Статья Малварь, способная сидеть в сети компании годами. PlugX, nccTrojan, dnsTrojan, dloTrojan

Недавно мы расследовали APT-атаку на одну российскую компанию и нашли много занятного софта. Сначала мы обнаружили продвинутый бэкдор PlugX, популярный у китайских группировок, APT-атаки которых обычно нацелены на похищение конфиденциальной информации, а не денег. Затем из скомпрометированной сети удалось вытащить несколько других схожих между собой бэкдоров (nccTrojan, dnsTrojan, dloTrojan) и даже общедоступных утилит.

Программы, используемые в этой преступной кампании, не отличаются сложностью, за исключением, может быть, PlugX. К тому же три из четырех вредоносов использовали при запуске давно известную технику DLL hijacking. Тем не менее, как показало наше исследование, даже при таких условиях злоумышленники могут годами оставаться в скомпрометированных сетях.

Мы решили изучить обнаруженный софт и поделиться своими наблюдениями.

PlugX

PlugX — сложная вредоносная программа. Мы постараемся рассказать о ее основных функциях, а более подробное описание малвари можно найти в отчете Dr. Web.

Запуск PlugX

PlugX, как правило, распространяется в виде самораспаковывающихся архивов, содержащих:

- невредоносный исполняемый файл (EXE), подписанный цифровой подписью (нам попадались подписи McAfee, Kaspersky, Support.com);
- вредоносную динамическую библиотеку (Dynamic Link Library — DLL);
- зашифрованную основную нагрузку — файлы с произвольными именем и расширением.

Такой набор характерен для техники DLL hijacking, при которой злоумышленник заменяет легитимную DLL на вредоносную. При этом малварь получает возможность работать от имени легитимного процесса и обходить таким образом средства защиты (рис. 1).

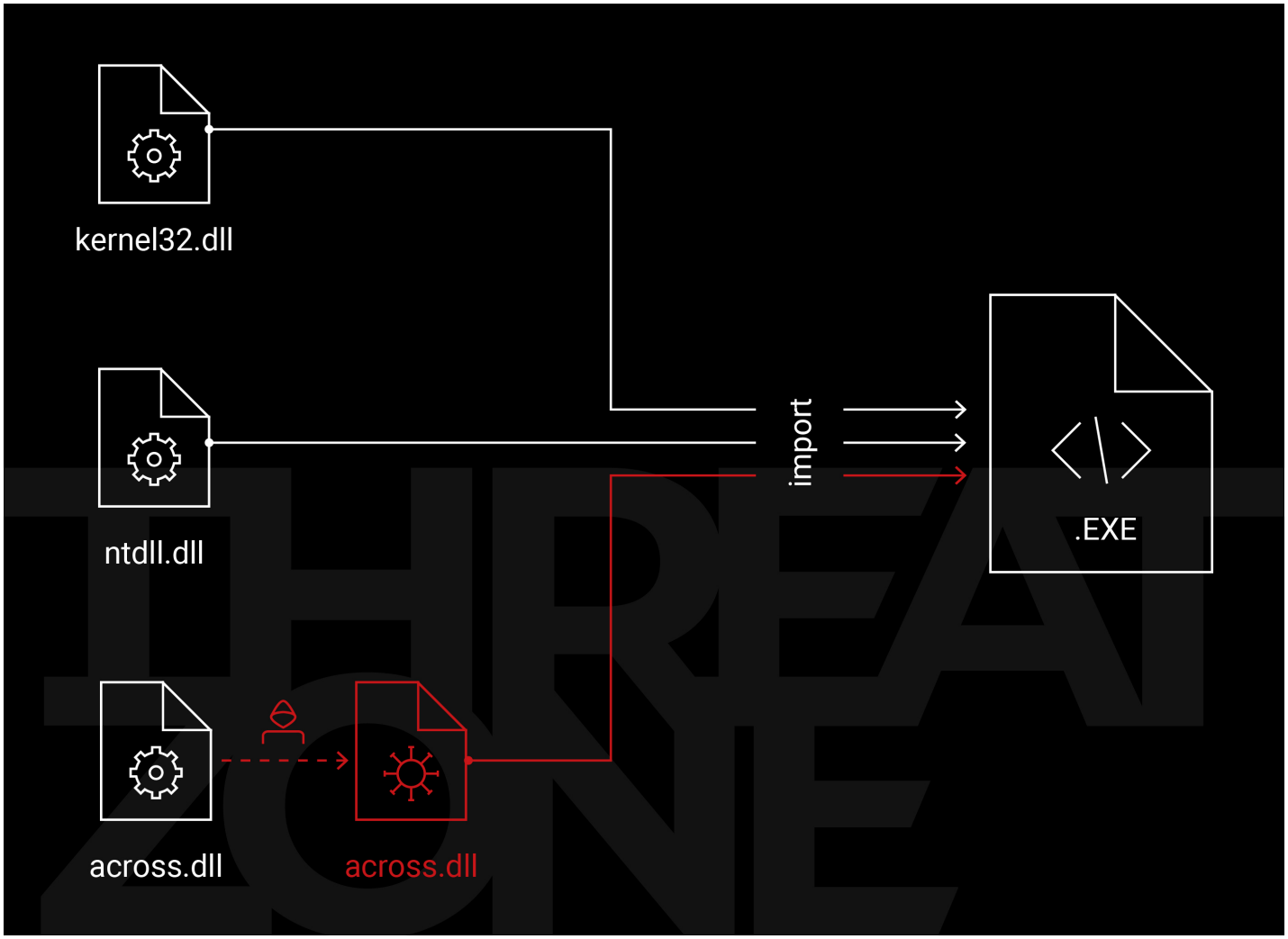

Рис. 1. Наглядное представление техники DLL hijacking

Рассмотрим в качестве примера один из экземпляров PlugX, характеристики которого приведены в табл. 1.

