kaspersky

Техники, тактики и процедуры атак на промышленные компании. Импланты для сбора данных

Кирилл Круглов
Вячеслав Копейцев
Артём Снегирёв

31.07.2023 Version 1.2

Это вторая часть нашего исследования, посвященного анализу серии атак на промышленные компании.

В ходе кампании атакующие стремились организовать постоянно действующий канал для вывода украденных данных, включая данные, размещенные на физически изолированных (air-gapped) системах.

В общей сложности мы обнаружили более 15 имплантов и их вариантов, установленных злоумышленниками в разных сочетаниях.

Весь стек имплантов, примененных в атаках, можно разделить на три категории исходя из их ролей:

- Импланты первого этапа для обеспечения бесперебойного удаленного доступа и первоначального сбора данных
- Импланты второго этапа для сбора данных и файлов, в том числе с физически изолированных систем
- Импланты третьего этапа и инструменты для выгрузки данных на командные серверы

Эта часть исследования посвящена вредоносному ПО второго этапа, предназначенному для сбора данных на зараженных системах.

Было обнаружено два типа имплантов для сбора данных на зараженных системах:

1. импланты первого типа предназначены для сбора и архивирования различных данных на локальной машине,
2. импланты второго типа — для сбора информации о съёмных носителях, теневого копирования содержимого этих носителей и их заражения компьютерным червем, с помощью которого украденные данные затем выводятся из физически изолированных сетей.

Полный текст отчёта опубликован на портале Kaspersky Threat Intelligence. Дополнительную информацию вы можете запросить по адресу ics-cert@kaspersky.com.

Специализированный имплант для сбора файлов на локальной машине

В мае 2022 года был обнаружен специализированный имплант, предназначенный для сбора файлов на локальном компьютере. Имплант использует схему загрузки, основанную на подмене DLL (DLL hijacking), при которой загрузчик вредоносной DLL обеспечивает закрепление в системе, создавая службу с именем «WinSystemHost», расшифровывая и внедряя в память легитимного процесса вредоносную нагрузку, которая хранится в виде бинарных данных в отдельном файле.

Поток выполнения загрузчика состоит из трёх шагов.

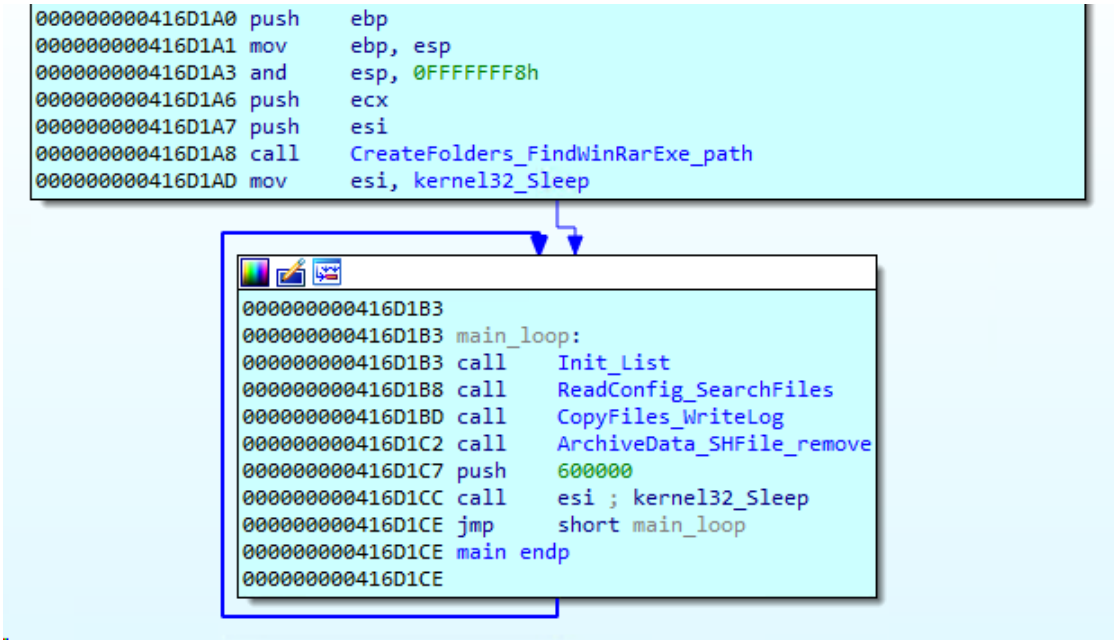
- При запуске без параметров он создает службу, которая запускает его с параметром «--2»
- При запуске с параметром «--2» он перезапускается с параметром «--1»
- При запуске с параметром «--1» он запускает процесс «msiexec.exe», читает и расшифровывает вредоносную нагрузку и внедряет ее в память процесса «msiexec.exe»

Сразу же после начала выполнения в памяти «msiexec.exe» вредоносная нагрузка переходит к бесконечному циклу, состоящему из 6 простых шагов.

