kaspersky

Техники, тактики и процедуры атак на промышленные компании. Импланты для удаленного доступа

Кирилл Круглов
Вячеслав Копейцев
Артём Снегирёв

20.07.2023 Version 1.1

В 2022 году мы расследовали серию атак на промышленные компании в Восточной Европе. В ходе описанных кампаний атакующие стремились организовать постоянно действующий канал для вывода украденных данных, включая данные, размещенные на физически изолированных (air-gapped) системах.

Исходя из обнаруженного сходства этих кампаний с ранее исследованными (такими как ExCone, DexCone), включая использование вариантов FourteenHi, конкретных тактик, техник и процедур (ТТП), а также выбор целей атаки, мы можем предположить со средней или высокой степенью уверенности, что за активностью, описанной в этом отчете, стоят злоумышленники из группы APT31, также известной как Judgment Panda и Zirconium.

Злоумышленники выводят украденные данные и доставляют вредоносное ПО следующего этапа через облачное хранилище данных, например, Dropbox, Yandex Disk, а также файлообменный сервис. Кроме того, они используют командные серверы, размещенные на обычных виртуальных выделенных серверах (VPS). Атакующие также используют стек имплантов для сбора данных в физически изолированных сетях с помощью зараженных сменных носителей данных.

В большинстве имплантов злоумышленники применяют сходные реализации техники подмены DLL (DLL hijacking), которые часто связывают с вредоносным ПО Shadowpad, и методы внедрения в память, а также шифрование RC4, позволяющее скрывать вредоносную нагрузку и избегать обнаружения. Кроме того, библиотека libssl.dll или libcurl.dll была статически прилинкована к имплантам, чтобы реализовать шифрованный обмен данными с серверами управления.

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

В статье (это первая часть отчёта) мы анализируем распространенные тактики, техники и процедуры (tactics, techniques and procedures, TTP), применяемые злоумышленниками в имплантах первого этапа с целью организации устойчивого канала для удаленного доступа к инфраструктуре промышленных предприятий.

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

Варианты FourteenHi

FourteenHi — это семейство вредоносных программ, обнаруженное в 2021 году при расследовании кампании, получившей название ExCone (см. здесь и здесь). Эта кампания была активна с середины марта 2021 года и была нацелена на государственные учреждения. В 2022 году мы обнаружили новые варианты программ из этого семейства, задействованные в атаках на инфраструктуру промышленных предприятий.

Разные образцы FourteenHi (как x64, так и x86) существенно отличаются друг от друга в плане структуры кода, реализации загрузчиков и типов командных серверов. Но их главные особенности, такие как используемый командными серверами протокол связи и список команд, практически одинаковы. Наиболее значительные различия обнаруживаются между вариантами FourteenHi для архитектур x86 и x64.

Образцы для архитектуры x64 имеют функции закрепления в системе и двухэтапный протокол взаимодействия с командным сервером. Они поддерживают длинный перечень команд, в том числе:

- выгрузить произвольные файлы на сервер,
- скачать произвольные файлы,
- выполнить произвольные команды,
- установить задержку передачи данных,

- запустить обратную оболочку,
- завершить собственный процесс и удалить механизмы закрепления в системе.

Для защиты связи с командным сервером в этих образцах используется API статически прилинкованной библиотеки OpenSSL. Кроме того, в них применяется алгоритм RC4 для шифрования / дешифрования данных, отправляемых на командный сервер и получаемых с него.

Код FourteenHi x64, производящий обработку ответа командного сервера

```
if ( command == 0x253AB )
{
  if ( v7 == 4 )
  {
    handle = kernel32_CreateRemoteThreat_902(0i64, 0i64, C2_command_ReadWrite_file, *buffer);
    kernelbase_CloseHandle_425(handle);
  }
}
else if ( command == 0xB8C2D )
{
  exception = check_alloc_exception(24i64);
  qword_1BAAAC251D8 = exception;
  *exception = 0i64;
  exception[1] = -1i64;
  exception[2] = -1i64;
  CreateRemoteThreat_C2_command_CMD_exec(exception, cmd_command, buffer, SHIDWORD(command));
}
```

Код FourteenHi x64, выполняющий обработку команд в ответе командного сервера

```
if ( subCommand == 0xDE372 )
{
  WriteFile(data, &filePath, dataLen);
}
else if ( subCommand == 0xCB76F )
{
  ReadFile(data, &filePath, dataLen);
}
```

Образцы для архитектуры x86 не имеют функционала закрепления в системе, не используют OpenSSL, но также используют шифрование RC4. Протокол взаимодействия с командным сервером одноступенчатый, однако список команд почти такой же, что в случае образцов x64, за исключением удаления механизмов закрепления в системе.

