Trojan.Mayachok.2: анализ первого известного VBR-буткита

Published: 2011-07-08 · Archived: 2026-04-05 20:11:11 UTC

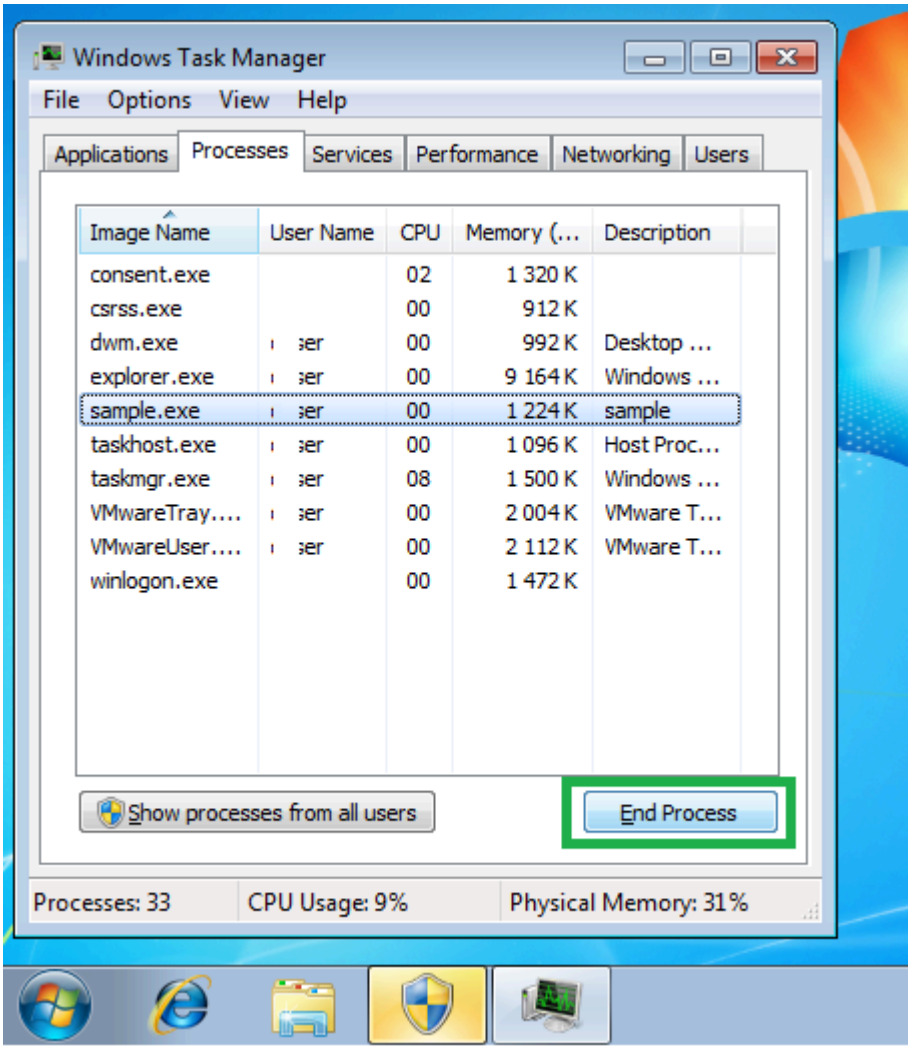
8 июля 2011 года

В процессе изучения статистики появившихся за последнее время угроз складывается впечатление, что модификация загрузочной записи в процессе инфицирования — новое модное веяние среди разработчиков вредоносного ПО. Особое место среди подобных программ занимает троянец Trojan.Mayachok.2, инфицирующий Volume Boot Record и нарушающий работу популярных браузеров.

Заражение системы

Перед началом атаки на инфицируемый компьютер дроппер вредоносной программы проверяет зараженность системы. Для этого на основе серийного номера системного раздела генерируется CLSID и проверяется его наличие в системном реестре: если в ветке **HKLM\Software\Classes\CLSID** отсутствует соответствующий раздел, то заражение продолжается.

В операционных системах Windows Vista и Windows 7 троянец пытается повысить собственные права. Этот весьма примитивный способ уже не раз использовался другими вредоносными программами: постоянный перезапуск самой себя с запросом на повышение привилегий. Опытный пользователь с легкостью может завершить такой назойливый процесс в «Диспетчере задач».



