Анализ руткита TDSS

Алиса Шевченко, eSage Lab
alisa@esagelab.ru

Вступление

Руткит TDSS также известен под именами Tidserv, TDSServ и Alureon. Отдельные его компоненты детектируются антивирусами под другими именами, такими как Trojan.Win32.DNSChanger и Trojan.FakeAlert.

Причины, по которым данный руткит удостоен детального исследования, таковы.

1. Судя по огромному количеству "воплей о помощи" от пользователей на публичных форумах, большинство антивирусов не справляются с лечением руткита, хотя успешно его обнаруживают. [1]
2. В публичном доступе нет полноценного описания функционала TDSS.
3. Реализация TDSS интересна тем, что, являясь со всей очевидностью эффективной, она не задействует никаких изощренных техник "нового поколения" или "0-day".
4. TDSS активно распространяется в живой среде, развиваясь в мощный ботнет. По данным Лаборатории Касперского за март-апрель 2009 г., в базы ежедневно добавляется от 100 до 300 сигнатур для новых и модифицированных файлов руткита. [2]

Таким образом, руткит TDSS - угроза пограничного типа: он достаточно мощный для поражения антивирусных программ, но недостаточно опасный для детального исследования; достаточно распространенный для публичных криков о помощи, но не дотягивает до эпидемии.

Обзор семейства

TDSS известен своей выдающейся способностью обходить активную защиту (антивирус, HIPS, фаервол), сложноудаляемостью и rootkit-функционалом. Его типичные наблюдаемые признаки в системе - всплывающие окна (pop-up), а также проблемы с запуском и обновлением антивирусных продуктов. На практике может иметь место любой функционал, что обеспечивается динамической подгрузкой дополнительных модулей.

Первые прецеденты заражения руткитом зафиксированы в середине 2008 года. Уже тогда TDSS эффективно обходил защиты в процессе инсталляции. Учитывая, что разработчикам руткита удается поддерживать работоспособность этой функции уже почти год, вопреки обновлениям антивирусов, а также учитывая грамотность реализации и архитектуры кода - TDSS создан и развивается силами команды профессиональных разработчиков и в рамках ясной стратегии.

Сам по себе TDSS - не более чем мощный загрузчик (downloader) с модульной архитектурой. Его основная задача - проникнуть на машину пользователя в обход защиты, хорошо закрепиться в системе, и в дальнейшем обеспечивать удаленное управление ресурсами зараженной машины, включая загрузку дополнительных функциональных модулей.

Схема распространения TDSS не менее продуманна, чем его архитектура. Используется несколько каналов доставки вредоносного кода. Известные вектора атаки включают загрузку кода через web-эксплойт (iframe attack)[3], инсталляцию его под видом видео-кодека [4] для просмотра порнографии, а также распространение в связке с легитимными [5] и пиратскими программами, генераторами ключей и crack'ами. [6]

Характерные особенности

- Настоящее название руткита TDSS - 'TDL'. Более поздние образцы руткита называют себя 'TDL2'.
- Файлы троянца защищены от анализа посредством несложной обфускации и шифрования.
- Некоторые файлы отмечены поддельным штампом версии, характерным для программ Microsoft.
- Основа механизма инсталляции TDSS - загрузка сервисом msiexec.exe (Microsoft Installer) своей собственной легитимной, но модифицированной DLL. [7]
- После инсталляции руткит блокирует загрузку или обновление антивирусных программ.
- Руткит тщательно закрепляется в системе, препятствуя деинсталляции. Например, некоторые модификации работают в режиме Safe Mode - это достигается модификацией ключей реестра HKLM\SYSTEM\ControlSet001\Control\SafeBoot\Minimal и HKLM\SYSTEM\ControlSet001\Control\SafeBoot\Network.
- Руткит хранит свои настройки в реестре: список антивирусных модулей, которые необходимо блокировать, список собственных модулей, которые необходимо подгружать в адресное пространство браузера, и т.д.
- Руткит скрывает свои файлы и ключи посредством перехвата нескольких функций.

- Драйвер руткита открывает своим модулям окно в пространство ядра. Для этой цели используется вызов функции ZwFlushInstructionCache со специфическим набором аргументов.