- Создание папок для хранения файлов (если их не существует) и поиск пути к «WinRar.exe»
- Расшифровка строк
- Чтение конфигурации и поиск файлов на всех дисках
- Копирование файлов и запись сообщения в журнал

- Архивирование скопированных файлов и удаление временных объектов
- Пауза 10 минут

Главный цикл импланта для сбора файлов на локальной машине



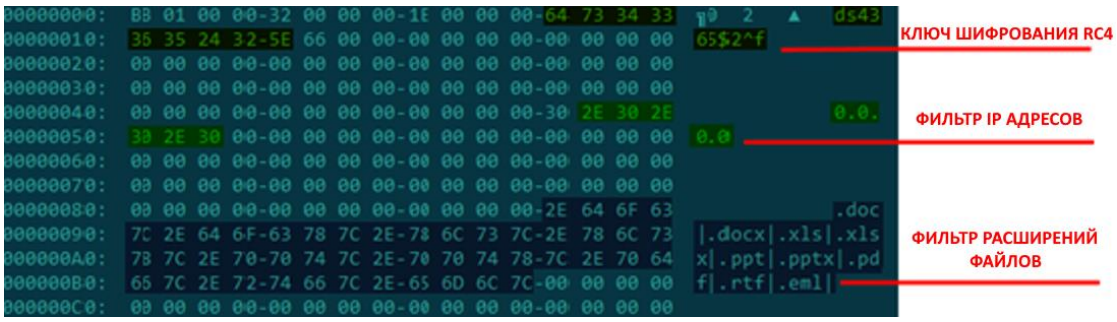
Вначале имплант создает папку «C:\ProgramData\NetWorks», затем подпапку для хранения временных файлов («C:\ProgramData\NetWorks\fl») и подпапку для хранения архивированных данных («C:\ProgramData\NetWorks\ZZ»). Затем он ищет существующее приложение «WinRar.exe» в %ProgramFiles% и поддиректориях C:\Windows\SysWow64.

Если найти файл «WinRar.exe» не удается, имплант завершает свою работу.

Цикл, входящий в состав главной функции, начинается с расшифровки строк, зашитых в код импланта. Затем имплант проверяет свой файл конфигурации — «C:\ProgramData\NetWorks\gfc» — и, если он не существует, имплант удаляет все ранее созданные папки и завершает свою работу. Это указывает на то, что запускать имплант предполагалось в заранее подготовленном окружении, где конфигурационный файл присутствует.

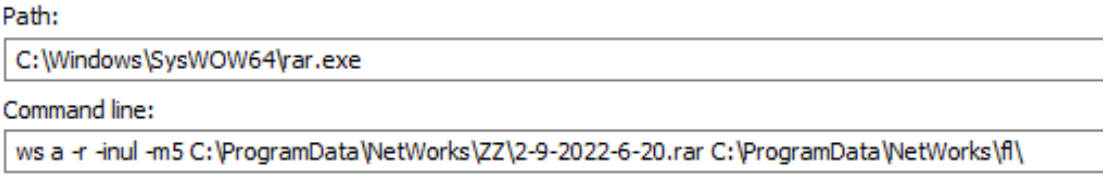
Файл конфигурации зашифрован с помощью алгоритма RC4 с ключом «bGkds&sy6$^3gsa» и используется для хранения еще одного ключа RC4, фильтра IP-адресов и списка расширений файлов, по которым следует фильтровать файлы, копируемые в папку «C:\ProgramData\NetWorks\fl». Имплант также записывает все пути в файл «1.log». Сразу же после прочтения файла конфигурации имплант начинает искать файлы на всех накопителях, подключенных к зараженной машине.

Расшифрованное содержимое файла конфигурации «gfc»



Все файлы с расширениями, соответствующими фильтру, копируются в «C:\ProgramData\NetWorks\fl». Затем вызывается «WinRar.exe» для создания архива в «C:\ProgramData\NetWorks\ZZ». Имя файла архива составляется из текущих даты и времени.

Вызов WinRar.exe для архивирования собранных файлов



После создания архива имплант использует команду SHFileOperationW для удаления файлов, находящихся в папке «C:\ProgramData\NetWorks\fl». В качестве последнего шага цикла имплант делает паузу на 10 минут.

Для вывода собранных файлов злоумышленники используют стек имплантов, предназначенных для выгрузки архивов в сервис Dropbox. Описание данного стека имплантов дано в третьей части отчёта «Импланты для выгрузки данных на сервер».

Стек имплантов для копирования данных из изолированных сетей через съёмные носители