Табл. 1. Свойства файлов одного из образцов PlugX

Свойство	EXE	DLL
Имя файла	mcut.exe	mcutil.dll
Тип файла	PE32 executable (EXE)	PE32 executable (DLL)
Размер (в байтах)	140 576	4 096
Время компиляции	13 июня 2008 года 02:39:28	9 декабря 2014 года 10:06:14
MD5	884d46c01c762ad6ddd2759fd921bf71	e9a1482a159d32ae57b3a9548fe8edec
SHA-1	d201b130232e0ea411daa23c1ba2892fe6468712	a2a6f813e2276c8a789200c0e9a8c71c57a5f2d6
SHA-256	3124fcb79da0bdf9d0d1995e37b06f7929d83c1c4b60e38c104743be71170efe	2f81cf43ef02a4170683307f99159c8e2e4014eded6aa5fc4ee8207

Вот что происходит при запуске невредоносного исполняемого файла (EXE) из пакета.

Сначала одна из импортируемых им библиотек (отдельная DLL) заменяется вредоносной. После загрузки в память процесса DLL открывает третий файл из пакета PlugX, который обходит средства защиты за счет отсутствия видимого исполняемого кода. Тем не менее он содержит шелл-код, после исполнения которого в памяти расшифровывается еще один дополнительный шелл-код. Он с помощью функции `RtlDecompressBuffer()` распаковывает PlugX (DLL). При открытии мы видим, что сигнатуры MZ и PE в исполняемом файле PlugX заменены на XV (рис. 2) — скорее всего, это тоже нужно, чтобы скрыть модуль от средств защиты.

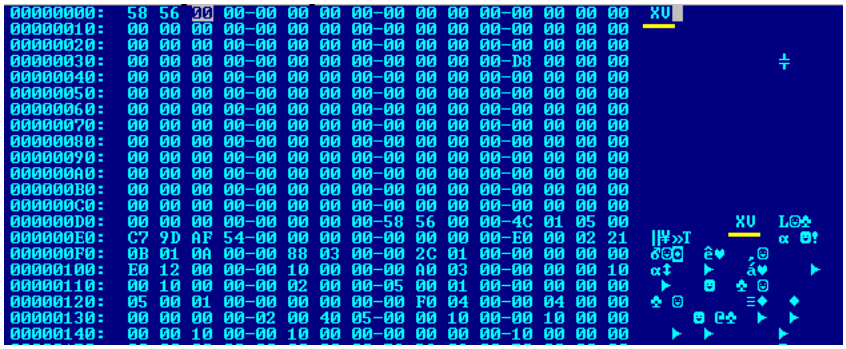

Рис. 2. Исполняемый файл PlugX в распакованном виде с измененными сигнатурами MZ и PE

Наконец, запускается распакованная вредоносная библиотека, и управление передается ей.

В другом экземпляре PlugX мы обнаружили интересную особенность: малварь пыталась скрыть некоторые библиотечные вызовы от песочниц. При восстановлении импортов вместо адреса импортируемой функции сохранялся адрес тремя байтами ранее. Результат для функции `SetFileAttributesW()` виден на рис. 3.

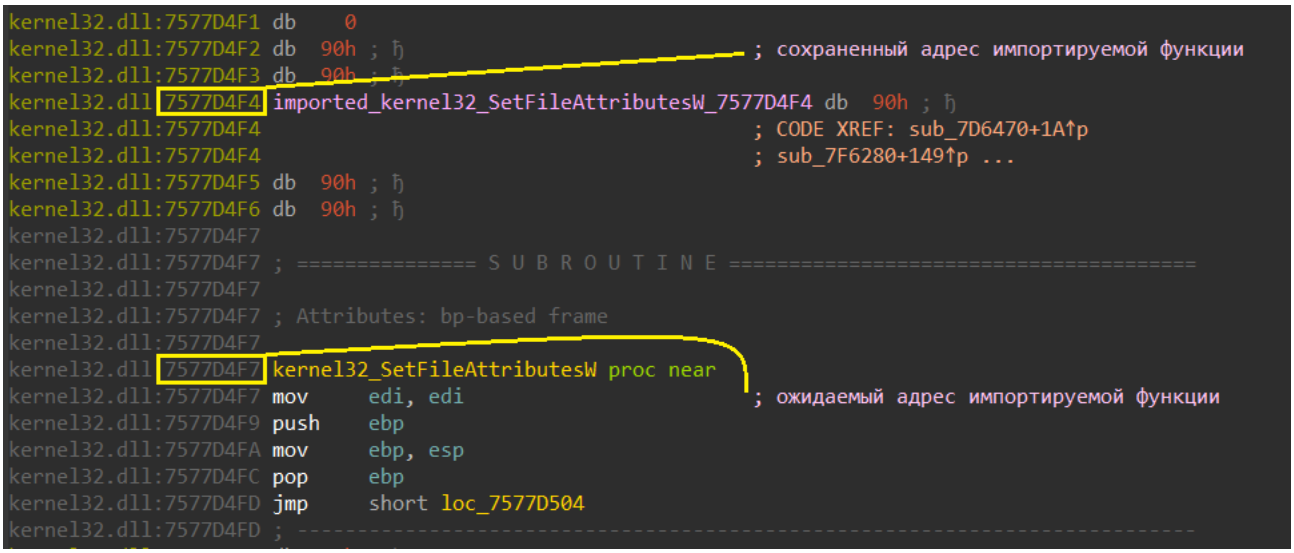

Рис. 3. При получении адреса функции SetFileAttributesW() сохраняется адрес 0x7577D4F4

В табл. 2 приведены характеристики этого экземпляра.

Табл. 2. Свойства файлов образца вредоносной программы PlugX, который пытался скрыть вызовы от песочниц

Свойство	EXE	DLL
Имя файла	mcut.exe	mcutil.dll
Тип файла	PE32 executable (EXE)	PE32 executable (DLL)
Размер	140 576	4 096
MD5	884d46c01c762ad6ddd2759fd921bf71	12ee1f96fb17e25e2305bd6a1ddc2de9
SHA-1	d201b130232e0ea411daa23c1ba2892fe6468712	bf25f1585d521bfba0c42992a6df5ac48285d763

SHA-256	3124fcb79da0bdf9d0d1995e37b06f7929d83c1c4b60e38c104743be71170efe	97ad6e95e219c22d71129285299c4717358844b90860bb7ab16c51