Внутривидовое разнообразие

Первые версии руткита в 2008 г. создавали в системе драйвер tdsserv.sys, от которого и произошло официальное имя TDSS. Впоследствии имя основного драйвера несколько раз менялось на clbdriver.sys, seneka*.sys, UACd*.sys, gaopdx*.sys, tdlserv.sys и другие.

Первые версии руткита использовали патчинг библиотеки advapi32.dll, подгружаемой сервисом Microsoft Installer, в ходе инсталляции. Более новые версии используют патчинг библиотеки msi.dll. Возможно, это изменение – реакция на новые поведенческие сигнатуры систем защиты.

Защита кода руткита от анализа выполнена в виде распаковщика, замаскированный под системный файл. При этом исполняемый файл визуально выглядит как большой кусок легитимного кода (отличающегося низкой энтропией в пределах байтового массива) с прицепленным в конце файла массивом данных (с высоким уровнем энтропии и без исполняемых инструкций). Исполняемый код украшен бессмысленными строками текста, призванными вызвать у поспешного аналитика ассоциации с системной утилитой. Не считая описанного «социального» трюка, защита кода не представляет интереса – это тривиально снимаемый «конверт», внутри которого спрятан оригинальный код троянца.

Новейшие образцы эволюции TDSS содержат функционал червя. В частности, руткит пытается – и, судя по отзывам жертв, вполне успешно – размножаться через съемные накопители. С этой целью копия руткита размещается в директории RECYCLER в виде скрытого файла с именем <многобукв>.com, и создается ссылка на него в файле autorun.inf. Это приводит к автоматической загрузке руткита при подключении зараженного накопителя к системе (за исключением случаев, когда функция Autorun для внешних дисков запрещена в настройках системы), а также при попытке открытия зараженного накопителя из «Проводника».

Анализ кода

Для анализа был использован экземпляр, обнаруженный примерно в Марте 2009 (MD5: 1DE66FC07C7B5893F5F83B397AC38F3D). [8] Разновидность TDSS, представленная этим экземпляром, отмечена Российским филиалом лаборатории Symantec как один из наиболее упоминаемых вредоносных программ в марте 2009 г. [9]

Общий алгоритм функционирования TDSS, характерный для семейства в целом, уже описан[10], также как и основные механизмы компонентов руткита в ring3. Суммарный высокоуровневый обзор функционала конкретного файла можно найти поиском по MD5 в архиве любой публичной песочницы (например, ThreatExpert). Поэтому я опишу только самые важные механизмы и функционал уровня ядра.

Инсталляция руткита и обход защиты

Процедура инсталляция руткита достойна упоминания, поскольку обеспечивает эффективный обход поведенческих защит и фаерволов. Ключевая идея алгоритма обхода – исполнение вредоносного кода в составе модифицированной системной библиотеки, автоматически подгружаемой легитимным системным сервисом. В результате такой манипуляции фаервол пропускает сетевую активность вредоносного кода в силу того, что его процесс-источник – системный сервис Windows – по умолчанию внесен в «белый список» и, как следствие, обладает всеми возможными привилегиями для загрузки и инсталляции файлов. Большинство проактивных защит также пропускают поведенческую активность руткита, поскольку она не вписывается в типичные паттерны нежелательного поведения.
Технически этот алгоритм реализован следующим образом. Модификация системной библиотеки – advapi32.dll или, в более поздних версиях руткита, msi.dll – осуществляется в копии библиотеки на диске, которая затем загружается в директорию \KnownDLLs в памяти. После этого стандартными средствами запускается Microsoft Installer, который автоматически подгружает модифицированную библиотеку из \KnownDLLs. [11]

```
// новая секция для кода инсталляции руткита
NtCreateSection(..”\knowndlls\dll.dll”..)
// подготовка системной DLL к патчингу
CopyFile(..”msi.dll”, ..)
// патчинг копии
WriteFile(.., ..)
NtOpenSection(..”\knowndlls\msi.dll”..)
// убираем флаг OBJ_PERMANENT, чтобы система
позволила
// удалить объект msi.dll и создать его за-
ново
NtMakeTemporaryObject(..)
CloseHandle(..)
// создаем новый маппинг для msi.dll, загру-
жая в него
// модифицированную библиотеку
NtCreateSection(..”\knowndlls\msi.dll”, ..)
..
// нормальный старт сервиса
StartService (..”Windows Installer”..)
```