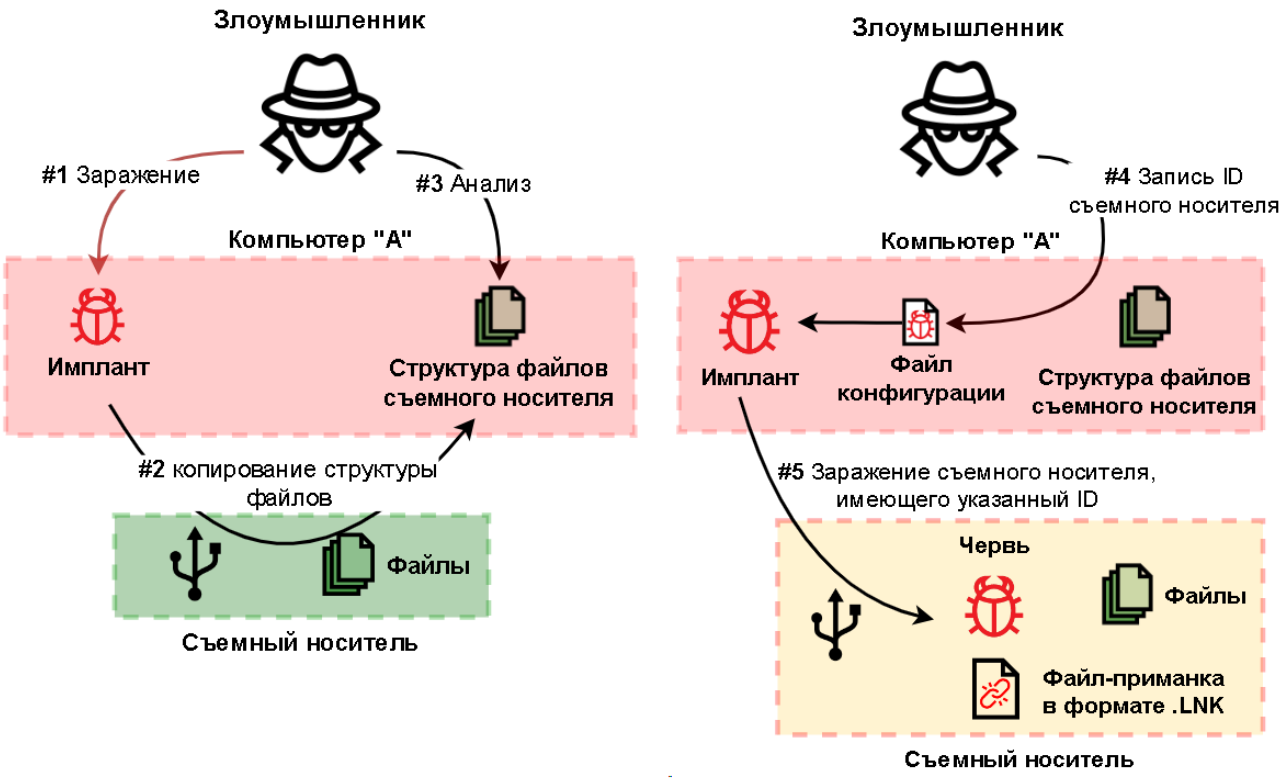
В апреле 2022 года мы обнаружили вредоносное ПО, предназначенное для копирования данных из физически изолированных (air-gapped) систем посредством заражения съёмных носителей. Обнаруженное вредоносное ПО состоит как минимум из трех модулей.

Шаг 1

Первый (главный) модуль отвечает за работу со съёмными носителями, включая сбор сведений о носителе, клонирование структуры файловой системы каждого носителя во временную локальную папку и поддержание структуры в актуальном состоянии, сбор украденных файлов с носителя и установку вредоносного ПО второго шага на вновь подключаемые носители, а также создание снимков экрана и сохранения заголовков окон на зараженной машине.

Поведение модуля можно настраивать с помощью файла конфигурации, размещаемого на хосте по статическому пути «C:\Users\Public\Libraries\main.ini». Файл конфигурации определяет, должен ли имплант пытаться осуществить заражение носителя и, если должен, какой метод заражения следует предпочесть.

Вначале главный модуль создает папку в «%TEMP%» (например, «TCABC8.tmp») в которой он в дальнейшем будет хранить журналы, информацию о подключенных носителях и их содержимом. Поскольку точное имя папки жестко закодировано в модуле, мы можем утверждать, что развертывалось по крайней мере 4 варианта данного импланта (по числу обнаруженных уникальных имен папок).



Упрощенная схема взаимодействия со съемными носителями

Затем для каждого съёмного носителя (от «D:» до «Z:») имплант создает подпапку (например, «%TEMP%\TCABC8.tmp\12345678»), где имя подпапки совпадает с серийным номером накопителя. Эти подпапки впоследствии используются для хранения журналов, создаваемых имплантом; копии структуры файловой системы каждого носителя, включая атрибуты каждого файла, но без содержимого файлов; а также украденных файлов и файлов, содержащих результаты работы третьего и четвертого модулей.

На каждом съёмном носителе имплант создает скрытую папку с именем «$RECYCLE.BIN» в корневой директории носителя и пустой файл с именем «S-1-5-21-963258» в этой папке. Этим файлом носитель помечается как зараженный. Кроме того, имплант создает такой же пустой файл в «c:\windows\tasks\S-1-5-21-963258», чтобы пометить хост как зараженный и не позволить имплантам следующих шагов собирать данные на этом хосте.

Имплант проверяет наличие следующих файлов в «%TEMP%\TCABC8.tmp», используемых для заражения съёмного носителя с серийным номером, соответствующим названию папки:

- «mcods.exe» — легитимного исполняемого файла McAfee, содержащего уязвимость подмены DLL
- «McVsoCfg.dll», который является вредоносной нагрузкой второго шага
- Файлов «DOC», «PDF» или «DIR», которые определяют, какой файл ярлыка будет использован в качестве приманки