Таблица функций FourteenHi x86 для обработки команд в ответах командного сервера

```
                align 4
CNC_CommandCode dd offset Command_841_ExecCmd_Send_CnC
                                        ; DATA XREF: sub_401E30+17E↑r
                dd offset Command_842_WriteFile ; jump table for switch statement
                dd offset Command_843_ReadFile_Send_CNC
                dd offset Command_844_Sleep
                dd offset Command_845_Exit
                align 10h
```

Отсутствие функциональности по закреплению в системе (которая, как правило, требует повышения уровня привилегий) в вариантах для архитектуры x86, а также незначительный размер скомпилированного кода делает их удачными вариантами для применения на этапе первоначального заражения. На этом этапе может происходить сбор информации об атакуемом хосте или локальной сети, загружаться вредоносное ПО следующего этапа и специализированные программы для кражи данных, а также предоставляться удаленная оболочка для злоумышленников.

При этом злоумышленники могут без проблем обеспечить закрепление импланта в системе, создав задачу в планировщике задач Windows — это поведение, которое мы наблюдали в дикой среде.

Схема загрузки всех вариантов более или менее одинакова и состоит из трех основных компонентов, применяемых злоумышленниками на машине жертвы:

1. Легитимное приложение, уязвимое для подмены DLL.
2. Вредоносная DLL, загружаемая путем подмены DLL и используемая для чтения из бинарного файла данных и расшифровки вредоносной нагрузки FourteenHi и ее внедрения в системный процесс, например, svchost.exe или msiexec.exe.
3. Бинарный файл данных, содержащий двоичный код FourteenHi, зашифрованный алгоритмом RC4.

Все известные варианты FourteenHi содержат внедренные в код конфигурационные данные, зашифрованные алгоритмом RC4. Конфигурация включает идентификатор кампании (campaign ID), адрес командного сервера и порт. Конфигурация FourteenHi x64 включает также имя и описание создаваемой при запуске без параметров службы Windows, которая используется для закрепления в системе.

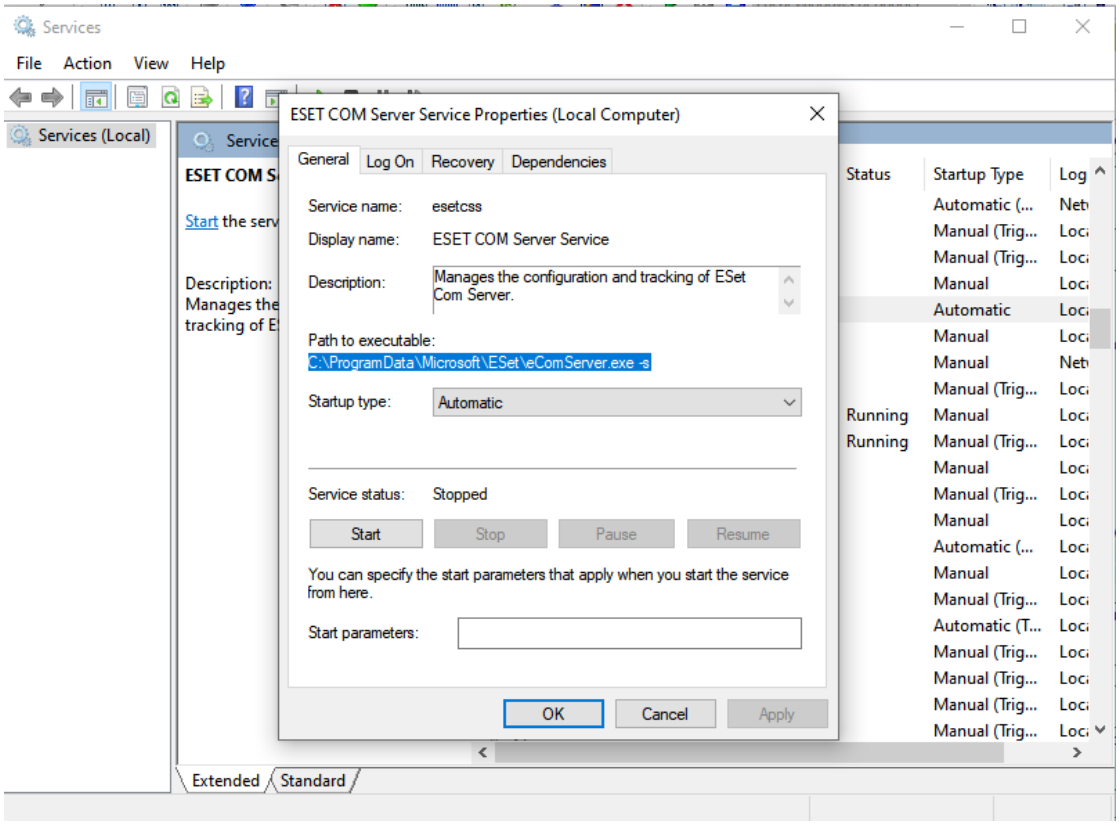
Бэкдор MeatBall

Бэкдор MeatBall — это новый имплант, обнаруженный нами в ходе исследования атак, с обширными возможностями удаленного доступа, включая создание списков активных процессов, подключенных устройств и накопителей, выполнение файловых операций, захват снимков экрана, использование удаленной оболочки и самообновление. Имплант существует в вариантах для архитектур x86 и x64.