На рис. 4 представлен фрагмент кода PlugX, где встречается функция SetFileAttributesW(), и листинг из одной из песочниц.

```
SetFileTime(file_handle, &lpCreationTime, &lpLastAccessTime, &lpLastWriteTime);
CloseHandle = CloseHandle_747253F8;
if ( !CloseHandle_747253F8 )
{
  v24 = kernel32_100453E4;
  if ( !kernel32_100453E4 )
  {
    v24 = LoadLibraryA("kernel32");
    kernel32_100453E4 = v24;
  }
  CloseHandle = GetProcAddress(v24, "CloseHandle");
  CloseHandle_747253F8 = CloseHandle;
}
CloseHandle(file_handle);
SetFileAttributesW(lpFileName, 6u);
```

а)

```
[0094.541] SetFileTime (hFile=0xd8, lpCreationTime=0x203fbec, lpLastAccessTime=0x203fbf4, lpLastWriteTime=0x203fbfc) returned 1
[0094.541] CloseHandle (hObject=0xd8) returned 1
[0094.541] lstrlenA (lpString="mcutil.dll") returned 10
[0094.541] lstrlenA (lpString="mcutil.dll") returned 10
[0094.542] MultiByteToWideChar (in: CodePage=0x0, dwFlags=0x0, lpMultiByteStr=0x4eef08, cbMultiByte=10, lpWideCharStr=0x0, cchWideChar=0 |
out: lpWideCharStr=0x0) returned 10
```

б)

Рис. 4. Фрагмент декомпилированного кода (а) и соответствующий ему фрагмент листинга перехваченных инструкций (б), где встречается вызов функции SetFileAttributesW()

В листинге из песочницы мы не видим вызов функции `SetFileAttributesW()`. Впрочем, вызовы функций `LoadLibrary()` и `GetProcAddress()`, которые были импортированы аналогичным образом, мы тоже не видим.

Основная нагрузка PlugX не сохраняется в расшифрованном виде на диске.

Работа PlugX

После запуска вредоносная программа расшифровывает конфигурацию, которая содержит адреса серверов управления, а также информацию, необходимую для дальнейшего функционирования (например, способ закрепления в системе или путь, по которому копируются файлы малвари).

При этом данные для конфигурации могут браться из основного загрузчика или из отдельного файла в текущей рабочей директории. Из того же файла может быть подтянута новая конфигурация при ее обновлении в ходе взаимодействия с сервером управления.

То, как вредонос будет вести себя дальше, во многом определяет его конфигурация.

В зависимости от значения `check_flag` в конфигурации PlugX вредоносная программа может начать поиск в зараженной системе сетевого адаптера, MAC-адрес которого совпадает с адресом, заданным в самой малвари. В случае совпадения вредоносная программа завершит свое исполнение. Вероятно, таким образом она пытается обнаружить виртуальную среду.

Режим работы вредоносной программы зависит от значения флага конфигурации mode_flag, которое может быть равно 0, 2, 3 или 4.

Если значение `mode_flag` равно 0, вредоносная программа закрепляется в системе (подробнее в разделе «Закрепление в системе»). Затем она переходит к инициализации плагинов и взаимодействию с сервером управления (подробнее в разделе «Функциональность плагинов и исполнение команд»).

Если значение `mode_flag` равно 2, вредоносная программа сразу переходит к инициализации плагинов и взаимодействию с сервером управления.

Если значение `mode_flag` равно 3, вредоносная программа внедряет шелл-код в Internet Explorer. Передача управления вредоносному коду осуществляется с помощью функции `CreateRemoteThread()`. Также производится инициализация плагинов, и создается именованный пайп, через который вредоносная программа получает команды, предназначенные для исполнения плагинами.

Если значение `mode_flag` равно 4, устанавливается перехват некоторых функций библиотеки `wininet.dll`. В функции-перехватчике извлекаются логины и пароли для подключения к прокси-серверу. Полученные данные будут сохранены в файле в рабочей директории вредоносной программы. При этом имена файлов, создаваемых малварью в процессе функционирования, генерируются на основе серийного номера диска и некоторого уникального для каждого файла значения. Перехватываются следующие функции:

- `HttpSendRequestA()`,
- `HttpSendRequestW()`,
- `HttpSendRequestExA()`,
- `HttpSendRequestExW()`.

Закрепление в системе

Если конфигурация PlugX предусматривает закрепление вредоноса в зараженной системе, то в ней прописан каталог, в который будут скопированы компоненты малвари.

Анализируемый образец выбирает одну из следующих директорий в зависимости от разрядности малвари:

- `%SystemRoot%\System32\winssxs`
- `%SystemRoot%\SysWOW64\winssxs`

Для созданной директории и скопированных в нее компонентов программа пытается установить временные метки, совпадающие с временными метками библиотеки `ntdll.dll` из зараженной системы. Возможно, это нужно для маскировки под компонент ОС. Также PlugX скрывает эти файлы от пользователя, выставляя атрибуты `FILE_ATTRIBUTE_SYSTEM` и `FILE_ATTRIBUTE_HIDDEN`. В конфигурации указывается путь к месту, где будут сохраняться скриншоты, сделанные вредоносной программой.

В зависимости от `persistence_flag` PlugX может закрепляться:

- как сервис (`persistence_flag=1`),
- через реестр (`persistence_flag=2`),
- любым из двух способов (`persistence_flag=0`), сначала попытаться создать сервис, а в случае неудачи — закрепиться через реестр.

Помним, что малварь может и не закрепляться вовсе.

Когда вредонос пытается закрепиться как сервис, он маскируется под легитимную программу. Имя службы и ее параметры задаются в конфигурации. Например, в нашем образце создается сервис с именем SSXSS (отображаемое имя — SSXS) и описанием McAfee OEM Info Copy Files, а качестве исполняемого файла службы используется невредоносный mcut.exe.

Если закрепиться в качестве службы не удается, программа пытается закрепиться через реестр и перезапускает себя из каталога, в который была скопирована ранее. Для этого она использует ключ реестра `HKCU\Software\Microsoft\Windows\CurrentVersion\Run`. Ключ реестра и параметр задаются через конфигурацию: в анализируемом образце в параметр, соответствующий отображаемому имени сервиса, программа прописывает путь до того же `mcut.exe`.

После того, как вредоносная программа закрепилась и перезапустилась, производится внедрение вредоносного кода. В анализируемом образце он работает в памяти легитимного процесса `svchost.exe`. Имя процесса подтягивается из конфигурации. Если внедрение кода прошло успешно, текущий процесс завершается.