В этой схеме модификация кода msi.dll минимальна, и обеспечивает только выполнение кода из секции \knowndlls\dll.dll, содержащей процедуры инсталляции руткита. Сам патч - <malicious_code_injection> - элегантен:

```
; адрес 7c906cbc - указатель на строку
'dll.dll'
; на самом деле, это часть строки имени ле-
гитимного модуля ntdll.dll
push 7c906cbc
; в стек помещается адрес следующей за call
инструкции
call $+5
; после выполнения следующей инструкции пер-
вый dword на стеке
; будет указывать на первую инструкцию пат-
ча. Таким образом,
; вызов LoadLibrary вернется туда, где к то-
му времени будет
; восстановлен оригинальный код
sub  dword ptr [esp], 0a
mov eax, LoadLibrary
jmp eax ; вызов LoadLibrary ('dll.dll')
```

Функционал секции dll.dll отображен на Рис.1.

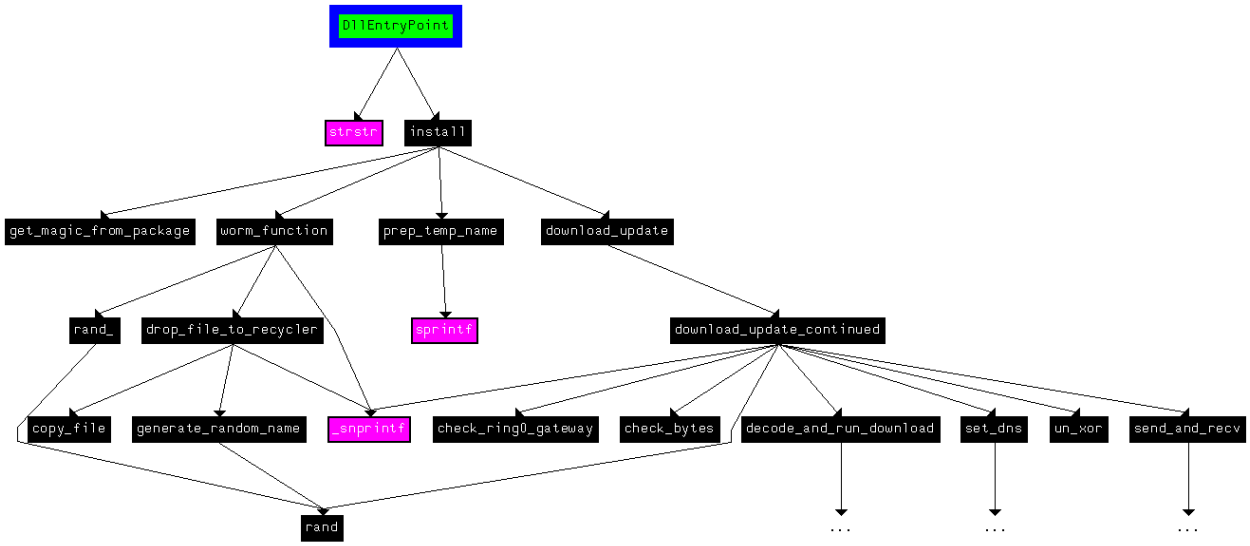

Рис. 1. Основная задача этой библиотеки – загрузка и инсталляция драйвера руткита.

Драйвер

TDSS не имеет собственного процесса в пользовательском пространстве. Базовые функции обеспечиваются драйвером, который автоматически загружается на старте. Дополнительные высокоуровневые функции обеспечиваются отдельными DLL-модулями, внедряемыми в заданные процессы.

Функционал драйвера включает в себя:

- сокрытие руткита
- обеспечение «портала» в режим ядра
- предотвращение загрузки антивирусов, перечисленных в конфигурационном ключе реестра
- внедрение DLL-модулей в процессы, перечисленные в конфигурационном ключе
- инсталляция новых DLL-модулей.