Имплант использует схему загрузки на основе техники подмены DLL, но, в отличие от многих других имплантов, вредоносная нагрузка хранится в загрузчике вредоносной DLL, а не в отдельном файле.

Когда приложение на хосте, уязвимое к атаке Dll hijacking, выполняется без параметров, имплант вызывает функцию IsNTAdmin и, если у него достаточный уровень привилегий, создает службу под именем “esetcss”. В противном случае он просто добавляет себя к ключу реестра “HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\esetcss”, что позволяет ему автоматически выполняться при загрузке ОС.

Служба, созданная имплантом MeatBall



В обоих случаях конфигурация имплантов предусматривает их выполнение с параметром “-S”, который означает, что имплант должен прочитать вредоносную нагрузку из файла (.dll) собственного модуля, расшифровать вредоносную нагрузку, используя однобайтовый ключ XOR, запустить “svchost.exe” и внедрить расшифрованную вредоносную нагрузку в этот процесс. Затем он запускает основной цикл взаимодействия с командным сервером, вызвав ResumeThread для “svchost.exe”.

К импланту статически прилинкована библиотека libssl.dll, которая применяется для SSL шифрования трафика между имплантом и командным сервером.

Коды команд	Описание
0x2, 0x11	Обновить адрес командного сервера
0x3	Создать список выполняемых процессов
0x5	Создать список подключенных устройств
0x6	Создать список подключенных носителей

0x7, 0x8	Собрать атрибуты даты и времени файлов в заданной папке.
0x9	Завершить процесс
0xB	Записать файл
0xC	Создать файл
0xD, 0xF	Загрузить на сервер размер и содержимое файла
0x10	Удалить файл
0x13	Выполнить файл
0x14	Закрыть соединение с командным сервером
0x15, 0x1C, 0x1D, 0x1E	Завершить собственный процесс
0x16, 0x17,0x18, 0xA, 0x1F	Создать удаленную оболочку
0x19	Рекурсивно удалить файлы из папки
0x1A, 0x1B	Сделать снимок экрана

Имплант, использующий Yandex Cloud в качестве командного сервера

Еще один обнаруженный интересный имплант использует сервис хранения данных Yandex Cloud в качестве командного сервера (https://cloud-api.yandex[.]net) аналогично вредоносному ПО, описанному в отчете. Имплант использует схему загрузки, основанную на подмене DLL. Вредоносная DLL расшифровывает тело импланта, сохраненное в отдельном файле, и внедряет его в память легитимного процесса.

Имплант использует статически прилинкованную библиотеку libcurl.dll для организации обмена данными, зашифрованными с помощью SSL. В начале своей работы он создает мьютекс "*Njg8*", чтобы не допустить выполнения более одного своего экземпляра в любой момент времени, затем собирает на хосте следующие данные:

- Имя компьютера
- Имя пользователя
- IP-адрес
- MAC-адрес
- Версия ОС
- Путь к %System%

Для выгрузки собранных данных на командный сервер имплант отправляет с помощью встроенного API-токена запрос на создание на сервере директории с уникальным именем, соответствующим хосту жертвы. Затем он создает файл с префиксом “1770_” и расширением “.dat” и сохраняет всю собранную информацию в этом файле.

Главный цикл импланта периодически проверяет наличие последних загруженных файлов с префиксами “1780_”, “1781_” и “1784_” в облачной папке с именем “content”:

- Файлы с префиксами “1780_” и “1781_” содержат код в формате PE, например, легитимное приложение и вредоносную DLL-библиотеку для операции подмены DLL на следующем шаге.
- Файлы с префиксом “1784_” содержат команды для выполнения с помощью cmd.exe. Результат выполнения команд сохраняется в файле журнала, который сразу же загружается обратно на командный сервер и удаляется с хоста жертвы.

Все загружаемые на сервер и на хост данные шифруются алгоритмом RC4.