В зависимости от конфигурации вредоносная программа может также попытаться создать процесс с повышенными привилегиями с последующим внедрением в него кода. В конфигурации могут быть перечислены до четырех целевых процессов.

Функциональность плагинов PlugX и исполняемые команды

Основная функциональность бэкдора реализована с помощью так называемых плагинов. Фрагмент функции, в которой производится инициализация плагинов, приведен на рис. 5.

Рис. 5. Фрагмент инициализации плагинов PlugX

PlugX может управлять процессами и службами, работать с файловой системой, вносить изменения в реестр. Он также имеет компоненты кейлоггера и скринлоггера и может получать удаленный доступ к зараженной системе — все это дает обширные возможности злоумышленникам в скомпрометированной сети.

Полный перечень функций вредоносной программы, доступной через плагины, приведен в табл. 3.

Табл. 3. Функциональность PlugX, доступная через плагины

Фрагмент функции обработки команд, полученных от сервера управления приведена на рис. 6.

```
_DISK_746ECF40(plugins);
(plugins->DISK1.init_DISK1)(&init_funcs);
_DISK_746ED020(&plugins->DISK2);
(plugins->DISK2.init_DISK2)(&init_funcs);
_KeyLogger_746ED100(&plugins->keylogger);
(plugins->keylogger.start_keylogger)(&init_funcs);
_Nethood_746ED1E0(&plugins->nethood);
(plugins->nethood.init_Nethood)(&init_funcs);
_Netstat_746ED2C0(&plugins->netstat);
(plugins->netstat.init_Netstat)(&init_funcs);
_Option_746ED3A0(&plugins->option);
(plugins->option.init_Option)(&init_funcs);
_PortMap_746ED480(&plugins->portmap);
(plugins->portmap.init_PortMap)(&init_funcs);
_Process_746ED560(&plugins->processes);
(plugins->processes.init_Process)(&init_funcs);
_Regedit_746ED640(&plugins->regedit);
(plugins->regedit.init_Regedit)(&init_funcs);
_Screen_746ED720(&plugins->screen);
(plugins->screen.start_screenlogger)(&init_funcs);
_Service_746ED800(&plugins->service);
(plugins->service.init_Service)(&init_funcs);
_Shell_746ED8E0(&plugins->shell);
(plugins->shell.init_Shell)(&init_funcs);
_SQL_746ED9C0(&plugins->sql);
(plugins->sql.init_SQL)(&init_funcs);
_Telnet_746EDAA0(&plugins->telnet);
(plugins->telnet.init_Telnet)(&init_funcs);
```

```
switch ( packet->command_id )
{
  case 1:
    result = send_system_info_to_cnc_746EBCD0(packet, sync, st);
    goto LABEL_16;
  case 2:
  case 9:
  case 0xA:
    return 0;
  case 3:
    result = plugins_manager_746EC2B0(packet, st, sync);
    goto LABEL_16;
  case 4:
    return 10054;
  case 5:
    result = uninstall_PlugX_746EC390(packet, st);
    goto LABEL_16;
  case 6:
    result = send_current_configuration_to_cnc_746EC570(packet, st);
    goto LABEL_16;
  case 7:
    result = receive_new_configuration_from_cnc_746EC610(packet);
    goto LABEL_16;
  case 8:
    result = send_processes_data_746EC700(packet, st);
    goto LABEL_16;
  default:
    global_plugins = global_mlw_struct_747253D8;
    if ( !global_mlw_struct_747253D8 )
    {
      struc = wrap_LocalAlloc_100325B7(0x45020u);
      if ( struc )
        global_plugins = critical_section_45020b_746EDB80(struc);
      else
        global_plugins = 0;
      global_mlw_struct_747253D8 = global_plugins;
    }
    result = execute_plugin_746EDFC0(global_plugins, packet, st, packet);
LABEL_16:
    if ( result == -1 )
    {
      packet->error_or_success_code = 0x3EB;
      packet->UncompressedBufferSize = 0;
      result = encrypt_and_WSASend_7470A0F0(packet, st, -1);
    }
    return result;
}
```

Рис. 6. Команды сервера управления, которые получает PlugX

Описание команд приведено в табл. 4.

nccTrojan

Один из обнаруженных нами бэкдоров найден в отчете VIRUS BULLETIN и назван авторами nccTrojan по константному значению в коде основного пейлоада. Характеристики попавшегося нам образца малвари приведены в табл. 5.

Табл. 5. Свойства анализируемого образца nccTrojan

Свойство	EXE	DLL
Имя файла	instsrv.exe	windowsreskits.dll
Тип файла	PE32 executable (EXE)	PE32 executable (DLL)
Размер (в байтах)	83 968	514 048

Время компиляции	18 декабря 2019 года 03:13:03	21 марта 2020 года 15:19:04
MD5	c999b26e4e3f15f94771326159c9b8f9	056078b1c424667e6a67f9867627f621
SHA-1	ec12c469463029861bd710aec3cb4a2c01907ad2	5bd080285a09c0abf742fb50957831310d9d9769
SHA-256	07d728aa996d48415f64bac640f330a28e551cd565f1c5249195477ccf7ecfc5	3be516735bafbb02ba71d56d35aee8ce2ef403d08a4dc47b46d5be9

Запуск nccTrojan

Вредоносная программа состоит из двух файлов: `EXE` и `DLL`, но в данном случае техника DLL hijacking не используется. После запуска вредоносный EXE-файл (MD5: c999b26e4e3f15f94771326159c9b8f9) регистрирует библиотеку `windowsreskits.dll` (MD5: 056078b1c424667e6a67f9867627f621) в качестве сервиса. В зависимости от разрядности библиотеки ее файл копируется в одну из директорий:

- `%SystemRoot%\System32`
- `%SystemRoot%\SysWOW64`

В результате создается сервис с именем `WindowsResKits`, работающий в контексте процесса `svchost.exe`.

Работа nccTrojan

nccTrojan расшифровывает конфигурацию, хранящуюся по определенному смещению в оверлее. Конфигурация зашифрована с помощью алгоритма AES-CFB-256, он же используется для шифрования взаимодействия с сервером управления. Пары «ключ шифрования + вектор инициализации» захардкоржены и различны для шифрования конфигурации и взаимодействия с сервером управления.