Rootkit-функционал

Руткит перехватывает методом сплайсинга следующие функции в пространстве ядра:

IofCallDriver
IofCompleteRequest
NtFlushInstructionCache
NtQueryVlaueKey
NtEnumerateKey

Перехват NtEnumerateKey используется для сокрытия ключей реестра, отвечающих за загрузку и конфигурацию руткита (в данном случае это все ключи, начинающиеся со строки "gaopdx*"). Перехват NtQueryValueKey используется для подмены адресов DNS-серверов (значения 'DhcpNameServer' и 'NameServer' [12]) в реальном времени, без модификации реестра. Перехват IofCallDriver обеспечивает глобальную фильтрацию всех IRP в системе, что позволяет руткиту прятать свои файлы (начинающиеся со строки "gaopdx*") всякий раз, когда он перехватывает IRP к драйверу файловой системы:

```
If ( FsRtlIsNameInExpression (.."*\\gaopdx*"
or "*\\TEMP\\gaopdx*"..) )
then return (STATUS_TOO_MANY_SECRETS)
```

Перехват IofCompleteRequest обеспечивает схожий функционал.

Портал в пространство ядра

Перехват NtFlushInstructionCache несколько более любопытен, поскольку он служит порталом в ring0 для пользовательских модулей руткита. Для того чтобы воспользоваться порталом, данная функция вызывается со специфическим набором аргументов, которые включают магическое значение, команду и параметр к ней.

```
; аргумент к команде
push 0
; команда (4 байта) – обеспечивает проверку
работоспособности портала
push 'VERG'
; магическое число, обеспечивающее выполне-
ние процедуры перехватчика,
; а не оригинальной API функции
push 'TDL2'
call ds:ZwFlushInstructionCache
```

Набор команд, обрабатываемых перехватчиком NtFlushInstructionCache, очень ограничен и не позволяет управлять руткитом. Доступные команды обеспечивают передачу переменных из ядра в пользовательский модуль, остановку заданного процесса или потока из ядра (посредством внедрения соответствующей задачи в очередь APC-вызовов ядра), и запуск процедур инсталляции нового dll-модуля.

Устойчивость в системе

Драйвер вызывает функцию ExQueueWorkItem для запуска несколько потоков в ядре. Потоки зациклены с периодом меньше секунды. Таким образом обеспечивается постоянная перерегистрация драйвера в системе (‘\registry\machine\system\currentcontrolset\services\gaopdxserv.sys’), отключение системного фаервола (‘\registry\machine\system\currentcontrolset\services\sharedaccess\parameters\firewallpolicy\’) и другие функции.

Блокирование антивирусов

Драйвер руткита посредством вызова API-функции PsSetLoadImageNotifyRoutine инсталлирует процедуру нотификации, получающую управление по факту загрузки любого модуля в память. Внутри процедуры происходит сверка имени загружаемого модуля со списком запрещенных, перечисленных в ключе ‘disallowed’ настроек руткита в реестре. Загрузка запрещенных модулей блокируется.

Удаление TDSS вручную

Перечисленные ниже инструкции составляют универсальный алгоритм удаления любого варианта руткита TDSS. Для воплощения алгоритма не требуется ни специальных знаний, ни специфических утилит.

1. В Device Manager (Компьютер → Управление устройствами), в разделе Non-PnP Driver, отключить и удалить драйвер руткита. Его можно локализовать по имени (tdsserv.sys, quadraserv.sys и т.д.). Поскольку полагаться на имя драйвера ненадежно – его лучше определить наверняка при помощи любого антируткита (GMER или Rootkit Unhooker предпочтительны; Avira Antirootkit также справляется с задачей). После этих манипуляций все файлы и ключи реестра руткита становятся видимыми и доступными для удаления вручную.
2. Удалить файл, соответствующий данному драйверу.
3. Удалить все ключи реестра, ссылающиеся на данный драйвер, и файлы дополнительных модулей, перечисленные в конфигурационном ключе.
4. Удалить (при их наличии) файлы autorun.inf и RECYCLER*.com на всех дисках.
5. Перезагрузиться.
6. Использовать антивирус для окончательной очистки системы от возможных пропущенных файлов.

Шаги 1-4 должны выполняться без помощи антивирусов, поскольку в случае отсутствия сигнатуры к какому-либо файлу дезинфекция не будет успешной.