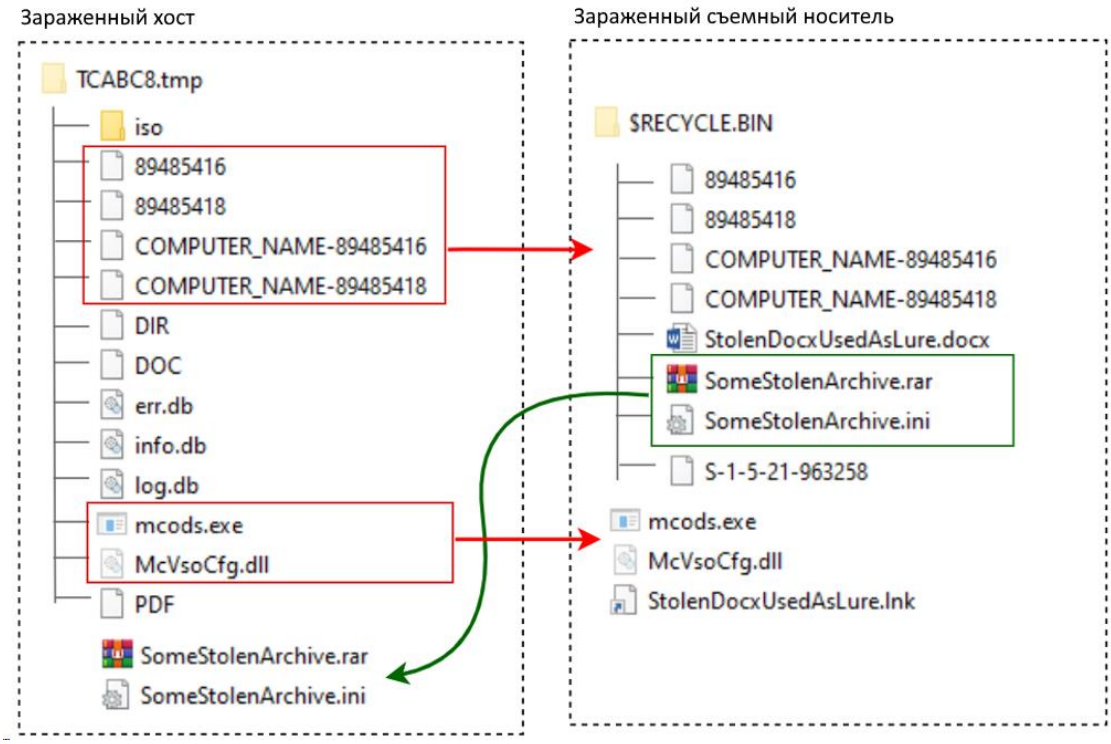
Наличие перечисленных файлов в папке, отведенной для конкретного съемного носителя, указывает на то, что злоумышленники предварительно анализируют содержимое съемных носителей некоторое время и лишь затем копируют файлы, используемые для заражения конкретного съёмного носителя, в заданную папку.

Шаг 2

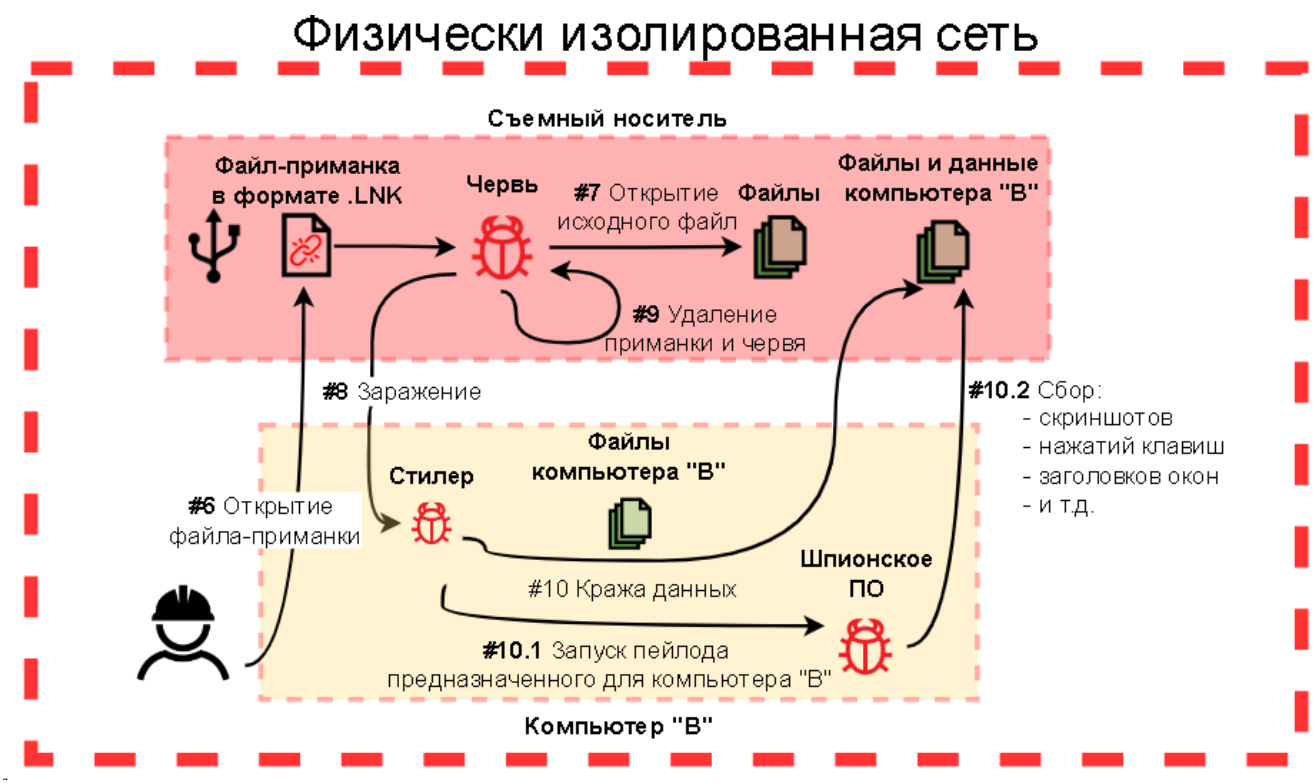
Чтобы заразить съёмный носитель, главный модуль просто копирует в его корневую директорию два файла — «mcods.exe» и «McVsoCfg.dll» — и устанавливает у обоих файлов атрибут «Hidden».