Завершение троянского процесса через «Диспетчер задач»

Дроппер несет в себе как 32-битный, так и 64-битный драйвер, в будущем способный обеспечить загрузку основного функционала данной вредоносной программы. На диске сохраняется соответствующий драйвер в зависимости от разрядности пользовательской операционной системы. Он может быть записан как в начало диска (до первого активного раздела), если там достаточно места, так и в его конец. Несложно заметить, что в логике троянца присутствует ошибка: так, например, если загрузочным разделом окажется не первый, то троянский драйвер может перезаписать случайные данные любого раздела до загрузочного, т. к. позиция для записи выбирается случайно в пределах свободных (как считает троянец) секторов.

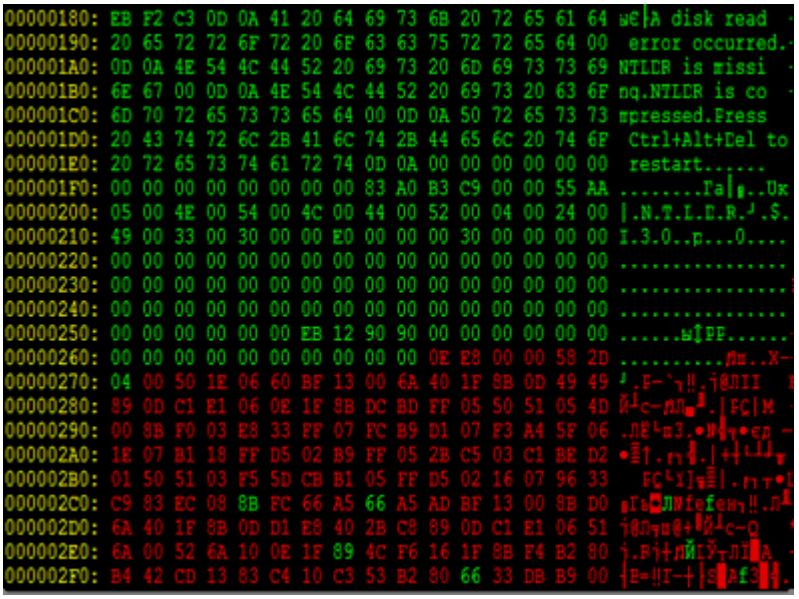
Только после этого начинается заражение VBR (Volume Boot Record). Вредоносная программа отказывается от заражения, если файловая система раздела имеет формат, отличный от NTFS. Анализируя загрузочную запись, троянец находит удобное место для своего размещения и перезаписывает имеющийся там код. Оригинальный код упаковывается при помощи библиотеки aplib (http://www.ibsensoftware.com) и дописывается следом за вирусным. Номер начального сектора размещенного ранее на диске драйвера и его размер также «прошиваются» в тело зараженной VBR.

Здесь следует еще раз сказать о необычности механизма заражения. Вирусы, модифицирующие MBR (Master Boot Record) и BOOT-секторы, известны еще со времен DOS, в то время как современные ОС

предоставляют новые возможности, в том числе и для вирусописателей. В рассматриваемом нами случае BOOT-сектор является первым сектором VBR, который, например, для раздела NTFS занимает 16 секторов. Таким образом, классическая проверка только загрузочного сектора не может обнаружить вредоносный объект, т. к. он располагается дальше — внутри VBR.

После заражения системы **Trojan.Mayachok.2** сбрасывает на диск небольшое приложение, предназначенное для автоматической перезагрузки системы. Стоит отметить, что аналогичным образом ведет себя и другой буткит — Trojan.Hashish. В завершение своей работы троянец пытается «замести следы» и удалить себя.

Запуск из VBR



Сравнение первых секторов VBR чистой и зараженной системы, красным показаны различия — код вирусного загрузчика

Получив управление, вирусный загрузчик действует по классической для MBR/BOOT-вирусов схеме. «Откусывает» себе небольшой кусок системной памяти, переносит себя туда и перехватывает прерывание int 13h для просмотра содержимого считываемых с диска секторов. Затем он целиком загружает с диска свой драйвер и распаковывает на прежнее место оригинальный код VBR. Управление возвращается системному загрузчику.

Далее идет череда снятий/установок перехватов в загружаемых модулях, таких как ntldr, bootmgr, osloader.exe, winload.exe и т. д., в зависимости от используемого операционной системой загрузчика. Следует отметить, что помимо обычных перехватов (сплайсинга) в ключевых мостах используются аппаратные отладочные регистры (dr0-dr7) и трассировка (пошаговое исполнение) кода. Это придает универсальность троянцу и одновременно является естественным способом обхода защиты целостности некоторых загрузочных модулей. В итоге в области памяти режима ядра (kernelmode memory) оказывается загруженный и готовый к работе вирусный драйвер.

