пункт, то при загрузке системы, до появления экрана выбора пользователей, Windows будет запрашивать начальный пароль, без которого дальнейшее продолжение работы невозможно. При изменении способа хранения ключа на другой, аналогично требуется знание начального пароля.

2.4 Отказ от поддержки syskey.exe

Итак, в новой версии Windows 10 и Windows Server 2016 будет удалена утилита syskey.exe. Следовательно, нельзя будет выбрать способ хранения ключа Syskey. Теперь он всегда будет храниться в реестре! "А в чем подвох?", - спросите вы. Подвох в том, что те способы хранения Syskey, которые пойдут под нож, обеспечивают более сильную защиту персональных данных пользователей. И хотя им пользовались единицы, профессионалы в области компьютерной безопасности знают, что если, например, выбрать способ хранения Syskey на дискете, то без этой дискеты никакой злоумышленник, даже если он получить физический доступ к компьютеру, не сможет ни расшифровать файлы, зашифрованные EFS, ни персональные данные учетной записи.

3 Причины отказа от syskey.exe

Давайте пройдемся по причинам отказа от syskey.exe, на которые указывает Microsoft.

3.1 Не обеспечивает безопасность данных

"В современных реалиях эта утилита не обеспечивает необходимый уровень безопасности данных"

Неверно. Хранение Syskey на дискете или генерация его по начальному паролю позволяет кардинально улучшить сохранность персональных данных пользователя при оффлайн атаках. Например, при установленном начальном пароле Syskey, даже если злоумышленник получит доступ к паролям skype/IE/chrome/wifi/lan и т.д., то без знания начального пароля Syskey расшифровать их не получится. После того, как выйдет очередное обновление Windows и ключ Syskey будет храниться только в реестре, жизнь потенциальных злоумышленников заметно упростится.

3.2 Устаревшие технологии криптографии

"Устаревшие технологии криптографии Syskey легко взламываются"

Не совсем так. Алгоритм получения ключа шифрования учетных записей SAM из ключа Syskey показан в конце статьи. Этот алгоритм, придуманный около двух десятков лет назад, несомненно, имеет свои минусы: скомпрометированная хэш функция MD5 и слабый потоковый алгоритм шифрования RC4. Кстати, в последних версиях Windows 10 они были заменены на SHA-256 и AES. Давайте попробуем посчитать, насколько быстро можно взломать защиту с установленным паролем или дискетой Syskey. Скорость подбора пароля Syskey на современном оборудовании составляет порядка

10-20 миллионов паролей в секунду для старого алгоритма (для нового - намного медленнее). Мы будем пессимистами и округлим это значение до 100. Но даже при таком раскладе взлом пароля Syskey будет проходить примерно в 100 раз медленнее, чем пароля учетной записи Windows. На перебор Syskey пароля 'Letmein123' потребуется пара с половиной сотен лет. А вот подбор ключа, хранимого на дискете, произойдет не раньше смерти солнечной системы.

3.3 Не защищает все важные данные

"Утилита не защищает все критически важные данные"

Не очень понятно, что именно имеет в виду Misrosoft. Скорее всего невозможность защиты некоторых файлов при помощи EFS. Например, системного реестра. В качестве альтернативы Microsoft предлагает использовать Bitlocker. Самое забавное при этом, что при включении шифрования Bitlocker для учетной записи Microsoft, ключ восстановления автоматически отправляется на серверы Microsoft. В домене все ключи восстановления Bitlocker хранятся в Active Directory, и любой, имеющий доступ к серверу, может легко расшифровать какой угодно ПК в домене. Мы уже писали об этом в одной из наших статей про уязвимость паролей доменных пользователей.

3.4 Часто используется хакерами

"Часто используется хакерами при развёртывании вредоносного программного обеспечения"

В 2014-2017 годах пользователи Windows подверглись одной из самых обширных атак с использованием встроенной утилиты syskey.exe. Как правило, это выглядело следующим образом: пользователю поступал звонок от якобы технической поддержки Microsoft. Человек на другом конце провода с сильным Индийским акцентом и невероятно английским именем, типа John Smith, утверждал, что на компьютере пользователя установлена вредоносная программа и для ее устранения требуется доступ к ПК. После того, как доверчивая жертва открывала доступ к своему компьютеру, злоумышленники запускали утилиту syskey.exe и устанавливали там стартовый пароль Syskey. Основная проблема восстановления компьютеров после syskey-атаки заключается в недостаточности сброса настроек и ключа Syskey. Для полной дешифрации требуется знание стартового пароля.

Без сомнений, отказ от syskey.exe сведет эту мошенническую схему к нулю. Но надеяться, что мошенники не будут использовать другие схемы обмана, по крайней мере, наивно. Люди, отвечающие за безопасность в Microsoft, скажите, кто помешает скаммерам менять не Syskey, а к примеру, пароль учетной записи? Сброс пароля на вход еще проще, чем установка Syskey. После получения доступа к компьютеру жертвы хакеру достаточно просто ввести в командной строке с правами администратора такую команду:

net user USERNAME NEWPASSWORD
Знание старого пароля при этом необязательно.

4 Выводы

Таким образом, после выхода обновления Windows 10 Fall Creators Update и отказа от утилиты syskey.exe, файлы и пароли пользователей станут более открыты как для мошенников, так и для спецслужб. Если Microsoft продолжит свою бездумную политику упрощения доступа к персональным данным пользователей, это кончится тем, что обычные пользователи, организации и целые страны начнут постепенно мигрировать на другие операционные системы. Достаточно сделать один финальный аккорд: прекратить поддержку обычных локальных учетных записей, оставив только учетные записи Microsoft. После этого шага пользователи больше не смогут рассматривать свои приватные данные как полностью безопасные. От слова, совсем. С одной стороны, для среднестатистического порядочного пользователя Сети это скорее плюс. С другой стороны, угнетает сознание того, что вся твоя частная жизнь может стать не только твоей, да и не совсем частной. Благо это или зло - решайте сами.

5 Приложение 1. Алгоритм получения ключа шифрования SAM из ключа Syskey

```
BYTE hash[16], pDecodedData[32];
static BYTE samc1[]="!@#$%^&*()qwertyUIOPAzxcvbnmQQQQQQQQQQ)(*@&%";
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 Приложение 2. Расшифровка Syskey из реестра

```
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;
}
```