Кроме того, может существовать файл вредоносной нагрузки четвертого шага с именем «89485416» или «89485418». Если он существует, то он также будет скопирован на съёмный носитель вместе с имплантом второго шага.

Структура файлов во временной папке на зараженном хосте и на зараженном съёмном носителе



Затем, в зависимости от конфигурации и расширения файла, найденного во временной директории («DOC», «PDF» или «DIR»), главный модуль осуществляет рекурсивный поиск файла (если расширение «.docx» или «.pdf») или папки в корневой директории накопителя. В обоих случаях имя найденного объекта используется для создания файла ярлыка с именем «[имя документа или папки].lnk» (в примере на рисунке выше файл назван «StolenDocxUsedAsLure.lnk») в корневой директории съемного носителя, который будет использоваться в качестве приманки, а исходный объект перемещается в «$RECYCLE.BIN» (если это файл) или, если это папка, просто помечается как скрытый («hidden») и оставляется для последующего использования.



Упрощенная схема заражения компьютера в изолированном сегменте сети через зараженный съемный носитель

Предназначенный для использования в качестве приманки файл ярлыка создается со следующим значением свойства «Target»: «rundll32.exe url.dll,FileProtocolHandler mcods.exe». При открытии пользователем ярлыка, который является файлом-приманкой, ОС загрузит в память файл «mcods.exe», который загрузит в память библиотеку «McVsoCfg.dll» и вызовет ее функцию «McVsoCfgGetObject». Такая длинная цепочка загрузок и вызовов имеет смысл, потому что вредоносное ПО второго шага — «McVsoCfg.dll» — должно выполняться без ведома пользователя и предпримет попытку удалить промежуточный исполняемый файл «mcods.exe». Это будет возможно только в том случае, если дескриптор файла «mcods.exe» будет закрыт в момент попытки ОС удалить файл.

При запуске на выполнение вредоносного ПО второго шага, размещенного на носителе, оно прежде всего ищет документ-приманку в «$RECYCLE.BIN» или папку, имеющую то же имя, что приманка. Затем оно проверяет, существует ли файл-метка «c:\windows\tasks\S-1-5-21-963258». Если он существует, имплант просто открывает исходный файл или папку, вызывая ShellExecuteW, и завершает свою работу.

Фрагмент кода червя второго шага «McVsoCfg.dll» — проверка, заражен ли хост, и открытие исходного документа или папки, чтобы сбить жертву с толку

```
wcscat_s(LURE_LNK, 0x104u, L".lnk");
if ( PathFileExistsW(L"c:\\windows\\tasks\\S-1-5-21-963258") )
{
  ShellExecuteW(0, L"open", &ORIGINAL_ENTITY_USED_FOR_LURE, 0, 0, 1);
  ExitProcess(0);
}
if ( MoveFileW(&ORIGINAL_ENTITY_USED_FOR_LURE, DstFileName) )
{
  ShellExecuteW(0, L"open", DstFileName, 0, 0, 1);
  DeleteFileW(LURE_LNK);
  DeleteFileW(&SELF_EXE);
}
```

В противном случае, если файл-метка не существует, вредоносное ПО копирует исходный файл из «$RECYCLE.BIN» в корневую директорию носителя, открывает его с помощью ShellExecuteW, затем удаляет ярлык-приманку и легитимный файл «mcods.exe» со съемного носителя. Сразу же после этого имплант устанавливает исполняемый файл вредоносного ПО третьего шага, извлекая его из своего собственного файла («McVsoCfg.dll») и сохраняя его в «%APPDATA%» с именем «msgui.exe» на атакуемом хосте. Он также создает файл-ярлык, указывающий на «msgui.exe», и затем открывает ярлык, вызывая ShellExecuteW.

Фрагмент кода червя второго шага «McVsoCfg.dll» — заражение хоста путем извлечения и запуска на выполнение вредоносного ПО третьего шага

```
SHGetSpecialFolderPathW(0, &pszPath, 7, 0);
wcscat_s(&pszPath, 0x104u, L"\\WordPress.lnk");
if ( !PathFileExistsW(&pszPath) )
  CREATE_DOC_SHORTCUT_RUNDLL_MSGUI((int)&pszPath);
GetTempPathW(0x104u, &Buffer);
wcscat_s(&Buffer, 0x104u, L"msgui.exe");
DeleteFileW(&Buffer);
v16 = CreateFileW(&Buffer, 0x40000000u, 0, 0, 1u, 0, 0);
if ( v16 != (HANDLE)-1 )
{
  WriteFile(v16, &PAYLOAD, 0xB000u, &NumberOfBytesWritten, 0);
  CloseHandle(v16);
}
ShellExecuteW(0, L"open", &Buffer, 0, 0, 0);
```

По завершении процедуры заражения имплант второго шага («McVsoCfg.dll») предпринимает попытку удалить себя с носителя с помощью пакетного сценария, выполняемого посредством ShellExecuteW, который посылает ping-запрос на localhost (чтобы дать процессу импланта на хосте время завершить свое выполнение) и затем удаляет модуль.