Строки, найденные в образце, использующем Yandex Disk

```
aCrateDir       db 'crate dir',0Ah,0     ; DATA XREF: sub_452050+5A2↑o
                align 10h
aUploadHostInfo db 'upload host info',0Ah,0 ; DATA XREF: sub_452050+75D↑o
                align 4
aBeginExeccomma db 'begin execCommand',0Ah,0 ; DATA XREF: sub_452050+A82↑o
                align 4
aSleeptimeD     db 'sleeptime:%d',0Ah,0 ; DATA XREF: sub_452050+BB5↑o
                align 4
asc_627A48      db '/',0                ; DATA XREF: sub_452C30+78↑o
                                        ; .text:loc_45C9CF↑o ...
                align 4
aContent_0      db '/content/',0        ; DATA XREF: sub_452C30+11D↑o
                align 4
a1780           db '1780',0             ; DATA XREF: sub_452C30+1D2↑o
                                        ; sub_452C30+342↑o
                align 10h
a1781           db '1781',0             ; DATA XREF: sub_452C30+20E↑o
                align 4
a1784           db '1784',0             ; DATA XREF: sub_452C30+3EA↑o
```

Журнал, содержащий результат выполнения команды с помощью cmd

```
C:\Windows\system32>sc query WinCoreSvc

SERVICE_NAME: WinCoreSvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 1  STOPPED
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

C:\Windows\system32>
```

Заключение

Использование злоумышленниками облачных сервисов (таких как Dropbox, Yandex, Google и т.п.) — не новое явление, но оно набирает силу, потому что в случаях, когда бизнес-процессы организации зависят от работы с такими сервисами, подобное использование сложно ограничить, а его последствия сложно минимизировать.

Злоумышленники постоянно усложняют обнаружение и анализ угроз, скрывая вредоносную нагрузку внутри отдельного зашифрованного бинарного файла данных и внедряя вредоносный код в память легитимных приложений с помощью подмены DLL и путем непосредственной записи в память.

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.
- Рассмотрите возможность применения технологий разрешительных списков и контроля программ, чтобы предотвратить выполнение неизвестных приложений.
- Рассмотрите возможность использования режима конфигурации эталонного образа ОС (Golden Image) в разрешительных списках (Allowlisting) и контроле программ (Application Control), чтобы предотвратить выполнение любого не разрешенного ПО (включая известные уязвимые приложения).
- Рассмотрите возможность ограничения доступа к интернету из технологической сети по умолчанию с разрешением доступа конкретным пользователям на определенный период времени и только если это необходимо для выполнения сотрудниками их должностных обязанностей.

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.

Варианты FourteenHi

MD5

```
7332710D10B26A5970C5A1DDF7C83FBA (mpsvc.dll)
2A1CFA6D17627EAAA7A63F73038A93DA (taskhost.doc)
BB02A5D3E8807D7B13BE46AD478F7FBB (cclib.dll)
22E66E0BE712F2843D8DB22060088751 (ToastUI.exe.png)
D75C7BD965C168D693CE8294138136AE (ToastUI.exe.dat)
```

IP-адрес/URL командного сервера

```
sfb.odk-saturn[.]com/dialin/login
87.121.52[.]86
```

Backdoor.Win32.MeatBall

MD5

```
FFF248DB8066AE3D30274996BAEDDAB6 (oleacc.dll)
```

IP-адреса/URL командных серверов

```
freetranslatecenter[.]com
help.freetranslatecenter[.]com
onlinenewscentral[.]com
onlinemapservices[.]com
search.onlinemapservices[.]com
help.onlinemapservices[.]com
apps.onlinemapservices[.]com
edit.onlinemapservices[.]com
booking-onlines[.]com
81.28.13[.]74
92.38.160[.]142
92.38.188[.]135
92.38.190[.]55
103.221.222[.]133
```

193.109.78[.]243
193.124.112[.]206
194.87.95[.]125

Имплант, использующий Yandex Cloud в качестве командного сервера

MD5

A05D6D7A6A1E9669FC4C61223DA3953F (dbghelp.dll)
2F5C889A819CFE0804005F7CE5FD956E (vmService.pkg)

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
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 терминале 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 с целью закрепления в системе.
	T1053.005	**Scheduled Task/Job: Scheduled Task** Выполнение вредоносного ПО посредством задачи планировщика задач Windows, созданной злоумышленниками.
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 вредоносных DLL.
Discovery	**T1033**	**System Owner/User Discovery** Использование злоумышленниками systeminfo, whoami и net для получения информации о пользователе и зараженной системе.
	T1057	**Process Discovery** Использование злоумышленниками tasklist для получения списка активных процессов.
Command and Control	**T1071.001**	**Application Layer Protocol: Web Protocols** Взаимодействие вредоносного ПО с командным сервером по протоколам HTTPS и raw TCP.
	T1573.001	**Encrypted Channel: Symmetric Cryptography** Использование вредоносным ПО алгоритмов RC4 и SSL TLS v3 (с помощью libssl.dll) для шифрования соединений.
Exfiltration	**T1041**	**Exfiltration Over C2 Channel** Вывод злоумышленниками украденных данных через Dropbox, Yandex Disk, почту Yandex и файлообменные сервисы в качестве канала связи с командным сервером.

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