Драйвер загрузчика

Точка выхода вирусного драйвера вызывается дважды. Это связано с тесной работой зараженного VBR и драйвера. Поскольку код вирусного VBR составляет всего 2078 байт, часть функционала авторы решили перенести в тело драйвера. При первом вызове он добавляет себя в списки из **LOADER_PARAMETER_BLOCK**: в **LoadOrderList** как копия первого модуля в списке (а это ядро ОС) и в **BootDriverList** как загрузочный драйвер, якобы прописанный в **\Registry\Machine\System\CurrentControlSet\Services\null**. Таким образом, вредоносная программа имитирует свою загрузку в качестве обычного boot-драйвера.

Второй раз драйвер вызывается операционной системой, которая уверена, что сама загрузила его. Данные манипуляции приводят к некоторым побочным эффектам.

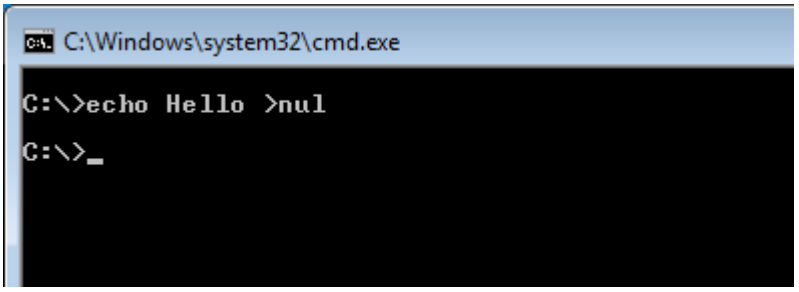
Например, в системе появляется драйвер Null, но при более внимательном рассмотрении оказывается, что он был создан ядром (ntoskrnl.exe).

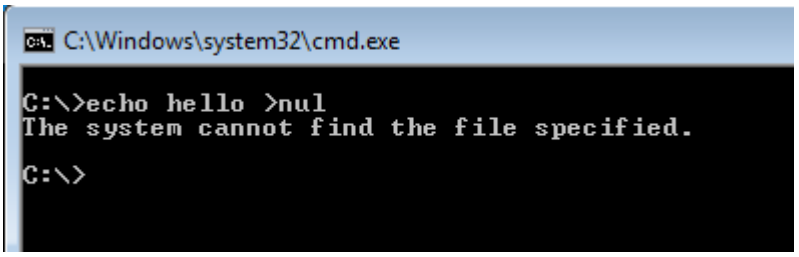
```
kd> !drvobj \driver\null
Driver object (81b7cf38) is for:
\Driver\Null
Driver Extension List: (id , addr)

Device Object list:
kd> dt nt!_DRIVER_OBJECT 81b7cf38
+0x000 Type : 0n4
+0x002 Size : 0n168
+0x004 DeviceObject : (null)
+0x008 Flags : 0x12
+0x00c DriverStart : 0x80800000 Void
+0x010 DriverSize : 0x5880
+0x014 DriverSection : 0x81bfc3a0 Void
+0x018 DriverExtension : 0x81b7cfe0 _DRIVER_EXTENSION
+0x01c DriverName : _UNICODE_STRING "\Driver\Null"
+0x024 HardwareDatabase : 0x80668fa90 _UNICODE_STRING "\REGISTRY\MACHINE\HARDWARE\DESCRIPTION\SYSTEM"
...
kd> dt nt!_LDR_DATA_TABLE_ENTRY 0x81bfc3a0
+0x000 InLoadOrderLinks : _LIST_ENTRY [ 0x81bfc338 - 0x8055b1c0 ]
+0x008 InMemoryOrderLinks : _LIST_ENTRY [ 0x0 - 0x0 ]
+0x010 InInitializationOrderLinks : _LIST_ENTRY [ 0x0 - 0x0 ]
+0x018 DllBase : 0x804d7000 Void
+0x01c EntryPoint : 0x806aeb2c Void
+0x020 SizeOfImage : 0x216680
+0x024 FullDllName : _UNICODE_STRING "\WINDOWS\system32\ntoskrnl.exe"
+0x02c BaseDllName : _UNICODE_STRING "ntoskrnl.exe"
...
```

В то же время среди загруженных модулей есть еще одно «ядро», в котором параметры **DllBase** и **SizeOfImage** принадлежат вредоносному драйверу.