Фрагмент кода червя второго шага — «McVsoCfg.dll», удаляющего себя с зараженного носителя

```
wcscat_s(&v23, 0x104u, &SELF_PATH);
wcscat_s(&v23, 0x104u, L"McVsoCfg.dll");
if ( GetEnvironmentVariableW(L"ComSpec", &v20, 0x400u) )
{
  v43 = 34;
  *Parameters = aCPingLocalhost;              // /c ping localhost & del
  v41 = MEMORY[0x10010BA4];
  v42 = MEMORY[0x10010BB4];
  wcscat_s(Parameters, 0x104u, &SelFilePath);
  wcscat_s(Parameters, 0x104u, L"\" /A:H & del \"");
  wcscat_s(Parameters, 0x104u, &v23);
  wcscat_s(Parameters, 0x104u, L"\" /A:H");
  ShellExecuteW(0, 0, &v20, Parameters, 0, 0);
}
ExitProcess(0);
```

Шаг 3

Имплант третьего шага — «msgui.exe» — имеет малый размер и простую функциональность. Он предназначен для выполнения пакетного скрипта для сбора данных с помощью «cmd.exe» и сохранения результатов его работы в папку «$RECYCLE.BIN» на носителе для последующего сбора главным модулем вредоносного ПО (при подключении к первоначально зараженному хосту). Затем он ищет файл вредоносного ПО четвертого шага, который следует запустить на выполнение и затем удалить (если он существует).

Фрагмент импланта «msgui.exe» — команды (CMD) для сбора информации

```
rsrxfPчRАдми.истраторы..89485418
....89485416....-...S-1-5-21-963
258.\$RECYCLE.BIN\..exit....del
%tmp%\*.exe.....dir·g:\·>>..dir
f:\·>>..dir·e:\·>>..dir·d:\·>>..
dir·c:\·>>.."·>>....dir·"...dir
"C:\Program·Files·(x86)"·>>.dir
"C:\Program·Files"·>>...net·grou
p·/do·>>....net·group·administra
tors·/do·>>.net·localgroup·admin
istrators·>>..../do·>>..net·grou
p·..·>>.net·localgroup·.tasklist
·/v>>...netstat·-ano·-p·tcp·>>..
ipconfig·/all·>>........systemin
fo·>....\cmd.exe....8F9540EN46..
\...open.....bat.....exe........
```

```
GetComputerNameA_0(COMPUTER_NAME);
v4 = CreateThread(0, 0, StartAddress, 0, 0, 0);// Execute batch cmd via pipes that save output in a file named as COMPUTER_NAM
WaitForSingleObject(v4, 0xFFFFFFFF);
while ( 1 )
{
  v5 = 'D';
  do
  {
    v9 = 0;
    v10 = 0;
    DrivePath_D_Z = v5;
    v11 = 0;
    v8 = 58;
    v12 = 0;
    v6 = GetDriveTypeA(&DrivePath_D_Z);
    if ( v6 == 3 || v6 == 2 )                   // REMOVABLE | FIXED
    {
      strcpy(&pszPath, &DrivePath_D_Z);
      strcat(&pszPath, aRecycle_bin);           // \$RECYCLE.BIN\
      strcat(&pszPath, aS1521963258);           // S-1-5-21-963258
      if ( PathFileExistsA(&pszPath) )
        COPY_FROM_TEMP_TO_DRIVE(&DrivePath_D_Z);
      strcpy(&FileName, &DrivePath_D_Z);
      strcat(&FileName, aRecycle_bin);
      strcat(&FileName, COMPUTER_NAME);
      strcat(&FileName, dash);
      strcat(&FileName, a89485416);
      if ( PathFileExistsA(&FileName) )
        IfPathExists_ShellExecute_and_Delete(&FileName);
      strcpy(&FileName, &DrivePath_D_Z);
      strcat(&FileName, aRecycle_bin);
      strcat(&FileName, a89485416);
      if ( PathFileExistsA(&FileName) )
        IfPathExists_ShellExecute_and_Delete(&FileName);
```

Фрагмент кода вредоносного ПО третьего шага — «msgui.exe», — для сбора информации на хосте и запуска вредоносного ПО четвертого шага (если оно существует)

Шаг 4

Вредоносное ПО четвертого шага состоит из двух файлов:

- Простого дроппера вредоносной нагрузки (аналогичного тому, что используется вредоносным ПО второго шага)
- Вредоносной нагрузки, фактически представляющей собой видоизмененный модуль первого шага, также предназначенный для сбора информации о носителе, сбора файлов, создания снимков экрана и регистрации нажатий клавиш, но без процедуры заражения съёмного носителя