Выводы и заключение

- Эффективность TDSS демонстрирует, что для обхода систем защиты нет необходимости изобретать нетривиальные решения.
- Разработчики вредоносных программ продолжают находить способы решения своих задач, основанные на обращении особенностей защиты в ее уязвимости. [13]
- Распространение вредоносной программы в связке с легитимной – очень эффективная техника, хотя и не новая. Идея этой техники заключается в том, что если пользователь добровольно запускает приложение, которое считает легитимным – то он добровольно устранит всевозможные предупреждения от поведенческой защиты или UAC’а. Он также пропустит инсталляцию драйвера, поскольку для некоторых приложений (например, инсталлятора кодеков[5]) использование драйвера вполне ожидаемо. Кроме того, наличие видимого окна инсталлятора может ввести в заблуждение простую поведенческую защиту.

Разработчикам поведенческих защит и систем HIPS имеет смысл следить за действиями в системе, обеспечивающими эффективность TDSS.

- Вызов функций NtOpenSection, NtMakeTemporaryObject и других API-интерфейсов к системным секциям.
- Получение доступа к системным библиотекам (даже если это простое копирование).
- Вызов LoadLibraryEx с параметром DONT_RESOLVE_DLL_REFERENCES (исполь-

зуется кодом dll.dll для восстановления кода msi.dll).
- Модификация системной конфигурации DNS и DHCP.
- Вызов PsSetLoadImageNotifyRoutine. Несмотря на высокую вероятность того, что к моменту вызова этой функции защита будет уже отключена – это не дает достаточных оснований для того, чтобы отказаться от контроля.

Очевидно, что перечисленные действия не могут расцениваться как вредоносные сами по себе. Тем не менее, они с существенной вероятностью могут составлять часть нежелательного поведения, и таким образом, должны входить в поведенческий паттерн детектирования в виде комбинаторной связки.

Ссылки

[1] Поиск Google, отзывы пользователей о TDSS на форумах
http://www.google.com/search?q=tdss+%7C+tidserv+%7C+tdsserv+daterange:01012009-26042009+inurl:forum

[2] Лаборатория Касперского, статистика по сигнатурам для TDSS
http://www.kaspersky.com/viruswatchlite?search_virus=TDSS

[3] Dancho Danchev, Embassy of India in Spain Serving Malware
http://ddanchev.blogspot.com/2009/01/embassy-of-india-in-spain-serving.html

Malware Analysis & Diagnostic, Etude de cas - Infection rootkit TDSS
http://mad.internetpol.fr/archives/3-Etude-de-cas-Infection-rootkit-TDSS.html

[4] ThreatExpert, отчет о действиях программы с MD5=2c5c874235a73fc50a69780c7ad1488a
http://www.threatexpert.com/report.aspx?md5=2c5c874235a73fc50a69780c7ad1488a

[5] ThreatExpert, отчет о действиях программы с MD5=d2ada2dba8e036d37726ebddbcc9e9d6
http://www.threatexpert.com/report.aspx?md5=d2ada2dba8e036d37726ebddbcc9e9d6

[6] ThreatExpert, отчет о действиях программы с MD5=b17d76537ef5d94547fc4ca8851b35da
http://www.threatexpert.com/report.aspx?md5=b17d76537ef5d94547fc4ca8851b35da

[7] Symantec, Backdoor.Tidserv Technical Details
http://www.symantec.com/security_response/writeup.jsp?docid=2008-091809-0911-99&tabid=2

[8] Virustotal.com, отчет о сканировании программы с MD5=1de66fc07c7b5893f5f83b397ac38f3d
http://www.virustotal.com/analisis/122e4ade1c0fa88cbab02880a3b2ed98

[9] Anti-malware.ru, История информационной безопасности за 4-ю неделю марта от Symantec
http://www.anti-malware.ru/node/1250

[10] F-Secure, Backdoor:W32/TDSS Virus Description
http://www.f-secure.com/v-descs/backdoor_w32_tdss.shtml

[11] Microsoft Support, INFO: Windows NT/2000/XP Uses KnownDLLs Registry Entry to Find DLLs
http://support.microsoft.com/kb/164501

[12] MSDN, DhcpNameServer registry key
http://technet.microsoft.com/en-us/library/cc962470.aspx

[13] Virus Bulletin January 2009. Shevchenko A., Advanced malware techniques 2008
http://esagelab.ru/files/AlisaShevchenko-Jan09.pdf

Приложения

Экземпляр руткита TDSS и файлы анализа IDA (по запросу)

Утилита для удаления руткита TDSS
http://www.esagelab.com/files/tdss_remover_latest.rar