Расшифрованная конфигурация содержит информацию о сервере управления и выглядит следующим образом:
Code:

```
***news.niiriip.com|#|none|#|none|#|443|#|none|#|none|#|passwd|#|ncc|#|v2.2[Service]***
```

Строка `v2_2[Service]`, вероятнее всего, характеризует версию вредоносной программы и также используется в качестве имени при создании мьютекса.

В зависимости от конфигурации nccTrojan может иметь до трех серверов управления. В исследуемом экземпляре вредоносной программы задан лишь один сервер управления — `news.niiriip[.]com`, а поля конфигурации под оставшиеся два заполнены `none`.

Дальнейшее взаимодействие осуществляется только в случае получения с сервера управления пакета, который после расшифровывания содержит строку `ncc`.

Если соединение установлено, то на сервер управления отправляется следующая информация:

- имя компьютера,
- контрольная сумма от отправляемых данных,
- имя пользователя,
- уровень привилегий пользователя в системе (SYSTEM, Administrator или User),
- IP-адреса зараженной системы,
- идентификатор версии операционной системы,
- язык системы.

При этом из собранных данных формируется строка, которая дальше зашифровывается и отправляется на сервер управления. Формат создаваемой строки:
Code:

```
passwd|#|<check_sum>|#|<computer_name>|#|<user_name>|#|<user_privilege>|#|<ip_addr_1 / ip_addr_2 / ...>|#|None|#|v2.2[Service]|#|<os_version_ID>|#|
<language_ID>
```

Далее вредоносная программа переходит к взаимодействию с сервером управления и может исполнять команды, приведенные в табл. 6.

dnsTrojan

Следующий бэкдор мы обнаружили впервые: на момент расследования мы не нашли упоминаний о нем в отчетах других экспертов. Его отличительная особенность — общение с сервером управления через DNS. В остальном по своей функциональности вредоносная программа схожа с бэкдором nccTrojan. Чтобы сохранить единообразие в названиях найденной малвари, назвали ее dnsTrojan.

В рамках инцидента обнаружен CAB-архив, содержащий вредоносный исполняемый файл (MD5: a3e41b04ed57201a3349fd42d0ed3253). Характеристики образца, который мы вытащила в ходе расследования, приведены в табл. 7.

Табл. 7. Свойства анализируемого образца dnsTrojan

Свойство	EXE
Имя	a.exe.ok
Тип файла	PE32 executable (EXE)
Размер (в байтах)	417 280
Время компиляции	13 октября 2020 года 20:05:59
MD5	a3e41b04ed57201a3349fd42d0ed3253
SHA-1	172d9317ca89d6d21f0094474a822720920eac02
SHA-256	826df8013af53312e961838d8d92ba24de19f094f61bc452cd6ccb9b270edae5

Запуск dnsTrojan

В файле захардкоржена строка `AB0d3d3MS5kb3RvbWF0ZXIuY2x1Yjsw`, которая является некоей конфигурацией. Вредоносная программа проверяет внутри этой строки третий байт и в зависимости от его значения выбирает свою рабочую директорию и некоторую функциональность:

- `b"30" ('0')` — вредоносная программа работает в своей текущей директории, в реестр не прописывается, по завершении исполнения все созданные в процессе функционирования файлы: `LiveUpdate.exe`, `ccL100U.dll` и сам исполняемый файл — удаляются (конфигурация анализируемого образца);
- `b"31" ('1')` — вредоносная программа работает в `%AppData%\Sep`, прописывается в реестре, по завершении исполнения удаляется только сам исполняемый файл.

Подстрока `d3d3MS5kb3RvbWF0ZXIuY2x1Yjsw` представляет собой закодированную в base64 конфигурацию сервера управления `www1.dotomater.club;0`.

После запуска вредоносная программа извлекает из ресурсов, распаковывает и сохраняет в рабочей директории два файла:

- `LiveUpdate.exe` (MD5: 1a7f595ba8c2897461958204odcad404),
- `ccL100U.dll` (MD5: 0697433432d209c1ed95f6f75a111234).

Далее запускается файл LiveUpdate.exe, имеющий цифровую подпись Symantec Corporation.

В этом случае злоумышленники снова используют технику DLL hijacking: легитимная DLL заменяется на вредоносную `ccL100U.dll`. В результате исполнения кода библиотеки из ее ресурсов извлекается и распаковывается код самого бэкдора, который внедряется и исполняется в памяти легитимного процесса `dllhost.exe`. Для внедрения кода применяется техника Process Hollowing.

Если вредоносная программа прописывается в реестр, в ключ реестра `HKCU\Environment` добавляется параметр `UserInitMprLogonScript` со значением `%AppData%\Roaming\Sep\LiveUpdate.exe`.

Работа dnsTrojan

Все свои действия вредоносная программа логирует в файл `%ProgramData%\logD.dat`, при этом записанные данные похожи на отладочную информацию для злоумышленников (рис. 7).

```
[2020/11/30 5:28:10 3020 i] i am start

[2020/11/30 5:28:30 3020 i] InitProcessJob ok

[2020/11/30 5:28:30 3020 i] get config domain:www1.dotomater.club dns:0

[2020/11/30 5:28:30 3020 i] Host Name: 8SdXcAXRZDJ


[2020/11/30 5:28:30 3020 i] Domain Name:


[2020/11/30 5:28:30 3020 i] DNS Servers: 192.168.0.1


[2020/11/30 5:28:30 3020 i] get local dns addr:192.168.0.1

[2020/11/30 5:28:30 3020 i]

------------------------------MainWork---------------------



[2020/11/30 5:28:30 3020 i] DnsClient construct 0x5a7e44

[2020/11/30 5:28:31 3020 i] Mac: 00-13-d2-e3-d6-2e nCount = 0



[2020/11/30 5:28:31 3020 i] szSendMsg: 8SDXCAXRZDJ;O0V2m0SImxhY;6.1.1;1;00-13-d2-e3-d6-2e;2020113052831619

[2020/11/30 5:28:31 3020 i] start recv dns pack

[2020/11/30 5:28:21 2480 i] Queue pop ok.

[2020/11/30 5:28:21 2480 i] start SendUntilAck

[2020/11/30 5:28:21 2480 i] start Send cmd:0x1 size:67

[2020/11/30 5:28:21 2480 i] >>>>>>>>>>>>>send crypted payload len:108

[2020/11/30 5:28:21 2480 i] set checkSeqNum:67

[2020/11/30 5:28:32 3020 i] recved dns pack...
```

Рис. 7. Фрагмент файла logD.dat

Закодированная в base64 конфигурация содержит адреса сервера управления и DNS-сервера. В текущем семпле раскодированная конфигурация выглядит так: `www1.dotomater.club;0` , то есть адрес конкретного DNS-сервера злоумышленников отсутствует, а для взаимодействия с сервером управления используется DNS-сервер зараженной системы.