Файл(ы) вредоносного ПО четвертого шага могут иметь имя «89485416» или «89485418». В первом случае вредоносная нагрузка представляет собой пакетный скрипт, во втором — исполняемый файл. Имя файла также может иметь префикс [ИМЯ_КОМПЬЮТЕРА], в таком случае вредоносное ПО четвертого шага развертывается только на указанном компьютере (в противном случае вредоносная нагрузка будет выполняться безотносительно к имени компьютера). Код вредоносного ПО четвертого шага имеет небольшой объём и реализует две основных процедуры:

- Сбор и архивирование файлов
- Запись нажатий клавиш, заголовка окна и снимка экрана

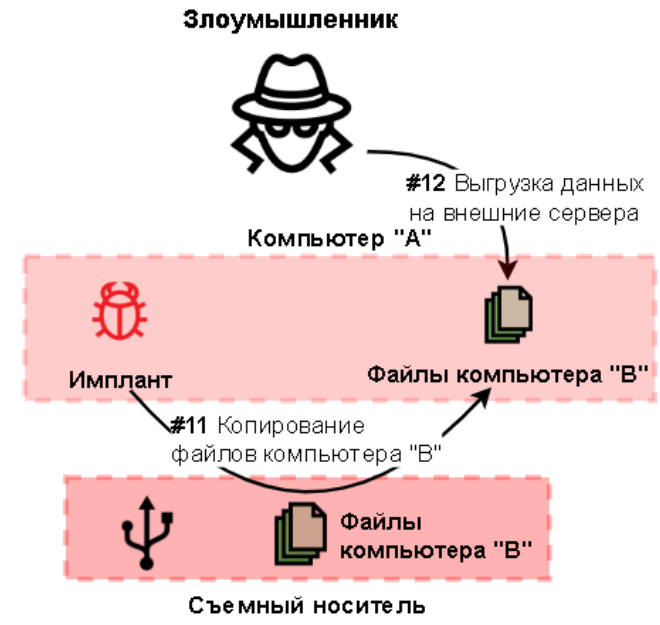
Процедура сбора файлов компилируется из того же исходного кода, что и соответствующая процедура, используемая главным модулем, а код для снятия снимков экрана и регистрации заголовка окна новый — как и код для записи нажатий клавиш.

Оба модуля (первого шага и четвертого шага) имеют аналогичные конфигурации и процедуры сохранения данных:

- Поведение обоих модулей зависит от настроек, сохраненных в «.ini» файле (вредоносное ПО четвертого шага использует «C:\Users\Public\Libraries\setting.ini», а вредоносное ПО первого шага — «C:\Users\Public\Libraries\main.ini»).
- В зависимости от настроек оба модуля могут собирать информацию о носителе, а также сохранять снимки экрана и заголовки окон зараженного хоста, искать и копировать документы (.doc, .docx, .xls, .xlsx, .ppt, .pptx) и изображения (.png, .jpeg, .jpg, .bmp).
- Оба модуля сохраняют базовую информацию (атрибуты) на съёмном носителе в файле «log.db» с использованием одинакового формата данных; при этом дополнительная информация (включая список файлов и директорий) сохраняется в файл «info.db», а сообщения об ошибках сохраняются в «err.db».

Чтобы получить все украденные данные, злоумышленники используют удаленную оболочку для запуска имплантов, предназначенных для выгрузки данных на сервер. Эти импланты описаны в следующем разделе отчета.

Упрощенная съема сбора данных, полученных с компьютера в изолированном сегменте сети, через зараженный съемный носитель



Заключение

Злоумышленники стремились усложнить обнаружение и анализ угрозы. Поэтому вредоносная нагрузка хранится в зашифрованном виде в отдельном бинарном файле данных, а вредоносный код скрывается в памяти легитимных приложений с помощью техники подмены DLL и внедрения кода в память процессов.

Для многих кампаний APT и целенаправленного кибершпионажа отправка данных из физически изолированных сетей — общая процедура. Несмотря на существование широкого разнообразия методов вывода данных из таких сетей, в большинстве случаев злоумышленники выбирают техники, тактики и процедуры (TTP), основанные на заражении съёмных носителей.

Рекомендации

- Установите защитное решение с поддержкой централизованного управления политиками безопасности на все серверы и рабочие станции и поддерживайте антивирусные базы и программные модули всех защитных решений в актуальном состоянии.
- Убедитесь, что все компоненты защитного решения включены на всех системах и что действует политика, требующая ввода пароля администратора при любой попытке отключить защиту.
- Обеспечьте регулярную проверку всех съёмных носителей, используемых в технологической сети.
- Рассмотрите возможность применения технологий разрешительных списков и контроля программ, чтобы предотвратить выполнение неизвестных приложений.
- Рассмотрите возможность применения технологий контроля устройств, чтобы обеспечить безопасное использование всех съёмных устройств.
- Убедитесь, что политики Active Directory предусматривают ограничения на попытки входа в систему для пользователей. Пользователь должен иметь возможность входа только в те системы, которые ему необходимы для выполнения служебных обязанностей.
- Ограничьте использование учетных записей с правами локального администратора и администратора домена, за исключением случаев, когда такие права необходимы для выполнения служебных обязанностей.
- Рассмотрите возможность использования специализированного решения для управления паролями учетных записей локальных администраторов на всех системах.