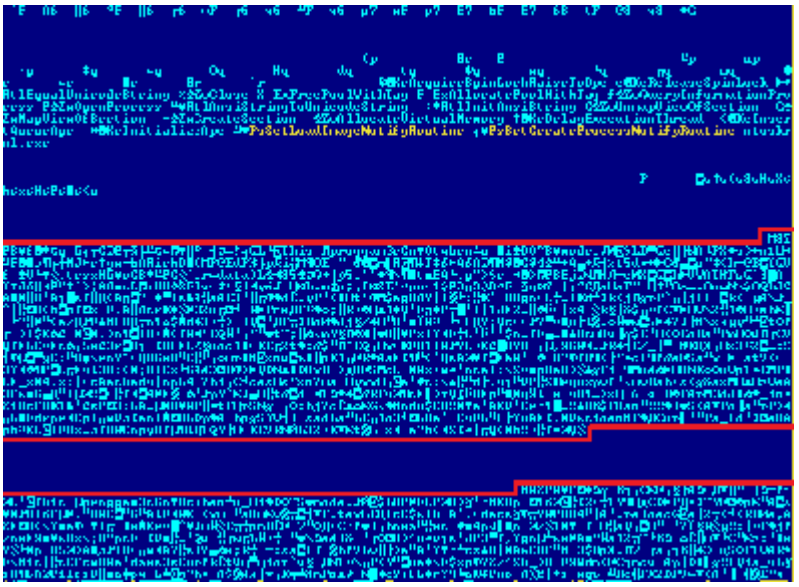
```
kd> dt nt!_LDR_DATA_TABLE_ENTRY 81bf1630
+0x000 InLoadOrderLinks : _LIST_ENTRY [ 0x81aa7db8 - 0x81bf16a0 ]
+0x008 InMemoryOrderLinks : _LIST_ENTRY [ 0x0 - 0x0 ]
+0x010 InInitializationOrderLinks : _LIST_ENTRY [ 0x0 - 0x0 ]
+0x018 DllBase : 0x80800000 Void
+0x01c EntryPoint : 0x80801110 Void
+0x020 SizeOfImage : 0x5880
+0x024 FullDllName : _UNICODE_STRING "\WINDOWS\system32\ntoskrnl.exe"
+0x02c BaseDllName : _UNICODE_STRING "ntoskrnl.exe"
+0x034 Flags : 0xc004000
```

В качестве экспресс-проверки системы на наличие или отсутствие заражения можно использовать простую команду «echo hello >nul», которая на неинфицированной системе успешно выполняется, а на зараженной выдает сообщение об ошибке.



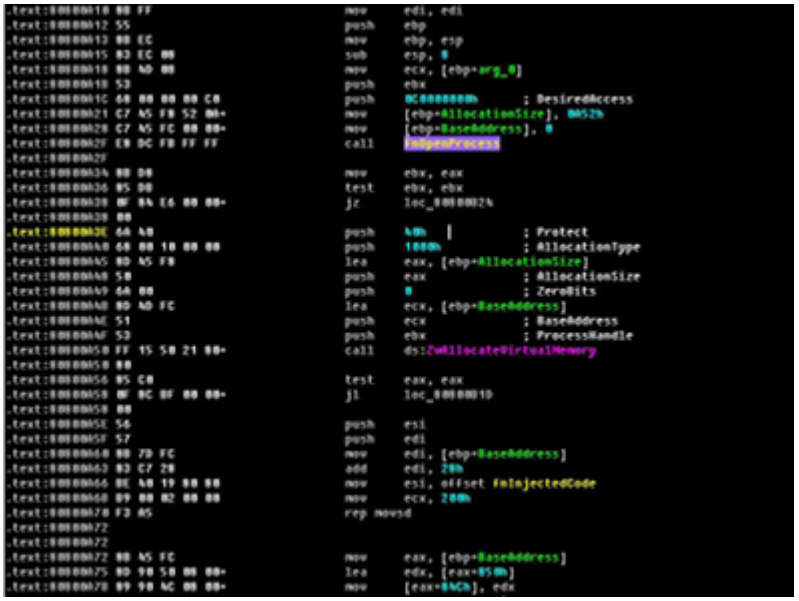


Задачей драйвера является инжект (внедерение) своего кода в запущенные процессы.



64-битный драйвер, красным выделены динамические библиотеки, которые упакованы aplib

Внедрение кода осуществляется обычной установкой нотификаций через функции PsCreateProcessNotifyRoutine и PsCreateProcessNotifyRoutine с последующим вызовом асинхронной функции через механизм APC. В процессе исследования выяснилось, что 64-битный драйвер несет «на борту» две библиотеки. При этом полезная нагрузка находится только в одной из них, а вторая, по всей видимости, является «заделом на будущее».



32-битный драйвер, который осуществляет заброс шелл-кода в процессы

В остальном же драйвер не представляет особого интереса. На сегодняшний день используемый **Trojan.Mayachok.2** механизм заражения является уникальным среди известных угроз. Специалисты компании «Доктор Веб» предполагают, что в недалеком будущем стоит ожидать использования подобной техники заражения другими вредоносными программами.