Взаимодействие с сервером управления осуществляется с использованием DNS-туннелирования. Данные передаются серверу управления в виде DNS-запроса TXT-записи в зашифрованном виде.

Сразу после запуска на сервер управления отправляются следующие данные:

- имя компьютера
- имя пользователя,
- текущие дата и время,
- версия операционной системы,
- информация о сетевом адаптере,
- архитектура процессора.

Из них формируется сообщение вида `8SDXCAXRZDJ;O0V2m0SImxhY;6.1.1;1;00-13-d2-e3-d6-2e;2020113052831619` .

Все передаваемые на сервер управления данные преобразуются следующим образом:

- шифруются с помощью алгоритма `AES-128-CBC` , ключ шифрования вырабатывается из константной строки `dadadadadadadada` с помощью функции `CryptDeriveKey()` ;
- кодируются в base64.

При формировании домена, для которого запрашивается TXT-запись, после каждого 64-го символа ставится точка. Запросы, отправляемые вредоносной программой, можно увидеть на рис. 8.

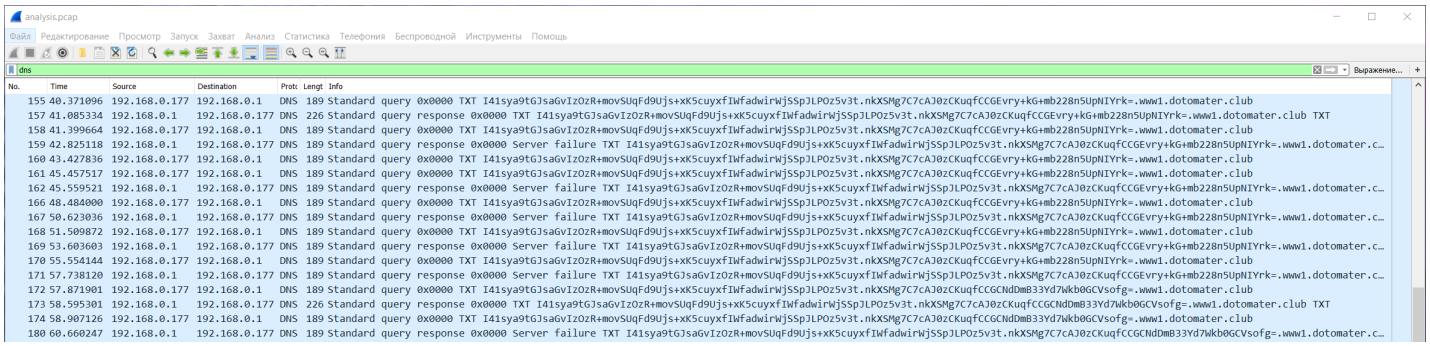

Рис. 8. Пример трафика вредоносной программы dnsTrojan

В ответ на запрос, отправленный на предыдущем шаге из TXT-записей, dnsTrojan получает команды сервера и может исполнить их (табл. 8).

dloTrojan

dloTrojan — еще одна обнаруженная в процессе расследования вредоносная программа, которую мы классифицировали как бэкдор. Эта малварь не относится ни к одному из известных семейств вредоносов.

В процессе исполнения кода расшифровывается строка `dlo`, поэтому по аналогии с предыдущими двумя программами мы назвали ее dloTrojan.

Характеристики файлов исследуемого нами образца приведены в табл. 9.

Табл. 9. Свойства анализируемого экземпляра dloTrojan

Свойство	EXE	DLL
Имя	ChromeFrameHelperSrv.exe	chrome_frame_helper.dll
Тип файла	PE32 executable (EXE)	PE32 executable (DLL)
Размер (в байтах)	82 896	240 128
Время компиляции	12 июля 2013 года 19:11:41	14 сентября 2020 года 16:34:44
MD5	55a365b1b7c50887e1cb99010d7c140a	bd23a69c2afe591ae93d56166d5985e1
SHA-1	6319b1c831d791f49d351bccb9e2ca559749293c	3439cf6f9c451ee89d72d6871f54c06cb0e0f1d2
SHA-256	be174d2499f30c14fd488e87e9d7d27e0035700cb2ba4b9f46c409318a19fd97	f0c07f742282dbd35519f7531259b1a36c86313e0a5a2cb5fe1dadc

Запуск dloTrojan

На сцену опять выходит DLL hijacking.

Итак, вредоносная программа dloTrojan состоит из двух компонентов:

- `ChromeFrameHelperSrv.exe` — невредоносный исполняемый файл (EXE), имеющий цифровую подпись Google Inc.,
- `chrome_frame_helper.dll` — вредоносная динамическая библиотека (DLL).

После запуска исполняемого EXE-файла подгружается код вредоносной DLL. При этом библиотека проверяет имя процесса, в который она загружена, и оно должно соответствовать имени `ChromeFrameHelperSrv.exe`. В противном случае, вредоносный код завершит свое исполнение.

Далее библиотека расшифровывает вредоносный исполняемый файл, код которого внедряется в еще один запущенный процесс ChromeFrameHelperSrv.exe с использованием техники Process Hollowing.

Работа dloTrojan

Вредоносная программа пытается получить данные значения с именем TID из одного из двух ключей реестра (это зависит от имеющихся привилегий в системе):

- `HKLM\Software\VS`
- `HKCU\Software\VS`

Если же значение в реестре отсутствует, создается один из указанных ключей реестра. В параметре TID прописывается строка из 16 произвольных символов, которую в дальнейшем можно рассматривать как ID зараженной системы.

Далее dloTrojan проверяет наличие подключения к сети в зараженной системе. Для этого она пытается установить соединение с `www.microsoft[.]com`.