- Внедрите парольную политику, предусматривающую минимальные требования к уровню сложности паролей и требующую регулярной смены паролей.
- Рассмотрите возможность использования сервисов класса Managed Detection and Response для получения оперативного доступа к знаниям и опыту экспертов высокого уровня в области безопасности.

Приложение I — Индикаторы компрометации

Замечание: Индикаторы в этом разделе актуальны на момент публикации.

Полная версия индикаторов компрометации, в том числе правила Yara, доступна в .ioc-файле на портале Kaspersky Threat Intelligence.

Специализированный имплант для сбора данных на локальной системе

MD5

```
4C1ADC1778CE07CD655DB129AF1DA7E0 (DynTray.dll)
71D919105627C67AB9FB9A7152015CF6 (Data)
```

Стек имплантов для отправки данных из физически изолированных сетей

MD5

```
3E22E7F5A6EE0A7D3D9A5CBFA7939C98 (tmp.exe)
2DB858C4CA836120D3124EB5490195EA (main.ini)
D2D7FD5C7372CD81D6BC4199F211A42C (RtkAudio.exe)
4D5963B7D931A02265EA5231961935E9 (mcvsocfg.dll)
3A532B8481F22B78ABC718AC5CDB3F06 (msgui.exe)
36A029CB62BFCB86394B49E5ACF36BEF (SCR)
1DBC1DEFC2AC6578D83D5C45D9836482 (abbyfine.exe)
9F402F0B2C84ED577E9EE76DCF640B70 (f04803w3.exe)
0E69850A0F67165D4E3D06987D14B2E6 (automonitor.exe)
C929DCC69CF6546D56C2A68D31D7728D ($rjkdi4v.exe)
```

Приложение II — категории MITRE ATT&CK

Представленная ниже таблица содержит все тактики, техники и процедуры, обнаруженные при анализе активности, описываемой в настоящем отчете.

Тактика	Номер техники	Название и описание техники
Initial Access	**T1566.001**	**Phishing: Spearphishing Attachment** Использование злоумышленниками документов-приманок для развертывания стандартного шпионского ПО.
Execution	**T1204.002**	**User Execution: Malicious File** Заражение системы при запуске вредоносного ПО пользователем, считающим, что это легитимный документ.
	T1059.003	**Command and Scripting Interpreter: Windows Command Shell** Использование cmd.exe для выполнения серии команд.
	T1106	**Native API** Использование функции CreateProcessW для выполнения команд в интерпретаторе командной строки Windows
	T1053.005	**Scheduled Task/Job: Scheduled Task** Выполнение вредоносного ПО с помощью созданной злоумышленниками задачи планировщика задач Windows.
Persistence	**T1547.001**	**Registry Run Keys / Startup Folder:** Закрепление вредоносного ПО в системе путем его добавления в ключ автозапуска системного реестра.
	T1543.003	**Create or Modify System Process: Windows Service** Установка вредоносным ПО себя в качестве службы для закрепления в системе.
	T1053.005	**Scheduled Task/Job: Scheduled Task** Выполнение вредоносного ПО посредством созданной злоумышленниками задачи планировщика задач Windows.
Defense Evasion	**T140**	**Deobfuscate/Decode Files or Information** Использование RC4-ключа для расшифровки конфигурации вредоносного ПО и сетевого взаимодействия.
	T1055.002	**Process Injection: Portable Executable Injection** Внедрение при выполнении вредоносного ПО его кода в различные легитимные процессы (msiexec.exe, svchost.exe).

	T1497.001	**System Checks** Осуществление различных проверок системы с целью обнаружить и предотвратить выполнение в средах виртуализации и анализа.
	T1497.003	**Time Based Evasion** Использование различных методов, основанных на учете времени, для обнаружения и избегания сред виртуализации и анализа.
	T1574.002	**Hijack Execution Flow: DLL Side-Loading** Использование злоумышленниками бинарных файлов легитимных приложений для загрузки вредоносной DLL.
Discovery	**T1083**	**File and Directory Discovery** Попытки со стороны вредоносного ПО обнаружить файлы различных типов (.doc, .docx, .xls, .xlsx, .ppt, .pptx, .pdf, .rtf, .eml).
	T1016	**System Network Configuration Discovery** Использование злоумышленниками утилит netstat и ipconfig для получения конфигурации локального сетевого интерфейса и списка открытых портов.
	T1033	**System Owner/User Discovery** Использование злоумышленниками утилит systeminfo, whoami и net для получения информации о пользователе и зараженной системе.
	T1057	**Process Discovery** Использование злоумышленниками tasklist для получения списка активных процессов.
Collection	**T1005**	**Data from Local System** Применение вредоносного ПО, предназначенного для сбора и отправки произвольных данных, в том числе с физически изолированных систем, через съёмные носители.
	T1052.001	**Data from Removable Media** Сохранение вредоносным ПО всех собранных данные на определенном зараженном USB-носителе для их последующего вывода из физически изолированной сети.

Центр реагирования на инциденты информационной безопасности промышленных инфраструктур «Лаборатории Касперского» (Kaspersky ICS CERT) — глобальный проект «Лаборатории Касперского», нацеленный на координацию действий производителей систем автоматизации, владельцев и операторов промышленных объектов, исследователей информационной безопасности при решении задач защиты промышленных предприятий и объектов критически важных инфраструктур.

Kaspersky ICS CERT ics-cert@kaspersky.com