Строки во вредоносной программе зашифрованы методом простого сложения по модулю двух с одним байтом (отличается для различных строк).

Затем малварь расшифровывает адрес сервера управления. В зависимости от конфигурации вредоносная программа может иметь несколько адресов, в текущей конфигурации адрес сервера управления один.

Теперь dloTrojan устанавливает соединение с сервером управления. Если подключиться к серверу не удалось, малварь пытается найти настроенные прокси-серверы одним из способов:

- вызывает функцию `InternetQueryOptionA()` с параметром `INTERNET_OPTION_PROXY` и получает список доступных прокси-серверов;
- достает из реестра `HKEY_USERS\<user_SID>\Software\Microsoft\Windows\CurrentVersion\Internet Settings` из параметра `ProxyServer`.

Далее на сервер управления отправляется следующая информация о зараженной системе:

- имя компьютера,
- имя пользователя;
- IP-адреса зараженной системы,
- сведения о версии операционной системы,
- принадлежность компьютера к рабочей группе или домену,
- идентификатор оригинального производителя оборудования — результат работы функции GetOEMCP(),
- ID, хранимый в реестре.

Данные передаются на сервер управления в зашифрованном виде.

В конце концов вредоносная программа получает возможность исполнять команды сервера управления: запускать cmd-шелл, создавать и удалять файлы, собирать информацию о дисках.

Перечень возможных команд приведен в табл. 10.

И еще несколько программ, которые мы раскопали в ходе расследования

Вернемся к общедоступным утилитам, найденным на зараженных системах. С их помощью можно залезть в систему, утащить конфиденциальные данные и выполнить другие вредоносные действия. Ловите краткое описание каждой.

GetPassword

GetPassword предназначена для получения паролей из зараженной системы. Раньше исходный код утилиты лежал в репозитории MimikatzLite, но сейчас его почему-то удалили. Можем только поделиться скриншотом на рис. 9.

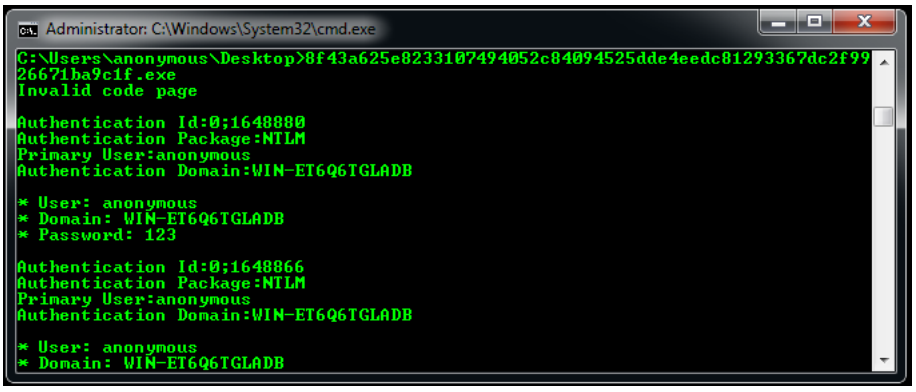

Рис. 9. Скриншот работы утилиты GetPassword

Quarks PwDump

Еще одна утилита для извлечения паролей из ОС Windows.

Исходный код можно найти в репозитории 0daytool-quarkspwdump. Скриншот утилиты приведен на рис. 10.

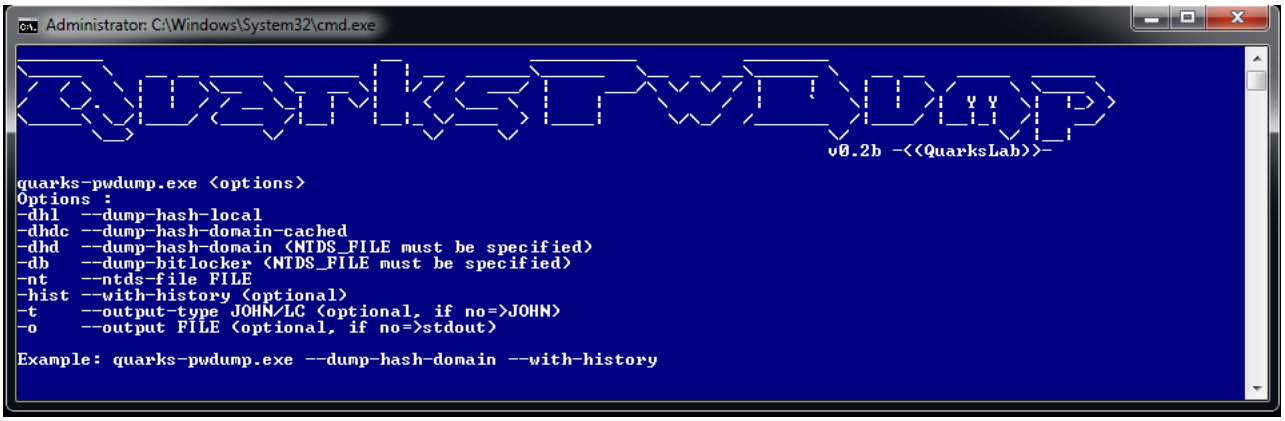

Рис. 10. Скриншот работы утилиты Quarks PwDump

wpmd v 2.3 (beta)

wpmd (windows password and masterkey decrypt) также предназначена для получения паролей в ОС Windows. Увы, источник мы не нашли, поэтому можем только показать скриншот (рис. 11).

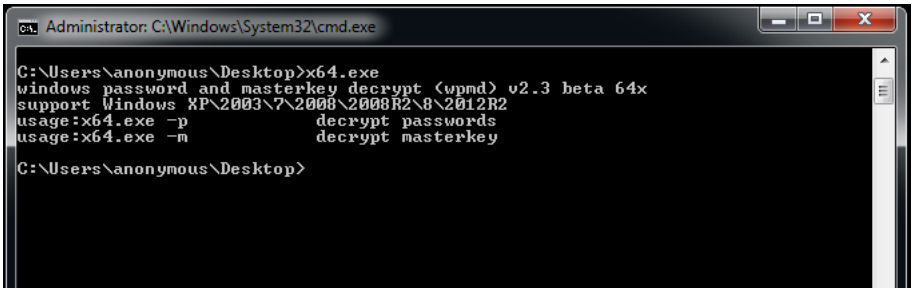

Рис. 11. Скриншот работы утилиты wpmd v 2.3 (beta)

os.exe

os.exe позволяет определить версию ОС Windows (рис. 12). Источник тоже не найден

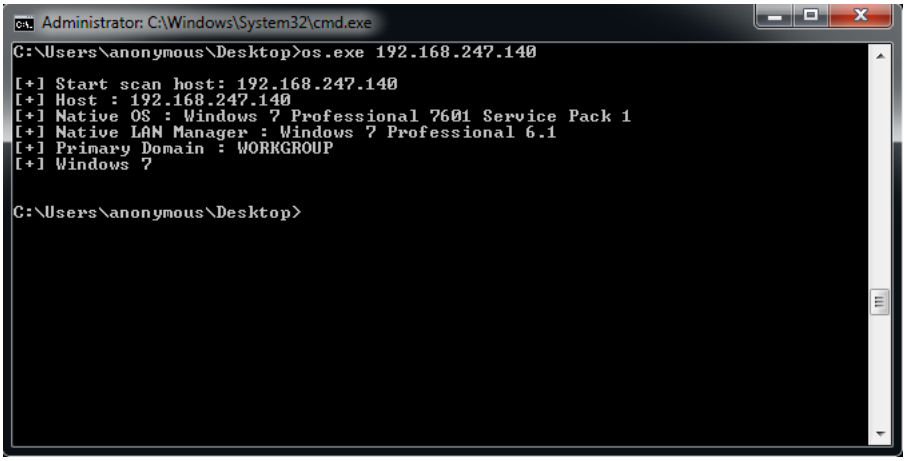

Рис. 12. Скриншот работы утилиты os.exe

nbtscan 1.0.35

nbtscan — утилита командной строки, предназначенная для сканирования открытых серверов имен NETBIOS в локальной или удаленной TCP/IP-сети. Она обеспечивает поиск открытых общих ресурсов (рис. 13). Доступна на ресурсе Unixwiz.net.

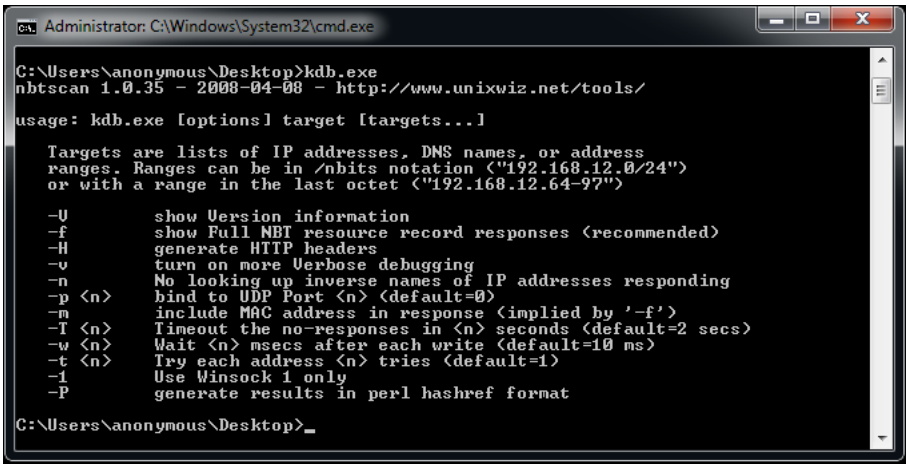

Рис. 13. Скриншот работы утилиты nbtscan

Это расследование в очередной раз убедило нас, что даже заезженные и понятные техники способны доставить жертвам много неприятностей. Злоумышленники могут годами копаться в IT-инфраструктуре жертвы, которая и подозревать ничего не будет. Думаем, выводы вы сделаете сами

PlugX (SHA256: EXE, DLL, Shell-code)
Code:

```
3124fcb79da0bdf9d0d1995e37b06f7929d83c1c4b60e38c104743be71170efe
2f81cf43ef02a4170683307f99159c8e2e4014eded6aa5fc4ee82078228f6c3c
0c831e5c3aecab14fe98ff4f3270d9ec1db237f075cd1fae85b7ffaf0eb2751e

94004d84edf72720b270a49bf673c98aba2e4da65dc5a8542566cec073ee7812
e3fcb055cf50884a192aca2f958f899eb0033c1a1b923deb7d56baaca9d7122d
55efc36799631f12f54dfa574aafa1c8e1d4d1b659c159253987d24fecc32185

18a98c2d905a1da1d9d855e86866921e543f4bf8621faea05eb14d8e5b23b60c
68d58f68591a8306b15de98913897a34bc96ffc6db10e4113144cc54aaa0dda4
5697c9086fd5abd030ceb937c396c6893ecc8d4a848785fac61ce13d5740edca

3124fcb79da0bdf9d0d1995e37b06f7929d83c1c4b60e38c104743be71170efe
97ad6e95e219c22d71129285299c4717358844b90860bb7ab16c5178da3f1686
81e53c7d7c8aa8f98c951106a656dbe9c931de465022f6bafa780a6ba96751eb
```

PlugX-executor: (SHA256: EXE)
Code:

```
2f73a3c7fa58d93d60d3011724af2c7beddc39469c0613ce097657849ab32e82
43f1bd298113934763205424 73d6c1dedea172a62ccf1a876a04a53ed876f3a4
```

nccTrojan (SHA256: EXE, DLL)
Code:

```
07d728aa996d48415f64bac640f330a28e551cd565f1c5249195477ccf7ecfc5
3be516735bafbb02ba71d56d35aee8ce2ef403d08a4dc47b46d5be96ac342bc9
```

dnsTrojan (SHA256: EXE)
Code:

```
826df8013af53312e961838d8d92ba24de19f094f61bc452cd6ccb9b270edae5
```

dloTrojan (SHA256: EXE, DLL)
Code:

```
be174d2499f30c14fd488e87e9d7d27e0035700cb2ba4b9f46c409318a19fd97
f0c07f742282dbd35519f7531259b1a36c86313e0a5a2cb5fe1dadcf1df9522d
```

Автор BiZone_team

Источник https://offzone.moscow/ru/